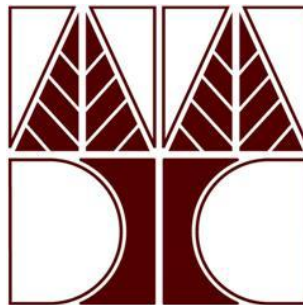


Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΕΝΟΣ ΟΛΟΚΛΗΡΩΜΕΝΟΥ
ΛΟΓΙΣΜΙΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ
ΔΙΑΧΕΙΡΙΣΗ ΚΤΗΝΟΤΡΟΦΙΚΩΝ ΜΟΝΑΔΩΝ**

Αργυρώ Κυριάκου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2012

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη ενός ολοκληρωμένου λογισμικού συστήματος για
την διαχείριση κτηνοτροφικών μονάδων**

Αργυρώ Κυριάκου

Επιβλέπων Καθηγητής

Κωνσταντίνος Παττίχης

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση
των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος
Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2012

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου, κ. Κωνσταντίνο Παττίχη, για την υποστήριξη και τις υποδείξεις του όσον αφορά την εκπόνηση της διπλωματικής αυτής εργασίας. Πέρα όμως από τα πλαίσια της διπλωματικής αυτής εργασίας, ο κ. Παττίχης με υποστήριξε και ηθικά δίνοντας μου ανεκτίμητες συμβουλές για την μετέπειτα πορεία μου και βοήθεια σε ότι και αν χρειάστηκα. Ο κ. Παττίχης αποτελεί αναμφισβήτητα ένα από τα θεμελιώδη και ανεκτίμητα πρόσωπα που έχει το τμήμα Πληροφορικής.

Ένα μεγάλο ευχαριστώ οφείλω να πω στον μέντορά μου, κ. Ιωάννη Κωνσταντίνου για την πραγματικά πολύτιμη βοήθεια του σε όλη την διάρκεια της εκπλήρωσης της διπλωματικής μου εργασίας. Ο κ. Ιωάννης υπήρξε πραγματικός συμπαράστατης, και ιδιαίτερα τον ευχαριστώ για την κατανόηση και την υποστήριξη που έδειξε στις δυσκολίες που αντιμετώπισα προς την επίτευξη των στόχων της παρούσας διπλωματικής.

Δεν θα μπορούσα βέβαια να μην ευχαριστήσω τους γονείς μου, Γιώργο και Κυριακή, για την ηθική και οικονομική υποστήριξη τους, σε όλα τα τέσσερα χρόνια φοίτησης μου στο Πανεπιστήμιο Κύπρου. Οι γονείς μου στάθηκαν δίπλα μου σε όλες τις δύσκολες, αλλά και ευχάριστες στιγμές, και αποτελούν ένα σημαντικό παράγοντα των επιτυχιών που είχα μέχρι σήμερα στο τμήμα Πληροφορικής.

Σημαντικό στήριγμα αποτέλεσαν βέβαια και οι φίλοι μου, και ιδιαίτερα θα ήθελα να ευχαριστήσω τις: Ελένη Ματσάγγου, Γαβριέλα Κουπεπίδου, Γιάννα Λοΐζου, Μαριάννα Χριστοφόρου, και Νίκη Κουππή. Τις ευχαριστώ για τις όμορφες και δύσκολες στιγμές που ζήσαμε μαζί σε όλη την διάρκεια φοίτησης μας στο Πανεπιστήμιο Κύπρου.

Σημαντικοί φορείς για την εκπλήρωση της διπλωματικής μου εργασίας αποτέλεσαν και ο κτηνίατρος Δρ. Κωνσταντίνος Κωνσταντίνου, ο κ. Δημήτρης Κωνσταντίνου - ιδιοκτήτης φάρμας, και ο κ. Νικόλας Αντωνιάδης, επίσης ιδιοκτήτης φάρμας.

Λευκωσία, 16 Μαΐου 2012

Αργυρώ Κυριάκου

Περίληψη

Στο κεφάλαιο 1 παρουσιάζονται οι λόγοι στους οποίους οφείλεται η ανάπτυξη του εν λόγω λογισμικού συστήματος, το οποίο προορίζεται για χρήση από τους ιδιοκτήτες κτηνοτροφικών μονάδων προς εκπλήρωση των στόχων τους.

Στο κεφάλαιο 2 παρουσιάζεται η ανάλυση απαιτήσεων του συστήματος. Παρουσιάζεται δηλαδή ο σκοπός ανάπτυξης του συστήματος και οι λειτουργίες που διεκπεραιώθηκαν. Επιπλέον, δίνεται μία γενική περιγραφή του συστήματος, καθώς και αναφορά στις συγκεκριμένες απαιτήσεις του συστήματος.

Στο κεφάλαιο 3 γίνεται η ανάλυση των προδιαγραφών του συστήματος. Η ανάλυση των προδιαγραφών περιλαμβάνει τα διαγράμματα ροής δεδομένων, την μηχανογράφηση επιλεγμένων τμημάτων, την ανάλυση διαδικασιών, τον προσδιορισμό αποθήκευσης των δεδομένων, τον καθορισμό των φυσικών πόρων που έχουν χρησιμοποιηθεί. Επιπλέον, παρουσιάζονται τα διαγράμματα αντικειμενοστραφούς ανάλυσης όπως τα use-case diagrams και τα analysis class diagrams. Στο κεφάλαιο αυτό δίνεται το πλάνο διαχείρισης έργου καθώς και τα διαγράμματα οντοτήτων.

Στο κεφάλαιο 4 δίνεται η ανάλυση σχεδίασης, όπου και παρουσιάζεται η αρχιτεκτονική και η αναλυτική σχεδίαση των διεργασιών του συστήματος. Κατά την αρχιτεκτονική σχεδίαση του συστήματος, γίνεται προσδιορισμός των μονάδων που αποτελούν το λογισμικό και των σχέσεων μεταξύ τους, καθώς και καθορισμός των διαγραμμάτων ροής δεδομένων που δίνονται ως είσοδος. Με την αναλυτική σχεδίαση, εξασφαλίζεται η εγκυρότητα των λειτουργιών που πρέπει θα εκτελούνται.

Στο κεφάλαιο 5 παρουσιάζεται η φάση της Υλοποίησης, όπου το σύστημα διαμορφώνεται ανάλογα με τις προδιαγραφές και την σχεδίαση, οι οποίες έχουν προηγηθεί. Το σύστημα υλοποιείται με τις κατάλληλες γλώσσες προγραμματισμού και τα αντίστοιχα εργαλεία τα οποία έχουν κριθεί ως τα πιο κατάλληλα, και τέλος, το σύστημα ελέγχεται και τίθεται σε λειτουργία. Για την υλοποίηση του συστήματος χρησιμοποιήθηκαν η Microsoft C - Sharp .NET και ο Microsoft SQL Server 2008.

Στο κεφάλαιο 6 γίνεται αναφορά στα συμπεράσματα που προέκυψαν από την ανάπτυξη του συγκεκριμένου λογισμικού, καθώς και εισηγήσεις για μελλοντική βελτίωση ή επέκταση του συστήματος.

Περιεχόμενα

Κεφάλαιο 1. Εισαγωγή	1
1.1 Κίνητρα.....	1
1.2 Συνεισφορά	1
1.3 Μεθοδολογία.....	2
1.4 Εφαρμογή μεθοδολογίας.....	4
Κεφάλαιο 2. Ανάλυση Απαιτήσεων	8
2.1 Εισαγωγή	8
2.1.1 Σκοπός	8
2.1.2 Εμβέλεια	8
2.1.3 Ορισμοί, επεξηγήσεις	9
2.1.4 Πηγές	9
2.1.5 Επισκόπηση περιεχομένων της φάσης απαιτήσεων	9
2.2 Γενική περιγραφή του συστήματος	10
2.2.1 Προοπτική συστήματος	10
2.2.2 Λειτουργίες του συστήματος	13
2.2.3 Χαρακτηριστικά χρηστών.....	13
2.2.4 Περιορισμοί	14
2.2.5 Υποθέσεις και εξαρτήσεις	14
2.2.6 Απαιτήσεις που αναβάλλονται.....	15
2.3 Συγκεκριμένες Απαιτήσεις	15
2.3.1 Απαιτήσεις εξωτερικών διασυνδέσεων	15
2.3.2. Λειτουργίες Συστήματος	17
2.3.3. Απαιτήσεις Απόδοσης.....	44
2.3.4. Χαρακτηριστικά συστήματος λογισμικού	44
2.3.5. Λογικές Απαιτήσεις Βάσης Δεδομένων	45
Κεφάλαιο 3. Ανάλυση Προδιαγραφών	50
3.1 Εισαγωγή	50
3.2 Τεχνική Δομημένης Ανάλυσης Συστήματος	51
3.2.1 Δημιουργία ΔΡΔ	52
3.2.2 Μηχανογράφηση Τμημάτων.....	83
3.2.3 Εκλέπτυνση Ροών Δεδομένων	84
3.2.4 Προσδιορισμός Λογικής των διαδικασιών	100
3.2.5 Προσδιορισμός Αποθήκευσης Δεδομένων.....	103
3.2.6 Καθορισμός φυσικών πόρων	105
3.2.7 Ορισμός προδιαγραφών για Είσοδο/Εξοδο δεδομένων	112

3.2.8 Εκτίμηση μεγέθους	119
3.2.9 Απαιτήσεις σε hardware	120
3.3 Βήματα Αντικειμενοστραφούς Ανάλυσης.....	121
3.3.1 Εισαγωγή	121
3.4 Διαγράμματα Οντοτήτων.....	131
Κεφάλαιο 4: Ανάλυση Σχεδίασης.....	135
4.1 Εισαγωγή.....	135
4.1.1 Σκοπός	135
4.1.2 Επισκόπηση Φάσης Σχεδίασης.....	136
4.2 Αρχιτεκτονική Σχεδίαση.....	137
4.2.1. Ανάλυση Ροών Δεδομένων.....	137
4.3 Αναλυτική Σχεδίαση.....	160
4.3.1 Οντότητες με κοινές λειτουργίες	160
4.4 Graphical User Interface	183
Κεφάλαιο 5. Φάση Υλοποίησης	191
5.1 Εισαγωγή	191
5.2 Σύντομη Περιγραφή Λειτουργικότητας του Προγράμματος.....	191
5.3 Εργαλεία και Τεχνολογίες	194
5.3.1 Microsoft C - Sharp .NET.....	195
5.3.2 Microsoft SQL Server®2008.....	196
5.3.3 SAP Crystal Reports	197
5.3.4 Περιγραφή βάσης δεδομένων	197
5.3.5 Δύσκολα σημεία στον κώδικα	199
5.3.6 Πηγαίος κώδικας με σχόλια.....	202
Κεφάλαιο 6. Συμπεράσματα	244
6.1 Εισαγωγή	244
6.2 Ευχρηστία.....	244
6.3 Μελλοντική Εργασία	245
Βιβλιογραφία	246
Παράρτημα Α.....	247

Κεφάλαιο 1. Εισαγωγή

Κεφάλαιο 1. Εισαγωγή	1
1.1 Κίνητρα	1
1.2 Συνεισφορά	1
1.3 Μεθοδολογία.....	2
1.4 Εφαρμογή μεθοδολογίας.....	4

Κεφάλαιο 1. Εισαγωγή

1.1 Κίνητρα

Για την διεκπεραίωση των απαιτήσεων του συστήματος εφαρμόστηκαν γνώσεις και τεχνικές, οι οποίες διδάχτηκαν σε προηγούμενα μαθήματα του προγράμματος σπουδών. Το γεγονός αυτό αποτέλεσε ένα από τα κίνητρα τα οποία με ώθησαν στο να επιλέξω αυτό το θέμα διπλωματικής, αφού με τον τρόπο αυτό θα είχα την ευκαιρία να εφαρμόσω στην πράξη όσα διδάχτηκα.

Αναμφισβήτητα, η ανάγκη του συστήματος αυτού για βάση δεδομένων, η οποία θα βοηθούσε στην εκπλήρωση των στόχων του συστήματος, ήταν ακόμη ένας από τους λόγους που το ενδιαφέρον μου για την διπλωματική αυτή μεγάλωσε. Οι βάσεις δεδομένων αποτελούν αναπόσπαστο κομμάτι για την ανάπτυξη ενός πληροφοριακού συστήματος, και παρέχουν κυρίως διευκολύνσεις σχετικά με την διαχείριση του υπό-ανάπτυξη συστήματος.

Επιπλέον, η ανάγκη ανάπτυξης ενός τέτοιου συστήματος το οποίο προορίζεται σε κτηνοτροφικές μονάδες, οι οποίες δεν είναι μηχανογραφημένες, αποτέλεσε σημαντικό κίνητρο για την αυτοματοποίηση και την διεκπεραίωση των λειτουργιών των κτηνοτροφικών μονάδων.

1.2 Συνεισφορά

Το σύστημα σχεδιάστηκε και υλοποιήθηκε με σκοπό να παρέχει βοήθεια και διευκολύνσεις στην διαχείριση των ποιμνιοστασιών από τους κτηνοτρόφους. Επιπλέον, το σύστημα παρέχει την δυνατότητα στους ιδιοκτήτες των κτηνοτροφικών μονάδων να ελέγχουν τα έσοδα και τα έξοδα τους καθώς τους επιτρέπει να επιβλέπουν την οικονομική διαχείριση της φάρμας τους. Ένα από τα σημαντικότερα θέματα που απασχολεί τους κτηνοτρόφους είναι η διαδικασία της επιλογής των προβάτων [9], τα οποία θα διαμορφώσουν την επόμενη γενεά προβάτων που θα έχει η φάρμα. Το λογισμικό αυτό σύστημα, με χρήση νευρωνικών δικτύων [7] και ανάλυσης στοιχείων των προβάτων, προσφέρει την δυνατότητα επιλογής των προβάτων. Αξίζει να σημειωθεί ότι, η διεπαφή του συστήματος έχει διαμορφωθεί με τέτοιο τρόπο ώστε το σύστημα να είναι προσιτό και εύκολο στην λειτουργία του καθώς τα άτομα που θα το χρησιμοποιούν δεν κατέχουν εξειδικευμένες γνώσεις πληροφοριακών συστημάτων.

1.3 Μεθοδολογία

Ο κύκλος ζωής ενός λογισμικού αποτελεί την παλαιότερη μέθοδο ανάπτυξης συστημάτων η οποία χρησιμοποιείται μέχρι και σήμερα σε μικρά και μεγάλα έργα ανάπτυξης λογισμικού. Βάση αυτής της μεθοδολογίας, ένα λογισμικό μπορεί να είναι λειτουργήσιμο και να ανταποκρίνεται στις προσδοκίες των χρηστών του, για δεδομένη χρονική περίοδο, έχοντας αρχή, μέση και τέλος. Ο κύκλος ζωής για την ανάπτυξη ενός λογισμικού συστήματος αποτελείται από τα εξής έξι στάδια:

1. Απαιτήσεις και Προδιαγραφές
2. Ανάλυση
3. Σχεδιασμός
4. Υλοποίηση
5. Εγκατάσταση
6. Συντήρηση

Για να ολοκληρωθεί ένα στάδιο και να ξεκινήσει το επόμενο, είναι απαραίτητο να διεκπεραιωθούν οι βασικές δραστηριότητες από τις οποίες αποτελείται το κάθε στάδιο.

Η συνολική εργασία που εκπονείται κατά τον κύκλο ζωής πραγματοποιείται ισάξια από τους τελικούς χρήστες και τους ειδικούς των συστημάτων πληροφορικής. Το μεγαλύτερο μέρος της δουλειάς που απαιτείται στις φάσεις: Ανάλυση Απαιτήσεων και Προδιαγραφών, Ανάλυση Συστήματος και Υλοποίησης, εκπονείται από τους αναλυτές και τους προγραμματιστές του συστήματος. Οι τελικοί χρήστες είναι υπεύθυνοι να αναφέρουν και να περιγράψουν τις ανάγκες τους, και να αξιολογούν βέβαια την δουλειά του τεχνικού προσωπικού. Αν οι τελικοί χρήστες συμφωνούν, το τρέχον στάδιο ολοκληρώνεται με μία επίσημη συμφωνία μεταξύ των χρηστών και των ειδικών του συστήματος.

Το στάδιο Απαιτήσεις και Προδιαγραφές απαντά σε ερωτήσεις του τύπου «Γιατί υπάρχει ανάγκη για την ανάπτυξη και υλοποίηση ενός νέου συστήματος;» και ερωτήματα όπως «Τι πρέπει να επιτευχθεί;». Στο στάδιο αυτό προσδιορίζεται κατά πόσο ο οργανισμός-εταιρεία αντιμετωπίζει κάποιο πρόβλημα, και κατ'επέκταση αν αυτό μπορεί να επιλυθεί μέσω της υλοποίησης ενός νέου συστήματος ή μέσω της τροποποίησης μέρους του υπάρχοντος συστήματος. Επιπλέον, στο στάδιο αυτό γίνεται

προσδιορισμός των γενικών στόχων, καθορισμός του σκοπού του έργου και σχεδιασμός του χρονοδιαγράμματος έτσι ώστε να παρουσιαστεί στην διοίκηση του οργανισμού ένα πλάνο πορείας.

Το στάδιο Ανάλυσης αναλύει λεπτομερώς τα προβλήματα που παρατηρούνται στο υπάρχον σύστημα, περιγράφει το πώς οι στόχοι που καθορίστηκαν στο στάδιο των απαιτήσεων και προδιαγραφών μπορούν να επιτευχθούν μέσω εναλλακτικών λύσεων. Για κάθε εναλλακτική λύση εξετάζεται η σκοπιμότητα της με σκοπό να γίνει η σωστή επιλογή από το διοικητικό προσωπικό του οργανισμού-εταιρείας. Η ανάλυση του συστήματος θέλει τους αναλυτές και του προγραμματιστές του συστήματος να προβαίνουν στην συλλογή πληροφοριών, στην έρευνα και στην μελέτη εγγράφων και αναφορών τα οποία δημιουργούνται από το υπάρχον σύστημα, και επιπλέον, να παρατηρούν τον τρόπο λειτουργίας του. Οι δραστηριότητες αυτές διεκπεραιώνονται μέσω ερωτηματολογίων προς τους χρήστες και μέσω προσωπικών συνεντεύξεων. Τα πορίσματα των πληροφοριών που εξάγονται από αυτές τις δραστηριότητες χρησιμοποιούνται για να καθοριστούν οι ανάγκες του συστήματος. Επιπλέον, κατά την φάση της ανάλυσης περιγράφονται εκτενώς όλες οι υπόλοιπες δραστηριότητες του κύκλου ζωής, καθώς και των εργασιών κάθε φάσης.

Κατά την φάση της ανάλυσης Σχεδίασης, σχεδιάζονται σε λογικό και φυσικό επίπεδο οι προδιαγραφές του υπό-υλοποίηση συστήματος και παρουσιάζονται μέσω διαγραμμάτων ροής δεδομένων, δομής προγραμμάτων, καθώς και μέσω της ροής του συστήματος. Το γεγονός αυτό βοηθά ώστε οι προγραμματιστές να δημιουργήσουν ορθά την κάθε διεργασία και να διευκολυνθούν στην όλη υλοποίηση του συστήματος.

Κατά την φάση της Υλοποίησης, γίνεται μετατροπή των διαγραμμάτων του σταδίου σχεδίασης σε κώδικα προγραμμάτων λογισμικού. Για την σωστή και προσδοκώμενη λειτουργία του συστήματος, απαιτείται η συνεργασία μεταξύ αναλυτών και προγραμματιστών ώστε να γραφτεί προσαρμοσμένος κώδικας προγραμμάτων ανάλογα με την κάθε περίπτωση.

Κατά την φάση της Εγκατάστασης, πραγματοποιούνται τα τελικά βήματα ώστε το νέο ή τροποποιημένο σύστημα να τεθεί σε λειτουργία. Μέσω δοκιμών και ελέγχων, διαπιστώνεται κατά πόσο το λογισμικό ανταποκρίνεται ικανοποιητικά στις αρχικές προσδοκίες του οργανισμού, τόσο από τεχνική, όσο και από επιχειρηματική και

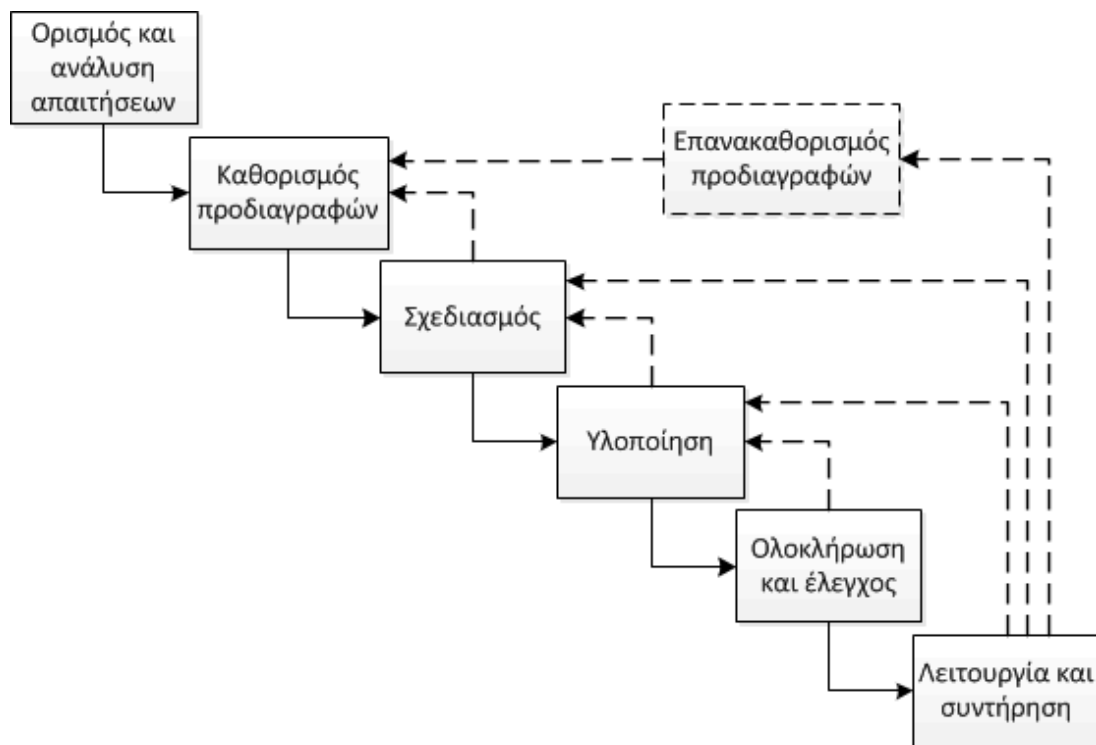
λειτουργική άποψη. Οι μελλοντικές χρήστες του συστήματος εκπαιδεύονται κατάλληλα για να μπορέσουν να λειτουργήσουν σωστά το νέο σύστημα. Το εγχειρίδιο χρήσης, το οποίο παραδίδεται στον οργανισμό, περιγράφει λεπτομερώς την εγκατάσταση και όλες τις λειτουργίες του νέου συστήματος.

Κατά την φάση Συντήρησης, γίνεται αξιολόγηση του συστήματος με βάση την χρήση του. Σκοπός της αξιολόγησης είναι να εξακριβωθεί σε ποιο βαθμό το νέο σύστημα όντως ανταποκρίνεται στους στόχους που είχαν τεθεί από την αρχή της ανάπτυξης. Αρνητικά σχόλια συνεπάγονται αναθεωρήσεις ή τροποποιήσεις στο υλοποιημένο σύστημα. Η απόδοση του συστήματος, καθώς και η υποκειμενική ικανοποίηση των χρηστών του, εξαρτώνται από την συνεχή συντήρηση του.

1.4 Εφαρμογή μεθοδολογίας

Η πιο πάνω μεθοδολογία εφαρμόστηκε κάνοντας χρήση του μοντέλου καταρράκτη (Waterfall) [3].

Το μοντέλο καταρράκτη είναι ένα κλασσικό μοντέλο κύκλου ζωής ανάπτυξης ενός συστήματος λογισμικού που χρησιμοποιεί μία γραμμική και ακολουθιακή μέθοδο. Η μέθοδος αυτή παρέχει στόχους σε κάθε φάση της ανάπτυξης του συστήματος. Το μοντέλο αυτό συνηθίζεται να χρησιμοποιείται ως μέθοδος ανάπτυξης συγκεκριμένων κομματιών ενός μεγάλου λογισμικού. Το όνομα του μοντέλου πηγάζει από την μορφολογική του απεικόνιση, αφού είναι εμφανής η κάθετη μετάβαση από την μία φάση στην επόμενη. Λόγω της έλλειψης επανάληψης ή επικάλυψης στα βήματα της μεθόδου, στο μοντέλο καταρράκτη παρατηρείται μία απλοποίηση του χρονοπρογραμματισμού των εργασιών του. Σε αυτό συμβάλλει και το γεγονός ότι πραγματοποιείται συνεχής έλεγχος και τεκμηρίωση του κάθε βήματος, και καμία φάση δεν θεωρείται τελειωμένη πριν γραφεί και εγκριθεί η τεκμηρίωση της.



Διάγραμμα 1.1 - Μοντέλο Καταρράκτη

Το μοντέλο καταρράκτη αποτελείται από τις ακόλουθες φάσεις:

1. Ορισμός και ανάλυση απαιτήσεων: καθορίζονται οι απαιτήσεις των χρηστών του συστήματος, καθώς επίσης και οι στόχοι και τυχόν περιορισμοί του συστήματος.
2. Καθορισμός προδιαγραφών: καταγράφονται οι απαιτήσεις και οι λειτουργικές προδιαγραφές του συστήματος.
3. Σχεδιασμός: οι λειτουργίες του λογισμικού παρουσιάζονται μέσω αρχιτεκτονικών και αναλυτικών διαγραμμάτων για να μπορέσουν να μεταφραστούν σε εκτελέσιμο κώδικα.
4. Υλοποίηση: αναπτύσσεται ο κώδικας της εφαρμογής.
5. Ολοκλήρωση και έλεγχος: οι κώδικες των επιμέρους ενοτήτων συνενώνονται και ελέγχεται το ολοκληρωμένο πλέον σύστημα. Μετά το τέλος του ελέγχου, το σύστημα παραδίδεται στον χρήστη.
6. Λειτουργία και συντήρηση: το σύστημα εγκαθίσταται και τίθεται σε λειτουργία. Κατά την συντήρηση διορθώνονται λάθη που δεν είχαν εντοπιστεί

προηγούμενως, βελτιώνεται ο κώδικας και γενικά οι λειτουργίες του συστήματος.

Το μοντέλο καταρράκτη, παρόλο που χρησιμοποιείται ευρέως για την ανάπτυξη μεγάλων συστημάτων, παρουσιάζει κάποια μειονεκτήματα. Η αναθεώρηση του υλοποιημένου συστήματος δεν είναι εφικτή. Αυτό αποτελεί ένα από τα βασικά μειονεκτήματα αυτού του μοντέλου. Αυτό συμβαίνει γιατί η προσαρμογή των αλλαγών, μετά από την ολοκλήρωση του συστήματος, είναι δύσκολη και ακριβή. Επομένως, το μοντέλο καταρράκτη χρησιμοποιείται μόνο όταν οι απαιτήσεις του υπό-υλοποίηση λογισμικού έχουν γίνει πλήρως κατανοητές [4].

Κεφάλαιο 2. Ανάλυση Απαιτήσεων

Κεφάλαιο 2. Ανάλυση Απαιτήσεων	8
2.1 Εισαγωγή	8
2.1.1 Σκοπός	8
2.1.2 Εμβέλεια	8
2.1.3 Ορισμοί, επεξηγήσεις	9
2.1.4 Πηγές	9
2.1.5 Επισκόπηση περιεχομένων της φάσης απαιτήσεων	9
2.2 Γενική περιγραφή του συστήματος	10
2.2.1 Προοπτική συστήματος	10
2.2.2 Λειτουργίες του συστήματος	13
2.2.3 Χαρακτηριστικά χρηστών.....	13
2.2.4 Περιορισμοί	14
2.2.5 Υποθέσεις και εξαρτήσεις	14
2.2.6 Απαιτήσεις που αναβάλλονται.....	15
2.3 Συγκεκριμένες Απαιτήσεις	15
2.3.1 Απαιτήσεις εξωτερικών διασυνδέσεων	15
2.3.2. Λειτουργίες Συστήματος	17
2.3.3. Απαιτήσεις Απόδοσης.....	44
2.3.4. Χαρακτηριστικά συστήματος λογισμικού	44
2.3.5. Λογικές Απαιτήσεις Βάσης Δεδομένων	45

Κεφάλαιο 2. Ανάλυση Απαιτήσεων

2.1 Εισαγωγή

2.1.1 Σκοπός

Σκοπός της διπλωματικής μου εργασίας είναι η ανάπτυξη ενός συστήματος για τις λειτουργίες ενός συστήματος τηλεπαρακολούθησης σε κτηνοτροφική μονάδα. Υπεύθυνος καθηγητής για την διπλωματική μου εργασία είναι ο κος. Κωνσταντίνος Παττίχης. Οι λειτουργίες που θα πρέπει να διεκπεραιωθούν είναι οι ακόλουθες:

- Βάση δεδομένων που θα διαχειρίζεται τα στοιχεία των ιδιοκτητών και τα στοιχεία για τις φάρμες τους
- Βάση δεδομένων που θα διαχειρίζεται τις πληροφορίες για τα πρόβατα, τις ασθένειες τους, την παραγωγή τους σε γάλα και την κατανάλωση τους σε τροφές
- Βάση δεδομένων που θα διαχειρίζεται τις πληροφορίες για τους εργοδοτούμενους και τους κτηνιάτρους που εργάζονται σε κάθε φάρμα
- Βάση δεδομένων που θα διαχειρίζεται τις πληροφορίες για τα έξοδα και τα έσοδα της κάθε φάρμας
- Βάση δεδομένων που θα διαχειρίζεται τις πληροφορίες για τα τιμολόγια και τις αποδείξεις σχετικά με τα έσοδα και τα έξοδα της κάθε φάρμας

Με βάση τις απαιτήσεις του καθηγητή κ. Παττίχη και του βοηθού του, δημιούργησα το παρών έγγραφο απαιτήσεων. Το έγγραφο απευθύνεται στην ομάδα η οποία θα αναλάβει τον καθορισμό των προδιαγραφών του συστήματος, καθώς και στον καθηγητή κ. Παττίχη και στον βοηθό του κ. Ιωάννη.

2.1.2 Εμβέλεια

Οι κύριες λειτουργίες του συστήματος που αναμένεται να υλοποιήσουμε περιλαμβάνουν την αποθήκευση και διαχείριση δεδομένων των ιδιοκτητών, φαρμών, εργοδοτούμενων, κτηνιάτρων, προβάτων, ασθενειών, έσοδα/έξοδα της κάθε φάρμας και των τιμολογίων/αποδείξεων για την κάθε φάρμα. Με βάση αυτές τις λειτουργίες, το σύστημά μας θα πρέπει να διαχωριστεί σε ένα back-office σύστημα το οποίο θα υποστηρίζεται από μια βάση δεδομένων.

Στη βάση δεδομένων θα αποθηκεύονται πληροφορίες για τα στοιχεία των ιδιοκτητών και για τις φάρμες τους, των εργοδοτούμενων και των κτηνιάτρων που εργάζονται σε κάθε φάρμα, των προβάτων και τα στοιχεία της παραγωγής και κατανάλωσης τους, των ασθενειών τους, πληροφορίες σχετικά με τα έσοδα/έξοδα της κάθε φάρμας και των τιμολογίων/αποδείξεων για την κάθε φάρμα.

Τα πλεονεκτήματα από τη χρήση βάσης δεδομένων αναμένουμε να είναι τα εξής:

- Καλύτερη διαχείριση πληροφοριών των προβάτων
- Καλύτερη διαχείριση των λογιστικών (έσοδα – έξοδα)
- Πιο γρήγορη και αποτελεσματική λειτουργία της κτηνοτροφικής μονάδας

2.1.3 Ορισμοί, επεξηγήσεις

ΒΔ – Βάση Δεδομένων

BO – Back-Office System

FO – Front-Office System

2.1.4 Πηγές

- “Software Requirements Specification Document”
(IEEE 830-1993)
- “Object – Oriented and Classical Software Engineering” 5th Edition, Stephen R.Schach
- “System Analysis and Design Methods” 5th Edition, J.Whitten, L.Bentley, K.Dittman, Mc Graw Hill Publishers
- Σημειώσεις από το μάθημα «Τεχνολογία Λογισμικού»
- <http://www.google.com>

2.1.5 Επισκόπηση περιεχομένων της φάσης απαιτήσεων

Η δομή που θα ακολουθήσουμε στο υπόλοιπο τμήμα της παρούσας φάσης θα είναι η εξής:

- Στο δεύτερο τμήμα θα περιγράψουμε περιληπτικά τους γενικούς παράγοντες που επηρεάζουν το προϊόν και τις απαιτήσεις του, όπως είναι οι διαπροσωπικές με τα υπάρχον συστήματα και υπολογιστικό υλικό, λειτουργίες του προϊόντος,

χαρακτηριστικά των χρηστών, περιορισμούς σε υλικό, υποθέσεις και εξαρτήσεις από μελλοντικές τροποποιήσεις και απαιτήσεις που πιθανόν να υλοποιηθούν σε μελλοντικές εκδόσεις του προϊόντος.

- Στο τρίτο τμήμα θα περιγράψουμε αναλυτικά τις συγκεκριμένες απαιτήσεις του προϊόντος σε αρκετό επίπεδο λεπτομέρειας, ώστε να μπορούμε στη συνέχεια να σχεδιάσουμε ένα προϊόν που να ικανοποιεί αυτές τις απαιτήσεις και να επαληθεύσουμε στη συνέχεια ότι όντως ικανοποιούνται. Συγκεκριμένα, θα αναφερθούμε στις απαιτήσεις των εξωτερικών διαπροσωπειών, στα γνωρίσματα του συστήματος, στις απαιτήσεις απόδοσης, στους σχεδιαστικούς περιορισμούς, στα χαρακτηριστικά του συστήματος και στις λογικές απαιτήσεις βάσης δεδομένων.
- Στο τελευταίο τμήμα θα έχουμε ευρετήριο με πίνακες, σχήματα και ότι χρησιμοποιήθηκε στο έγγραφο.

2.2 Γενική περιγραφή του συστήματος

2.2.1 Προοπτική συστήματος

Στο σύστημα που θα αναπτύξω δεν υπάρχει ανάγκη για διασύνδεση με κάποια άλλα προγράμματα καθώς δεν υπάρχει κάποιο άλλο παρόμοιο σύστημα το οποίο ήδη να υπάρχει και να χρειάζεται να γίνεται σύνδεση του με το υπο-ανάπτυξη σύστημα.

Οι προοπτικές που μπορεί να έχει το σύστημα είναι οι εξής:

- Διαχείριση από περισσότερους χρήστες εκτός από τον γενικό διαχειριστή
- Αλλαγές στο προϊόν λόγω τεχνολογικών εξελίξεων
- Καταναγκαστικοί περιορισμοί που τους επιβάλλει το κράτος
- Αλλαγές λόγω απαιτήσεων του πελάτη

2.2.1.1 Διασύνδεση συστήματος με τα υπάρχοντα συστήματα

Το λογισμικό το οποίο θα αναπτύξω είναι ένα ΒΟ σύστημα που θα στηρίζεται σε μια βάση δεδομένων. Η κτηνοτροφική μονάδα δεν έχει οποιαδήποτε υποδομή όσο αφορά σε λογισμικό, αφού μέχρι σήμερα χρησιμοποιεί προγράμματα office για την εκπόνηση των πιο πάνω λειτουργιών. Οπότε το σύστημα που θα αναπτύξουμε δε θα έχει οποιαδήποτε διασύνδεση με άλλα συστήματα αφού δεν υπάρχουν.

2.2.1.2 Διεπιφάνεια χρηστών

Όπως έχω επαναλάβει ξανά, το σύστημα το οποίο θα αναπτύξω θα αποτελείται από ένα ΒΟ. Το ΒΟ που θα αναπτύξουμε θα περιέχει ένα κυρίως μενού, στο οποίο θα υπάρχει ένας αριθμός επιλογών οι οποίες θα παραπέμπουν σε κάποια άλλα υπομενού τα οποία θα παρέχουν στους χρήστες τις βασικές λειτουργίες της βάσης. Τα συστατικά στοιχεία που θα υπάρχουν στο ΒΟ είναι:

- Command buttons για επιλογή λειτουργίας
- Φόρμες για εισαγωγή δεδομένων
- Text boxes για εισαγωγή δεδομένων
- Combo boxes για εισαγωγή δεδομένων
- Πίνακες για παρουσίαση αποτελεσμάτων

2.2.1.3 Διασύνδεση υλικού

Το πρόγραμμα θα επικοινωνεί με τη βάση του συστήματος που θα αναπτύξω. Δηλαδή το πρόγραμμα θα παίρνει πληροφορίες από τη βάση δεδομένων. Επίσης θα υπάρχουν εκτυπωτές που θα επικοινωνούν με το σύστημα για την εκτύπωση διαφόρων πληροφοριών από το σύστημα καθώς και των αναφορών, όπως αποδείξεων, τιμολογίων, κλπ.

2.2.1.4 Λογισμικό διασύνδεσης με άλλα συστήματα

Τα συστήματα που θα συνδέονται με το λογισμικό είναι τα προγράμματα που θα χρησιμοποιεί ο χρήστης για την εισαγωγή στοιχείων σχετικά με την παραγωγή γάλακτος και την κατανάλωση τροφής για κάθε πρόβατο, αφού η καταμέτρηση τους γίνεται με την χρήση ειδικών οργάνων .

2.2.1.5 Επικοινωνιακή διασύνδεση

Το ΒΟ θα είναι εγκατεστημένο στον κεντρικό εξυπηρετητή, πάνω στον οποίο θα ενώνεται ο χρήστης του συστήματος για να κάνει τις αλλαγές που επιθυμεί πάνω στη ΒΔ. Επίσης το πρόγραμμα το οποίο θα υποστηρίζει τις λειτουργικότητες που θα αναπτύξω θα είναι επίσης εγκατεστημένο στον ίδιο κεντρικό εξυπηρετητή και θα ενημερώνεται με βάση τις αλλαγές που θα γίνονται μέσω του ΒΟ.

2.2.1.6 Περιορισμοί μνήμης

Λόγω του ότι για κάθε πρόβατο που υπάρχει στην φάρμα χρειάζεται να καταγράφουμε στοιχεία σχετικά με την ημερήσια παραγωγή του σε γάλα και την κατανάλωση του σε τροφή, και επιπλέον χρειάζεται η καταγραφή πληροφοριών σχετικά με τις ασθένειες και τα φάρμακα του κάθε προβάτου, θα χρειαστεί τον απαιτούμενο χώρο μνήμης που να καλύπτει αυτές τις ανάγκες. Σε περίπτωση που δεν υπάρχει αυτός ο χώρος στον υφιστάμενο εξυπηρετητή, ο διαχειριστής μπορεί να προμηθευτεί εξωτερικό χώρο μνήμης.

2.2.1.7 Εξειδικευμένες λειτουργίες

Στο χρήστη του συστήματος θα παρέχονται ένας αριθμός λειτουργιών, κάποιων εκ των οποίων μπορεί να εκτελεί ανά τακτά χρονικά διαστήματα. Οι λειτουργίες αυτές σχετίζονται τόσο με την αποθήκευση δεδομένων και πληροφοριών των χρηστών, όσο και με την ανάκτησή τους σε περίπτωση σφαλμάτων είτε του συστήματος, είτε του ίδιου του χρήστη.

Τέτοιες λειτουργίες που θέλουμε να υποστηρίξει το σύστημά μας είναι:

- Backup – αποθήκευση δεδομένων της βάσης σε τακτά χρονικά διαστήματα έτσι ώστε σε περίπτωση οποιασδήποτε βλάβης να μην χαθούν τα δεδομένα και να έχουμε αυτά τα οποία αποθηκεύτηκαν αυτόματα από το σύστημα την τελευταία φορά.
- Recover – επανάκτηση των δεδομένων τα οποία κάναμε backup σε περίπτωση που προκλήθηκε οποιοδήποτε πρόβλημα.
- Security (Antivirus Secure Web Link)

2.2.1.8 Προετοιμασία για υποδοχή του συστήματος

Ο διαχειριστής της κτηνοτροφικής μονάδας δε χρησιμοποιεί κάποιο υπάρχον λογισμικό αλλά προγράμματα office. Έτσι θα υπάρχει η δυνατότητα φόρτωσης των υφιστάμενων δεδομένων της βάσης.

2.2.2 Λειτουργίες του συστήματος

2.2.2.1 Διαχείριση Φάρμων

2.2.2.2 Διαχείριση Προβάτων

2.2.2.3 Διαχείριση Ιδιοκτητών

2.2.2.4 Διαχείριση Εργοδοτούμενων

2.2.2.5 Διαχείριση Κτηνιάτρων

2.2.2.6 Διαχείριση Ιατρικών εξετάσεων για κάθε πρόβατο

2.2.2.7 Διαχείριση Φάρμακα προβάτων

2.2.2.8 Διαχείριση Ασθένειες προβάτων

2.2.2.9 Διαχείριση έσοδα φάρμας

2.2.2.10 Διαχείριση έξοδα φάρμας

2.2.2.11 Διαχείριση τιμολογίων

2.2.2.12 Διαχείριση αποδείξεων

2.2.2.13 Διαχείριση LOG FILE

2.2.3 Χαρακτηριστικά χρηστών

Πρόσβαση στο σύστημα θα έχει μόνο ο ιδιοκτήτης της κτηνοτροφικής μονάδας, καθώς και ο επιστάτης της κάθε φάρμας και κάποιοι άλλοι υπάλληλοι, αλλά στο μέλλον μπορεί να προκύψουν και άλλοι χρήστες οι οποίοι θα είναι υπάλληλοι της κτηνοτροφικής μονάδας.

Ο διαχειριστής δεν διαθέτει αρκετές γνώσεις όσον αφορά το θέμα των λογισμικών και γενικά της πληροφορικής. Παρόλα αυτά η διεπιφάνεια του συστήματος που θα αναπτύξω θα είναι φιλική και απλή ώστε να διευκολύνει τον χρήστη να προσαρμοστεί όσο πιο γρήγορα γίνεται. Θα γίνει παρουσίαση και επεξήγηση του λογισμικού και θα υπάρχει συνεχής επικοινωνία μεταξύ των προγραμματιστών και του διαχειριστή για οποιεσδήποτε διευκρινήσεις σχετικά με το λογισμικό.

2.2.4 Περιορισμοί

- Safety and security

Το λογισμικό και το ΒΟ τα οποία θα αναπτύξω πρέπει να είναι ασφαλή έτσι ώστε να εμποδίζεται η πρόσβαση σε μη εξουσιοδοτημένα άτομα. Για το λόγο αυτό θα εφαρμόσουμε συστήματα ελέγχου και περιορισμού της πρόσβασης τα οποία σε συνδυασμό με το firewall του server θα αποτρέπουν την πρόσβαση σε μη εξουσιοδοτημένα άτομα.

- Για τη χρησιμοποίηση του λογισμικού που θα αναπτυχθεί θα είναι αναγκαία η ύπαρξη ενός ή περισσότερων ηλεκτρονικών υπολογιστών για τον ιδιοκτήτη.
- Η κτηνοτροφική μονάδα δεν διαθέτει κάποιο κεντρικό εξυπηρετητή για τοποθέτηση της ΒΔ γι' αυτό υπάρχει η ανάγκη για την ενοικίαση ή αγορά εξυπηρετητή από κάποια εταιρεία παροχής υπηρεσιών πληροφορικής.
- Επειδή το σύστημα θα χρησιμοποιεί μια βάση δεδομένων και επομένως υπάρχει ανάγκη για μεγάλη χωρητικότητα σε μνήμη θα είναι αναγκαία η αγορά ενός external hardware μνήμης.
- Περιορισμοί από το κράτος, όπως αλλαγές στη νομοθεσία για τον αριθμό των προβάτων και των γονότυπων τους, περιορισμοί σχετικά με τις εγκαταστάσεις της φάρμας κλπ...
- Πρέπει το λογισμικό να παρέχει επικοινωνία με τα άλλα προγράμματα που χρησιμοποιεί ο ιδιοκτήτης της κτηνοτροφικής μονάδας για την εκπόνηση των εργασιών του.
- Πρέπει οι υπεύθυνοι για την ανάπτυξη της ΒΔ να φορτώσουν τα υπάρχοντα δεδομένα της κτηνοτροφικής μονάδας στο νέο σύστημα.

2.2.5 Υποθέσεις και εξαρτήσεις

Παράγοντες οι οποίοι μπορούν να επηρεάσουν τις απαιτήσεις του συστήματος αφορούν μελλοντικές αλλαγές όσον αφορά τους εργαζόμενους και τις εγκαταστάσεις της κτηνοτροφικής μονάδας. Στην περίπτωση που προστεθούν εργαζόμενοι στην εταιρεία θα πρέπει η μονάδα να δημιουργήσει περισσότερους λογαριασμούς. Θα πρέπει επίσης όλοι οι εξουσιοδοτημένοι εργαζόμενοι να μπορούν να έχουν πρόσβαση και να χρησιμοποιούν το πρόγραμμα που θα συνδέεται με τη ΒΔ. Στην περίπτωση που επεκταθούν οι εγκαταστάσεις της κτηνοτροφικής μονάδας θα πρέπει η διασύνδεση του συστήματος με τα υπόλοιπα παραρτήματα της μονάδας να γίνεται μέσω του διαδικτύου.

2.2.6 Απαιτήσεις που αναβάλλονται

Σε οποιαδήποτε στιγμή λειτουργίας του το λογισμικό θα πρέπει να ικανοποιεί όλες τις λειτουργικότητες για τις οποίες σχεδιάστηκε. Αν προκύψουν μελλοντικές αλλαγές δεν πρέπει να υπάρχει καμιά καθυστέρηση σε καμιά από τις λειτουργίες του λογισμικού.

2.3 Συγκεκριμένες Απαιτήσεις

2.3.1 Απαιτήσεις εξωτερικών διασυνδέσεων

Όλα τα δεδομένα και οι πληροφορίες που χρειάζονται για τη λειτουργία του συστήματος βρίσκονται συγκεντρωμένα σε μια βάση δεδομένων. Το λογισμικό που θα υλοποιήσω θα επικοινωνεί με αυτή τη βάση δεδομένων για να εισάγει, να επεξεργάζεται και να εξάγει τις πληροφορίες. Επίσης το λογισμικό και η βάση θα πρέπει να επικοινωνούν με άλλα υφιστάμενα προγράμματα που χρησιμοποιεί ο ιδιοκτήτης για την εισαγωγή στοιχείων παραγωγής γάλακτος, κατανάλωση τροφής για κάθε πρόβατο, τα έσοδα και τα έξοδα της κάθε φάρμας, αποδείξεις και τιμολόγια κ.α.

2.3.1.1 Διεπιφάνεια χρηστών

Η διεπιφάνεια έχει απώτερο στόχο να καθιστά εύκολο το έργο του χρήστη καθώς δεν έχει το κατάλληλο επίπεδο γνώσης σε θέματα λογισμικού. Ο χρήστης θα πρέπει να καθοδηγείται από την ίδια την διεπιφάνεια, καθιστώντας έτσι πιο εύκολη τη χρήση του λογισμικού για τον χρήστη, καθώς δε χρειάζεται να θυμάται τα βήματα που πρέπει να ακολουθήσει για μια ενέργεια που επιθυμεί να εκτελέσει.

Front office:

Το Front office πρέπει να έχει φιλικό προς τον χρήστη περιβάλλον. Το μενού που θα υπάρχει θα διευκολύνει την πλοήγηση του χρήστη μέσα στο σύστημα. Σε προφανές σημείο θα υπάρχει κουμπί το οποίο θα επιτρέπει στο χρήστη να επιστρέψει στο αρχικό παράθυρο.

Back Office:

Οι τεχνικές αλληλεπίδρασης που επέλεξα να χρησιμοποιήσω είναι φόρμες για εισαγωγή δεδομένων μέσα στη βάση και μενού επιλογής. Οι τεχνικές αυτές παρέχουν

ευκολία στο χρήστη , βελτιστοποίηση των λειτουργιών της κτηνοτροφικής μονάδας, αλλά ομοιόμορφη και λογική δομή του προγράμματος.

Με τη χρήση του μενού διευκολύνεται ο χρήστης στην επιλογή και εκτέλεση των διαφόρων λειτουργιών του συστήματος. Ανάλογα με τη σειρά των κουμπιών που επιλέγει ο χρήστης γίνεται και η σταδιακή εκτέλεση των λειτουργιών . Κάθε λειτουργία που επιλέγεται από τον χρήστη έχει επεξηγηματικό τίτλο.

Τα στοιχεία που αποθηκεύονται στη βάση του συστήματος θα εισάγονται μέσω φορμών εισαγωγής, για διευκόλυνση του χρήστη. Σε κάθε φόρμα θα υπάρχει σαφής και κατατοπιστικός τίτλος που θα υπενθυμίζει στον χρήστη για πιο αντικείμενο θα εισαχθούν τα στοιχεία . Επίσης έλεγχος των τιμών αν έχουν εισαχθεί σωστά.

2.3.1.2. Διεπιφάνεια υλικού

Το σύστημα και η βάση δεδομένων θα εγκατασταθούν σε ένα κεντρικό εξυπηρετητή. Επίσης, η διασύνδεση του συστήματος θα έχει την δυνατότητα σύνδεσης και με άλλα συστήματα σε περίπτωση που ο ιδιοκτήτης θέλει να επεκτείνει τις εγκαταστάσεις της κτηνοτροφικής του μονάδας. Να αναφέρουμε ότι η κτηνοτροφική μονάδα δεν έχει οποιοδήποτε υλικό όσο αφορά servers.

2.3.1.3. Διεπιφάνεια Λογισμικού

Προς το παρόν δεν χρειάζεται να υπάρχει κάποιος εγκαταστημένος browser στο σύστημα της κτηνοτροφικής μονάδας.

Η βάση δεδομένων του συστήματος θα αναπτυχθεί στη γλώσσα MySQL και για τη διεπαφή με το χρήστη θα χρησιμοποιήσουμε τη γλώσσα C-sharp.

2.3.1.4. Επικοινωνιακή διασύνδεση

Το σύστημα και η βάση δεδομένων θα εγκατασταθεί σε ένα κεντρικό εξυπηρετητή. Το ΒΟ θα είναι εγκαταστημένο στον κεντρικό εξυπηρετητή ο οποίος θα ενημερώνει τη βάση δεδομένων. Η σύνδεση με τον κεντρικό εξυπηρετητή θα γίνεται μέσω του connector της C-sharp.

2.3.2. Λειτουργίες Συστήματος

2.3.2.1. Λειτουργία 1: Διαχείριση Φάρμων

Σκοπός:

Λειτουργίες που αφορούν τη βάση του συστήματος και το λογισμικό της κτηνοτροφικής μονάδας έχοντας ως επίκεντρο την διαχείριση πληροφοριών σχετικά με τις φάρμες.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας :

Η λειτουργία αυτή θα ξεκινά με την είσοδο του χρήστη από την οθόνη των γενικών λειτουργιών του συστήματος , μέσω του πατήματος ενός κουμπιού. Η ανταπόκριση θα είναι η μεταφορά σε μια νέα οθόνη όπου ο χρήστης θα έχει τη δυνατότητα να επιλέξει μεταξύ διάφορων υπολειτουργιών που αφορούν τη διαχείριση των φαρμών πατώντας το ανάλογο κουμπί που θα τον μεταφέρει στην οθόνη της αντίστοιχης υπολειτουργίας.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή φάρμας**

Εισαγωγή:

Θα υπάρχει δυνατότητα εισαγωγής νέας φάρμας στο σύστημα της κτηνοτροφικής μονάδας.

Δεδομένα εισόδου:

Οι πληροφορίες που πρέπει να εισάγουμε για την νέα φάρμα είναι το όνομα, η διεύθυνση, το τηλέφωνο σταθερό και κινητό, ο αριθμός του fax ,το email και η άδεια λειτουργίας της φάρμας από την κυβέρνηση.

Επεξεργασία :

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της φάρμας.

Αποτελέσματα:

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της φάρμας στη βάση δεδομένων.

- **Διαγραφή φάρμας**

Εισαγωγή:

Θα υπάρχει δυνατότητα διαγραφής μιας φάρμας στο σύστημα.

Δεδομένα εισόδου:

Για την ανεύρεση μίας φάρμας για διαγραφή θα δίνεται ως δεδομένο εισόδου η ταυτότητα της φάρμας, αλλά θα υπάρχει και δυνατότητα αναζήτησης της με άλλο κριτήριο.

Επεξεργασία:

Εύρεση και διαγραφή της συγκεκριμένης φάρμας

Αποτελέσματα:

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτής της φάρμας από την βάση δεδομένων.

- **Ενημέρωση φάρμας**

Εισαγωγή

Θα υπάρχει δυνατότητα ενημέρωσης των στοιχείων της φάρμας στη βάση. Για παράδειγμα μπορεί να αλλάξει το όνομα της φάρμας καθώς και κάποια από τα στοιχεία επικοινωνίας.

Δεδομένα Εισόδου

Αρχικά θα γίνεται η ανεύρεση της συγκεκριμένης φάρμας και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή της φάρμας.

Επεξεργασία

Εύρεση και ενημέρωση των στοιχείων της συγκεκριμένης φάρμας

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων της φάρμας.

- **Αναζήτηση φάρμας**

Θα υπάρχει δυνατότητα αναζήτησης φάρμας από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση της φάρμας, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν την φάρμα, δηλαδή το όνομα ή το email ή η διεύθυνση της.

Επεξεργασία

Εύρεση της συγκεκριμένης φάρμας από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την φάρμα που πληροί τα κριτήρια της αναζήτησης.

2.3.2.2. Λειτουργία 2: Διαχείριση Προβάτων

Σκοπός :

Λειτουργίες που έχουν να κάνουν με τα στοιχεία των προβάτων και πως διασυνδέεται με το σύστημα.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας:

Ο χρήστης μέσω του ΒΟ θα επιλέγει να ξεκινά τη λειτουργία αυτή. Ο χρήστης θα μπορεί να μεταφέρεται σε νέο παράθυρο όπου θα μπορεί να διαλέξει μεταξύ διάφορων υπολειτουργιών, οι οποίες αφορούν τη διαχείριση των προβάτων. Για να μεταφερθεί στην αντίστοιχη υπολειτουργία θα πρέπει να πατά το ανάλογο κουμπί.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή προβάτων**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο πρόβατο στο σύστημα.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέο πρόβατο στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : Το όνομα του προβάτου, ο αριθμός του, η ημερομηνία γέννησης του, το φύλο του, ο γονότυπος του, το βάρος του, η κατανάλωση του σε τροφή, η φάρμα στην οποία ανήκει, η φωτογραφία του καθώς και οι αριθμοί της μητέρας και του πατέρα του.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι το νέο πρόβατο εισήχθη με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση προβάτων**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των προβάτων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των προβάτων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή όνομα του προβάτου, ο αριθμός του, η ημερομηνία γέννησης του, το φύλο του, ο γονότυπος του, το βάρος του, η κατανάλωση του σε τροφή, η φάρμα στην οποία ανήκει, η φωτογραφία του καθώς και οι αριθμοί της μητέρας και του πατέρα του.

Επεξεργασία:

Ενημέρωση των στοιχείων του προβάτου στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του προβάτου.

- **Διαγραφή προβάτων**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποια υπάρχον πρόβατα.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να αναζητήσει από μία λίστα το πρόβατο που θέλει να διαγράψει.

Επεξεργασία:

Με βάση το id του προβάτου που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του προβάτου αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση προβάτου**

Θα υπάρχει δυνατότητα αναζήτησης του προβάτου από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση του προβάτου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το πρόβατο, δηλαδή το όνομα ή ο αριθμός του.

Επεξεργασία

Εύρεση του συγκεκριμένου προβάτου από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το πρόβατο που πληροί τα κριτήρια της αναζήτησης.

2.3.2.3. Λειτουργία 3: Διαχείριση Ιδιοκτητών

Σκοπός:

Λειτουργίες που έχουν να κάνουν προσωπικά με τον ιδιοκτήτη και πως διασυνδέεται με το σύστημα.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας:

Ο χρήστης μέσω του ΒΟ θα επιλέγει να ξεκινά τη λειτουργία αυτή. Ο χρήστης θα μπορεί να μεταφέρεται σε νέο παράθυρο όπου θα μπορεί να διαλέξει μεταξύ διάφορων υπολειτουργιών, οι οποίες αφορούν τη διαχείριση των ιδιοκτητών. Για να μεταφερθεί στην αντίστοιχη υπολειτουργία θα πρέπει να πατά το ανάλογο κουμπί.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή ιδιοκτητών**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο ιδιοκτήτη στο σύστημα. Κάθε ιδιοκτήτης στο σύστημα θα πρέπει να έχει ένα δικό του username και ένα password. Έτσι όταν ο διαχειριστής θα εισάγει νέο ιδιοκτήτη, θα πρέπει να του αναθέτει το username του και το password του, καθώς και τα προνόμια που θα έχει πάνω στη βάση.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέο ιδιοκτήτη στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : Το ονοματεπώνυμο του ιδιοκτήτη, ο αριθμός ταυτότητας του, η ημερομηνία γέννησης του, τα στοιχεία επικοινωνίας, το username του, το password του και τα προνόμια που μπορεί να έχει.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι ο νέος ιδιοκτήτης εισήχθη με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση ιδιοκτητών**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των ιδιοκτητών

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των ιδιοκτητών θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή το ονοματεπώνυμο του ιδιοκτήτη, ο αριθμός ταυτότητας του, η ημερομηνία γέννησης του, τα στοιχεία επικοινωνίας, το username του, το password του και τα προνόμια που μπορεί να έχει.

Επεξεργασία:

Ενημέρωση των στοιχείων του ιδιοκτήτη στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του ιδιοκτήτη.

- **Διαγραφή ιδιοκτητών**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποιους υπάρχον ιδιοκτήτες.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα τον ιδιοκτήτη που θέλει να διαγράψει.

Επεξεργασία:

Με βάση τον ιδιοκτήτη που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του ιδιοκτήτη αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση ιδιοκτητών**

Θα υπάρχει δυνατότητα αναζήτησης του ιδιοκτήτη από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση του ιδιοκτήτη, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τον ιδιοκτήτη, δηλαδή το όνομα ή ο αριθμός ταυτότητας του.

Επεξεργασία

Εύρεση του συγκεκριμένου ιδιοκτήτη από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν τον ιδιοκτήτη που πληροί τα κριτήρια της αναζήτησης.

2.3.2.4. Λειτουργία 4: Διαχείριση Εργοδοτούμενων

Σκοπός :

Λειτουργίες που έχουν να κάνουν προσωπικά με τον εργοδοτούμενων και πως διασυνδέεται με το σύστημα.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας:

Ο χρήστης μέσω του ΒΟ θα επιλέγει να ξεκινά τη λειτουργία αυτή. Ο χρήστης θα μπορεί να μεταφέρεται σε νέο παράθυρο όπου θα μπορεί να διαλέξει μεταξύ διάφορων υπολειτουργιών, οι οποίες αφορούν τη διαχείριση των εργοδοτούμενων. Για να μεταφερθεί στην αντίστοιχη υπολειτουργία θα πρέπει να πατά το ανάλογο κουμπί.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή εργοδοτούμενων**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο εργοδοτούμενο στο σύστημα. Για κάποιους από τους εργοδοτούμενους στο σύστημα θα υπάρχει η δυνατότητα να έχουν το δικό τους username και ένα password. Έτσι όταν ο διαχειριστής θα εισάγει νέο εργοδοτούμενο, θα πρέπει να του αναθέτει το username του και το password του, καθώς και τα προνόμια που θα έχει πάνω στη βάση.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέο εργοδοτούμενο στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : Το ονοματεπώνυμο του εργοδοτούμενου, ο αριθμός ταυτότητας του, η ημερομηνία γέννησης του, τα στοιχεία επικοινωνίας, το username του, το password του και τα προνόμια που μπορεί να έχει.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι ο νέος εργοδοτούμενος εισήχθηκε με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση εργοδοτούμενων**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των εργοδοτούμενων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των εργοδοτούμενων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή το ονοματεπώνυμο του εργοδοτούμενου, ο αριθμός ταυτότητας του, η ημερομηνία γέννησης του, τα στοιχεία επικοινωνίας, το username του, το password του και τα προνόμια που μπορεί να έχει.

Επεξεργασία:

Ενημέρωση των στοιχείων του εργοδοτούμενου στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του εργοδοτούμενου.

- **Διαγραφή εργοδοτούμενων**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποιους υπάρχον εργοδοτούμενους.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα τον εργοδοτούμενο που θέλει να διαγράψει.

Επεξεργασία:

Με βάση τον εργοδοτούμενο που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του εργοδοτούμενου αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση εργοδοτούμενων**

Θα υπάρχει δυνατότητα αναζήτησης του εργοδοτούμενου από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση του εργοδοτούμενου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τον εργοδοτούμενο, δηλαδή το όνομα ή ο αριθμός ταυτότητας του.

Επεξεργασία

Εύρεση του συγκεκριμένου εργοδοτούμενου από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν τον εργοδοτούμενο που πληροί τα κριτήρια της αναζήτησης.

2.3.2.5. Λειτουργία 5: Διαχείριση Κτηνιάτρων

Σκοπός :

Λειτουργίες που έχουν να κάνουν προσωπικά με τον κτηνίατρο και πως διασυνδέεται με το σύστημα.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας:

Ο χρήστης μέσω του ΒΟ θα επιλέγει να ξεκινά τη λειτουργία αυτή. Ο χρήστης θα μπορεί να μεταφέρεται σε νέο παράθυρο όπου θα μπορεί να διαλέξει μεταξύ διάφορων υπολειτουργιών, οι οποίες αφορούν τη διαχείριση των κτηνιάτρων. Για να μεταφερθεί στην αντίστοιχη υπολειτουργία θα πρέπει να πατά το ανάλογο κουμπί.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή κτηνιάτρων**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο κτηνίατρο στο σύστημα.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέο κτηνίατρο στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : Το ονοματεπώνυμο του κτηνιάτρου, ο αριθμός ταυτότητας του, η ημερομηνία γέννησης του, τα στοιχεία επικοινωνίας, η διεύθυνση οικίας του και η διεύθυνση του ιατρείου του.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι ο νέος κτηνίατρος εισήχθηκε με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση κτηνιάτρων**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των κτηνιάτρων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των κτηνιάτρων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή ονοματεπώνυμο του κτηνιάτρου, ο αριθμός ταυτότητας του, η ημερομηνία γέννησης του, τα στοιχεία επικοινωνίας, η διεύθυνση οικίας του και η διεύθυνση του ιατρείου του.

Επεξεργασία:

Ενημέρωση των στοιχείων του κτηνιάτρου στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του κτηνιάτρου.

- **Διαγραφή κτηνιάτρων**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποιους υπάρχον κτηνιάτρους.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα τον κτηνίατρο που θέλει να διαγράψει.

Επεξεργασία:

Με βάση τον κτηνίατρο που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του κτηνιάτρου αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση κτηνιάτρων**

Θα υπάρχει δυνατότητα αναζήτησης του κτηνιάτρου από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση του κτηνιάτρου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τον κτηνίατρο, δηλαδή το όνομα ή ο αριθμός ταυτότητας του.

Επεξεργασία

Εύρεση του συγκεκριμένου κτηνιάτρου από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν τον κτηνίατρο που πληροί τα κριτήρια της αναζήτησης.

2.3.2.6. Λειτουργία 6: Διαχείριση Ιατρικών εξετάσεων για τα πρόβατα

Σκοπός:

Λειτουργίες που αφορούν τη βάση του συστήματος και το λογισμικό που θα δημιουργήσουμε για τη κτηνοτροφική μονάδα ως επίκεντρο την διαχείριση πληροφοριών με τις ιατρικές εξετάσεις των προβάτων.

Ενεργοποίηση/ανταπόκριση της λειτουργίας:

Η λειτουργία αυτή θα ξεκινά με την είσοδο του χρήστη από την οθόνη των γενικών λειτουργιών του συστήματος, μέσω του πατήματος ενός κουμπιού. Η ανταπόκριση θα είναι η μεταφορά σε μία νέα οθόνη όπου ο χρήστης θα έχει τη δυνατότητα να επιλέξει μεταξύ διάφορων υπολειτουργιών που αφορούν τη διαχείριση των ιατρικών εξετάσεων πατώντας το ανάλογο κουμπί που θα τον μεταφέρει στην οθόνη της αντίστοιχης υπολειτουργίας.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή εξέτασης**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει μία νέα εξέταση από κάποιο κτηνίατρο σε ένα πρόβατο στο σύστημα.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέα εξέταση στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : Την ημερομηνία που έγινε η εξέταση, το όνομα του κτηνίατρου ο οποίος έκαμε την εξέταση, τον αριθμό του προβάτου στο οποίο έγινε η εξέταση, τις ασθένειες που διαγνώστηκαν και τα φάρμακα που θα χορηγηθούν στο πρόβατο, καθώς και κάποια σχόλια από τον κτηνίατρο.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι η νέα εξέταση εισήχθη με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση εξέτασης**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των εξετάσεων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των εξετάσεων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή η ημερομηνία που έγινε η εξέταση, το όνομα του κτηνίατρου ο οποίος έκαμε την εξέταση, τον αριθμό του προβάτου στο οποίο έγινε η εξέταση, τις ασθένειες που διαγνώστηκαν και τα φάρμακα που θα χορηγηθούν στο πρόβατο, καθώς και κάποια σχόλια από τον κτηνίατρο.

Επεξεργασία:

Ενημέρωση των στοιχείων της εξέτασης στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία της εξέτασης.

- **Διαγραφή εξέτασης**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποιες υπάρχον εξετάσεις.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα την εξέταση που θέλει να διαγράψει.

Επεξεργασία:

Με βάση την εξέταση που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες της εξέτασης αυτής.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση εξέτασης**

Θα υπάρχει δυνατότητα αναζήτησης της εξέτασης από τη βάση δεδομένων και παρουσίαση των στοιχείων της σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση της εξέτασης, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν την εξέταση, δηλαδή η ημερομηνία διεξαγωγής της, το όνομα του κτηνιάτρου και ο αριθμός του προβάτου.

Επεξεργασία

Εύρεση της συγκεκριμένης εξέτασης από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την εξέταση που πληροί τα κριτήρια της αναζήτησης.

2.3.2.7. Λειτουργία 7: Διαχείριση Φαρμάκων που χορηγούνται στα πρόβατα

Σκοπός:

Λειτουργίες που αφορούν τη βάση του συστήματος και το λογισμικό που θα δημιουργήσουμε για τη κτηνοτροφική μονάδα ως επίκεντρο την διαχείριση πληροφοριών με τα φάρμακα που χορηγούνται στα πρόβατα.

Ενεργοποίηση/ανταπόκριση της λειτουργίας:

Η λειτουργία αυτή θα ξεκινά με την είσοδο του χρήστη από την οθόνη των γενικών λειτουργιών του συστήματος, μέσω του πατήματος ενός κουμπιού. Η ανταπόκριση θα είναι η μεταφορά σε μία νέα οθόνη όπου ο χρήστης θα έχει τη δυνατότητα να επιλέξει μεταξύ διάφορων υπολειτουργιών που αφορούν τη διαχείριση των φαρμάκων που χορηγούνται στα πρόβατα πατώντας το ανάλογο κουμπί που θα τον μεταφέρει στην οθόνη της αντίστοιχης υπολειτουργίας.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή φαρμάκου**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει ένα νέο φάρμακο.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει ένα νέο φάρμακο στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : το όνομα του φαρμάκου, μία περιγραφή/σχόλια του και την τιμή του.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι το νέο φάρμακο εισήχθηκε με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση φαρμάκου**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των φαρμάκων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των φαρμάκων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή το όνομα του φαρμάκου, μία περιγραφή/σχόλια του και την τιμή του.

Επεξεργασία:

Ενημέρωση των στοιχείων του φαρμάκου στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του φαρμάκου.

- **Διαγραφή φαρμάκου**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποια υπάρχον φάρμακα.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα το φάρμακο που θέλει να διαγράψει.

Επεξεργασία:

Με βάση το φάρμακο που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του φαρμάκου αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση φαρμάκου**

Θα υπάρχει δυνατότητα αναζήτησης του φαρμάκου από τη βάση δεδομένων και παρουσίαση των στοιχείων του σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση του φαρμάκου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το φάρμακο, δηλαδή το όνομα του φαρμάκου ή η περιγραφή του.

Επεξεργασία

Εύρεση του συγκεκριμένου φαρμάκου από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το φάρμακο που πληροί τα κριτήρια της αναζήτησης.

2.3.2.8. Λειτουργία 8: Διαχείριση Ασθενειών που διαγνώσθηκαν στα πρόβατα

Σκοπός:

Λειτουργίες που αφορούν τη βάση του συστήματος και το λογισμικό που θα δημιουργήσουμε για τη κτηνοτροφική μονάδα ως επίκεντρο την διαχείριση πληροφοριών με τις ασθένειες που διαγνώσθηκαν στα πρόβατα.

Ενεργοποίηση/ανταπόκριση της λειτουργίας:

Η λειτουργία αυτή θα ξεκινά με την είσοδο του χρήστη από την οθόνη των γενικών λειτουργιών του συστήματος, μέσω του πατήματος ενός κουμπιού. Η ανταπόκριση θα είναι η μεταφορά σε μία νέα οθόνη όπου ο χρήστης θα έχει τη δυνατότητα να επιλέξει μεταξύ διάφορων υπολειτουργιών που αφορούν τη διαχείριση των ασθενειών που διαγνώσθηκαν στα πρόβατα πατώντας το ανάλογο κουμπί που θα τον μεταφέρει στην οθόνη της αντίστοιχης υπολειτουργίας.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή ασθένειας**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει μία νέα ασθένεια.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει ένα μία νέα ασθένεια στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : το όνομα της ασθένειας και μία περιγραφή/σχόλια της. Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι η νέα ασθένεια εισήχθηκε με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση ασθένειας**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των ασθενειών.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των ασθενειών θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή το όνομα της ασθένειας και μία περιγραφή/σχόλια της. Επεξεργασία:

Ενημέρωση των στοιχείων της ασθένειας στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία της ασθένειας.

- **Διαγραφή ασθένειας**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποια υπάρχον ασθένεια.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα την ασθένεια που θέλει να διαγράψει.

Επεξεργασία:

Με βάση την ασθένεια που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες της ασθένειας αυτής.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

- **Αναζήτηση ασθένειας**

Θα υπάρχει δυνατότητα αναζήτησης της ασθένειας από τη βάση δεδομένων και παρουσίαση των στοιχείων του σε μορφή Report.

Δεδομένα Εισόδου

Για την αναζήτηση της ασθένειας, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν την ασθένεια, δηλαδή το όνομα της ασθένειας ή η περιγραφή της.

Επεξεργασία

Εύρεση της συγκεκριμένου ασθένειας από τη βάση.

Αποτελέσματα

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την ασθένεια που πληροί τα κριτήρια της αναζήτησης.

2.3.2.9. Λειτουργία 9: Διαχείριση Εσόδων Φάρμας

Σκοπός:

Λειτουργίες που έχουν να κάνουν με τα έσοδα της φάρμας και πως αυτά διασυνδέονται με το σύστημα.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας:

Ο χρήστης μέσω του ΒΟ θα επιλέγει να ξεκινά τη λειτουργία αυτή. Ο χρήστης θα μπορεί να μεταφέρεται σε νέο παράθυρο όπου θα μπορεί να διαλέξει μεταξύ διάφορων υπολειτουργιών, οι οποίες αφορούν τη διαχείριση των εσόδων της φάρμας. Για να μεταφερθεί στην αντίστοιχη υπολειτουργία θα πρέπει να πατά το ανάλογο κουμπί.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή εσόδων**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο έσοδο στο σύστημα.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέο έσοδο στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : την φάρμα στην οποία αναφέρεται το έσοδο, την ημερομηνία, τον κωδικό και την περιγραφή του εσόδου, την μονάδα μέτρησης του εσόδου, και την τιμή ανά μονάδα μέτρησης του εσόδου.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι το νέο έσοδο εισήχθηκε με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση εσόδων**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των εσόδων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των εσόδων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή την φάρμα στην οποία αναφέρεται το έσοδο, την ημερομηνία, τον κωδικό και την περιγραφή του εσόδου, την μονάδα μέτρησης του εσόδου, και την τιμή ανά μονάδα μέτρησης του εσόδου.

Επεξεργασία:

Ενημέρωση των στοιχείων του εσόδου στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του εσόδου.

- **Διαγραφή εσόδων**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποια από τα υπάρχον έσοδα.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα το έσοδο που θέλει να διαγράψει.

Επεξεργασία:

Με βάση το έσοδο που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του εσόδου αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

2.3.2.10. Λειτουργία 10: Διαχείριση Εξόδων Φάρμας

Σκοπός :

Λειτουργίες που έχουν να κάνουν με τα έξοδα της φάρμας και πως αυτά διασυνδέονται με το σύστημα.

Ενεργοποίηση/Ανταπόκριση της λειτουργίας:

Ο χρήστης μέσω του ΒΟ θα επιλέγει να ξεκινά τη λειτουργία αυτή. Ο χρήστης θα μπορεί να μεταφέρεται σε νέο παράθυρο όπου θα μπορεί να διαλέξει μεταξύ διάφορων υπολειτουργιών, οι οποίες αφορούν τη διαχείριση των εξόδων της φάρμας. Για να μεταφερθεί στην αντίστοιχη υπολειτουργία θα πρέπει να πατά το ανάλογο κουμπί.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή εξόδων**

Εισαγωγή:

Ο διαχειριστής έχει τη δυνατότητα να εισάγει νέο έξοδο στο σύστημα.

Δεδομένα Εισόδου:

Ο διαχειριστής για να εισάγει νέο έξοδο στο σύστημα θα πρέπει να εισάγει τα ακόλουθα από το πληκτρολόγιο : την φάρμα στην οποία αναφέρεται το έξοδο, την ημερομηνία, τον κωδικό και την περιγραφή του εξόδου, την μονάδα μέτρησης του

εξόδου, την τιμή ανά μονάδα μέτρησης του εξόδου και αν το έξοδο ανήκει στα τρέχοντα ή στα σταθερά έξοδα.

Επεξεργασία:

Ανάγνωση των δεδομένων εισόδου, επικύρωση τους και αποθήκευση τους στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα ότι το νέο έξοδο εισήχθηκε με επιτυχία ή μήνυμα εσφαλμένης εισαγωγής σε περίπτωση λάθους.

- **Ενημέρωση εξόδων**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να κάνει αλλαγές πάνω στα στοιχεία των εξόδων

Δεδομένα εισόδου:

Τα δεδομένα εισόδου για την ενημέρωση των εξόδων θα μπορούν να είναι όλα τα στοιχεία του που θα αλλαχτούν. Δηλαδή την φάρμα στην οποία αναφέρεται το έξοδο, την ημερομηνία, τον κωδικό και την περιγραφή του εξόδου, την μονάδα μέτρησης του εξόδου, την τιμή ανά μονάδα μέτρησης του εξόδου και αν το έξοδο ανήκει στα τρέχοντα ή στα σταθερά έξοδα.

Επεξεργασία:

Ενημέρωση των στοιχείων του εξόδου στη βάση δεδομένων.

Αποτελέσματα:

Μήνυμα στην οθόνη που βεβαιώνει ότι η αλλαγή έγινε σωστά στα στοιχεία του εξόδου.

- **Διαγραφή εξόδων**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να διαγράψει κάποια από τα υπάρχον έξοδα.

Δεδομένα εισόδου:

Ο διαχειριστής θα μπορεί να επιλέξει από μία λίστα το έξοδο που θέλει να διαγράψει.

Επεξεργασία:

Με βάση το έξοδο που επιλέχθηκε, θα διαγράφονται από τη βάση δεδομένων οι πληροφορίες του εξόδου αυτού.

Αποτελέσματα:

Μήνυμα για να επιβεβαιώσει ο διαχειριστής την διαγραφή που πρόκειται να γίνει και στη συνέχεια εμφάνιση δεύτερου μηνύματος ότι η διαγραφή ολοκληρώθηκε.

2.3.2.11. Λειτουργία 11: Διαχείριση τιμολογίων

Σκοπός:

Ο υπολογισμός και εκτύπωση του τιμολογίου για να δοθεί στον πελάτη.

Ενεργοποίηση/ανταπόκριση της λειτουργίας:

Η λειτουργία θα ξεκινά με την είσοδο του χρήστη από την οθόνη λειτουργίας τιμολογίων του συστήματος. Η ανταπόκριση θα είναι η μεταφορά σε μια νέα οθόνη όπου ο χρήστης θα έχει την δυνατότητα να εισάγει το κόστος που στοιχίζει το έσοδο/έξοδο για το συγκεκριμένο πελάτη. Ακολούθως με το πάτημα ενός κουμπιού θα μπορεί να εκτυπώνεται το τιμολόγιο.

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή Τιμολογίου**

Εισαγωγή:

Θα υπάρχει η δυνατότητα δημιουργίας νέου τιμολογίου από το σύστημα όταν ο πελάτης εισέλθει στην κτηνοτροφική μονάδα.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου θα είναι η εισαγωγή του ποσού για το οποίο κοστίζει το έσοδο/έξοδο, και τα στοιχεία του πελάτη που θα πραγματοποιήσει την πληρωμή.

Επεξεργασία:

Η επεξεργασία που θα γίνει στο σύστημα είναι η εισαγωγή στη βάση δεδομένων του νέου τιμολογίου.

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα μας επιβεβαιώνει την επιτυχή εισαγωγή του νέου τιμολογίου στη βάση.

- **Εκτύπωση Τιμολογίου**

Εισαγωγή:

Μέσα από αυτή τη λειτουργία θα υπάρχει η δυνατότητα εκτύπωσης του τιμολογίου από το σύστημα όταν ο πελάτης εισέλθει στην κτηνοτροφική μονάδα.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου θα είναι η εισαγωγή του ποσού για το οποίο κοστίζει το έσοδο/έξοδο, και τα στοιχεία του πελάτη που θα πραγματοποιήσει την πληρωμή.

Επεξεργασία:

Η επεξεργασία σε αυτή τη λειτουργία είναι η εκτύπωση του νέου τιμολογίου.

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που να επιβεβαιώνει την εκτύπωση του νέου τιμολογίου.

○ **Αναζήτηση Τιμολογίου**

Εισαγωγή:

Μέσα από αυτή τη λειτουργία θα μπορεί να γίνεται αναζήτηση κάποιου συγκεκριμένου τιμολογίου με βάση τον αριθμό του εσόδου/εξόδου, και το όνομα του πελάτη στον οποίο αντιστοιχεί το έσοδο/έξοδο.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου σε αυτή την περίπτωση θα είναι η επιλογή των στοιχείων που περιγράφουν το τιμολόγιο.

Επεξεργασία:

Θα γίνεται μια επεξεργασία που στην συγκεκριμένη περίπτωση θα είναι αναζήτηση

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη των αποτελεσμάτων της αναζήτησης.

○ **Διαγραφή Τιμολογίου**

Εισαγωγή:

Μέσα από αυτή τη λειτουργία θα μπορεί να γίνεται διαγραφή κάποιου συγκεκριμένου τιμολογίου με βάση τον αριθμό του εσόδου/εξόδου, και το όνομα του πελάτη στον οποίο αντιστοιχεί το έσοδο/έξοδο.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου σε αυτή την περίπτωση θα είναι η επιλογή των στοιχείων που περιγράφουν το τιμολόγιο.

Επεξεργασία:

Θα γίνεται μια επεξεργασία που στην συγκεκριμένη περίπτωση θα είναι διαγραφή

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη των αποτελεσμάτων της διαγραφής.

2.3.2.12. Λειτουργία 12: Διαχείριση αποδείξεων

Σκοπός:

Η εκτύπωση απόδειξης της πληρωμής των πελατών.

Ενεργοποίηση/ανταπόκριση της λειτουργίας:

Η λειτουργία θα ξεκινά με την είσοδο του χρήστη από την οθόνη λειτουργίας πληρωμής του συστήματος. Η ανταπόκριση θα είναι η μεταφορά σε μια νέα οθόνη όπου ο χρήστης θα έχει την δυνατότητα να επιλέξει το ποσό το οποίο πρόκειται να πληρώσει ο πελάτης. Ακολούθως με το πάτημα ενός κουμπιού θα μπορεί να εκτυπώνεται η απόδειξη για την συναλλαγή.

Σχετικές λειτουργικές απαιτήσεις:

○ **Εισαγωγή Απόδειξης**

Εισαγωγή:

Θα υπάρχει η δυνατότητα δημιουργίας νέας απόδειξης από το σύστημα όταν ολοκληρωθεί μια πληρωμή.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου θα είναι η επιλογή του εσόδου/εξόδου για τον οποίο θα γίνει η πληρωμή, και τα στοιχεία του πελάτη που θα πραγματοποιήσει την πληρωμή.

Επεξεργασία:

Η επεξεργασία που θα γίνει στο σύστημα είναι η εισαγωγή στη βάση δεδομένων της νέας απόδειξης.

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα μας επιβεβαιώνει την επιτυχή εισαγωγή της νέας απόδειξης στη βάση.

○ **Εκτύπωση Απόδειξης**

Εισαγωγή:

Μέσα από αυτή τη λειτουργία θα υπάρχει η δυνατότητα εκτύπωσης της απόδειξης από το σύστημα όταν ολοκληρωθεί μια πληρωμή.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου θα είναι η επιλογή των πληρωμών και τα στοιχεία του πελάτη.

Επεξεργασία:

Η επεξεργασία σε αυτή τη λειτουργία είναι η εκτύπωση της νέας απόδειξης.

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που να μας επιβεβαιώνει την εκτύπωση της νέας απόδειξης.

○ **Αναζήτηση Απόδειξης**

Εισαγωγή:

Μέσα από αυτή τη λειτουργία θα μπορεί να γίνεται αναζήτηση κάποιας συγκεκριμένης απόδειξης με βάση τον αριθμό της απόδειξης, και το όνομα του πελάτη στον οποίο αντιστοιχεί η απόδειξη.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου σε αυτή την περίπτωση θα είναι η επιλογή των στοιχείων που περιγράφουν την απόδειξη.

Επεξεργασία:

Θα γίνεται μια επεξεργασία που στην συγκεκριμένη περίπτωση θα είναι αναζήτηση

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη των αποτελεσμάτων της αναζήτησης.

○ **Διαγραφή Απόδειξης**

Εισαγωγή:

Μέσα από αυτή τη λειτουργία θα μπορεί να γίνεται διαγραφή κάποιας συγκεκριμένης απόδειξης με βάση τον αριθμό της απόδειξης, και το όνομα του πελάτη στον οποίο αντιστοιχεί η απόδειξη.

Δεδομένα εισόδου:

Τα δεδομένα εισόδου σε αυτή την περίπτωση θα είναι η επιλογή των στοιχείων που περιγράφουν την απόδειξη.

Επεξεργασία:

Θα γίνεται μια επεξεργασία που στην συγκεκριμένη περίπτωση θα είναι διαγραφή

Αποτελέσματα:

Το αποτέλεσμα θα είναι η εκτύπωση στην οθόνη των αποτελεσμάτων της διαγραφής.

2.3.2.13. Λειτουργία 13: Διαχείριση LOG FILE

Σκοπός:

Ενεργοποίηση/Ανταπόκριση της λειτουργίας :

Σχετικές λειτουργικές απαιτήσεις:

- **Εισαγωγή εγγραφής στο LOG FILE**

Εισαγωγή:

Κάθε φορά που ένας χρήστης κάνει μία αλλαγή που να επηρεάζει τη βάση του ΒΟ συστήματος, τότε το LOG FILE θα ενημερώνεται αποθηκεύοντας μία νέα εγγραφή σε αυτό, η οποία θα αντιπροσωπεύει την αλλαγή που έγινε. Οι αλλαγές αυτές αφορούν την εισαγωγή κάποιας νέας εγγραφής, τη διαγραφή της και την ενημέρωση της.

Δεδομένα εισόδου:

Η λειτουργία αυτή θα εκτελείται αυτόματα κάθε φορά που ο χρήστης κάνει μία αλλαγή στο σύστημα. Σαν είσοδο θα παίρνει ένα αύξον αριθμό του χρήστη, την ημερομηνία σύνδεσης, την ώρα σύνδεσης, τη διάρκεια σύνδεσης καθώς και την ενέργεια που εκτελέστηκε.

Επεξεργασία:

Εισαγωγή στη βάση μίας νέας εγγραφής LOG FILE με το δικό της αύξον αριθμό.

Αποτελέσματα:

Η λειτουργία αυτή δεν είναι ορατή προς το χρήστη. Έτσι δεν θα εμφανίζεται κάποιο μήνυμα στην οθόνη. Οι αλλαγές γίνονται μόνο στο ΒΟ.

- **Προβολή LOG FILE**

Εισαγωγή:

Ο διαχειριστής του συστήματος θα μπορεί να βλέπει το LOG FILE και να πληροφορείται ποιες αλλαγές έγιναν από το σύστημα και από ποιους. Έτσι σε περίπτωση σφάλματος ή χάσιμο δεδομένων, τα δεδομένα θα μπορούν να ανακτηθούν.

Δεδομένα εισόδου:

Η μόνη είσοδος είναι το πάτημα του κουμπιού για το LOG FILE.

Επεξεργασία:

Φόρτωση LOG FILE

Αποτελέσματα:

Εμφάνιση του LOG FILE στην οθόνη.

- **Προβολή πιο πρόσφατων εγγράφων LOG FILE που έχουν σχέση με το λογισμικό διαχείρισης της ΒΔ.**

Εισαγωγή:

Ο διαχειριστής κατά την είσοδο του στο ΒΟ θα μπορεί να βλέπει τις πιο πρόσφατες εγγραφές που ενημερώθηκαν, και οι οποίες έγιναν από τη τελευταία φορά που είχε κάνει login στο ΒΟ.

Δεδομένα εισόδου:

Η λειτουργία αυτή θα εκτελείται αυτόματα κάθε φορά που ο διαχειριστής κάνει login στο ΒΟ. Σαν είσοδο θα παίρνει την ημερομηνία σύνδεσης και την ώρα σύνδεσης, καθώς και την ημερομηνία που έγινε το τελευταίο logout του διαχειριστή. Επίσης σαν είσοδο θα παίρνει κάποιο flag ώστε να επιλέγονται οι κατάλληλες εγγραφές του log file για να παρουσιάζονται στο ΒΟ.

Επεξεργασία:

Επιλογή των κατάλληλων εγγράφων του log file με βάση τα κριτήρια των δεδομένων εισόδου.

Αποτελέσματα:

Εμφάνιση των εγγράφων του log file που έχουν επιλεγεί στην οθόνη.

- **Καθαρισμός LOG FILE**

Εισαγωγή:

Ο διαχειριστής θα μπορεί να σβήνει πληροφορίες από το log file που είναι πλέον αχρείαστες. Αυτό βοηθάει σε θέματα ασφαλείας, αλλά και σε θέματα μνήμης, αφού απελευθερώνουμε χώρο από το log file.

Δεδομένα εισόδου:

Ο διαχειριστής του συστήματος θα έχει την ικανότητα να επιλέγει με κάποιο τρόπο τις παλιές και αχρείαστες εγγραφές και να μπορεί να τις διαγράψει. Επίσης θα μπορεί να διαγράψει όλες τις εγγραφές ή μία ομάδα αυτών. Ακόμα και ολική εκκαθάριση του log file.

Επεξεργασία:

Διαγραφή όλων των εγγραφών από το log file που επέλεξε ο χρήστης να διαγράψει.

Αποτελέσματα:

Μήνυμα επιβεβαίωσης ότι οι επιλεγμένες εγγραφές έχουν διαγραφεί επιτυχώς από το log file, ή ότι έχει καθαρίσει ολόκληρο το log file.

2.3.3. Απαιτήσεις Απόδοσης

Για να μπορέσει το πρόγραμμα να είναι πιο αποδοτικό θα πρέπει να προσπαθήσουμε να είναι τέτοιο ώστε να είναι χαμηλός ο χρόνος απόκρισης της Βάσης Δεδομένων. Με αυτό το τρόπο το πρόγραμμα θα είναι πιο αποδοτικό.

2.3.4. Χαρακτηριστικά συστήματος λογισμικού

2.3.4.1. Αξιοπιστία Πληροφοριών συστήματος

Για να έχουμε αξιοπιστία στο σύστημα θα πρέπει να γίνεται συστηματικός έλεγχος σε όλα τα δεδομένα που θα εισάγονται στο σύστημα, και πριν καταχωρηθούν στη βάση. Με αυτό το τρόπο θα καταφέρουμε τις λανθασμένες εισαγωγές στη βάση και εξασφαλίζουμε την αξιοπιστία των δεδομένων μας.

Για να περιορίσουμε τις επιλογές κατά την εισαγωγή μπορούμε να χρησιμοποιήσουμε μηχανισμούς που αφορούν την διεπαφή τους συστήματος.

Για παράδειγμα σε περίπτωση που ο χρήστης επιχειρήσει να εισάγει τον γονότυπο του προβάτου, θα μπορεί να επιλέξει τον γονότυπο μέσα από μια λίστα που θα εμφανίζεται.

2.3.4.2. Διαθεσιμότητα Πληροφορίας

Για να καταφέρουμε να τηρήσουμε την διασφάλιση της διαθεσιμότητας της πληροφορίας θα μπορούσαμε σε περίπτωση σφάλματος ή κατάρρευσης του συστήματος τα δεδομένα και οι πληροφορίες μας να φυλάγονται σε εφεδρικά αντίγραφα ασφαλείας. Έτσι με αυτό το τρόπο διασφαλίζεται η γρήγορη αποκατάσταση της κανονικής λειτουργίας του συστήματος.

2.3.4.3. Ασφάλεια

Μέσα στο σύστημα που θα δημιουργήσουμε θα αποθηκεύονται πληροφορίες οι οποίες θα είναι εμπιστευτικές. Με βάση αυτή την παράμετρο επιβάλλεται η χρήση μηχανισμών ασφαλείας. Με αυτό το τρόπο μειώνουμε τις πιθανότητες υποκλοπής. Για να πετύχουμε να φτιάξουμε μηχανισμό ασφαλείας θα πρέπει να έχουμε μια ελεγχόμενη πρόσβαση στο σύστημα. Ο ιδιοκτήτης της φάρμας θα έχει ένα username password με τα οποία θα του επιτρέπεται η είσοδος στο σύστημα. Επίσης θα υπάρχει ένα log file πάνω στο οποίο θα φυλάγονται όλες οι αλλαγές που γίνονται στο σύστημα καθώς και η τελευταία μέρα αλλαγής δεδομένων, και ημερομηνία της τελευταίας εισόδου στο σύστημα. Έτσι θα διασφαλίζεται η ακεραιότητα των πληροφοριών που θα φυλάγονται στο σύστημα.

2.3.4.4. Συντήρηση

Η συντήρηση του συστήματος είναι μια διαδικασία η οποία είναι η δυσκολότερη στην παραγωγή λογισμικού, και θα προσπαθήσω να επιτύχω όσο πιο απλή σχεδίαση μπορώ.

2.3.5. Λογικές Απαιτήσεις Βάσης Δεδομένων

Η Βάση Δεδομένων τους συστήματος θα διαχειρίζεται πληροφορίες τύπου κειμένων και φωτογραφίες(προβάτων). Μέσα στην βάση θα μπορούμε να κάνουμε εισαγωγές, ενημερώσεις, διαγραφές και αναζητήσεις. Την ολική πρόσβαση στο σύστημα θα έχει ο ιδιοκτήτης της κτηνοτροφικής μονάδας.

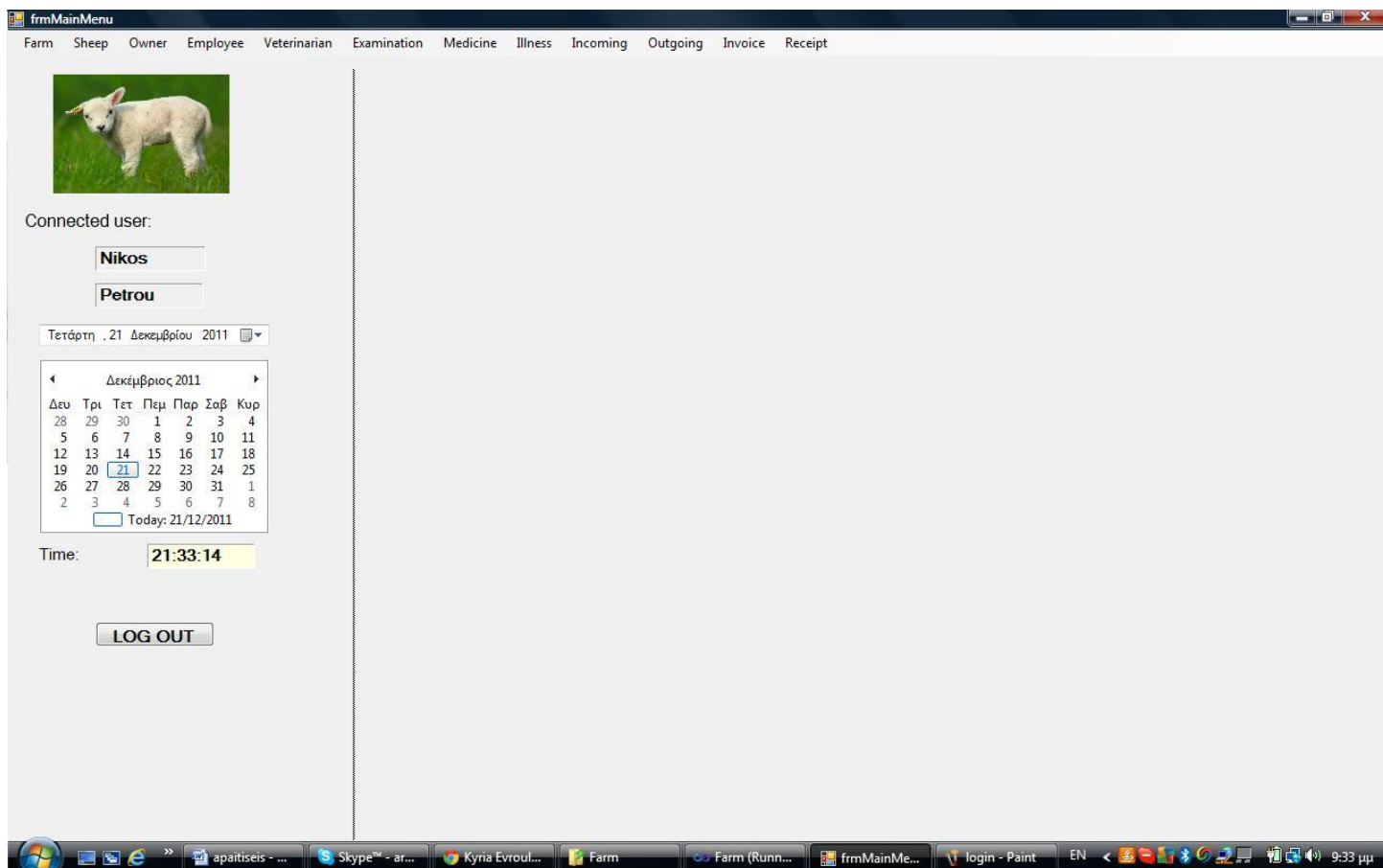
Σε αυτό το σύστημα θα μπορεί ο ιδιοκτήτης να έχει ανά πάσα στιγμή πρόσβαση.

ΓΡΗΓΟΡΟ ΠΡΟΤΟΤΥΠΟ

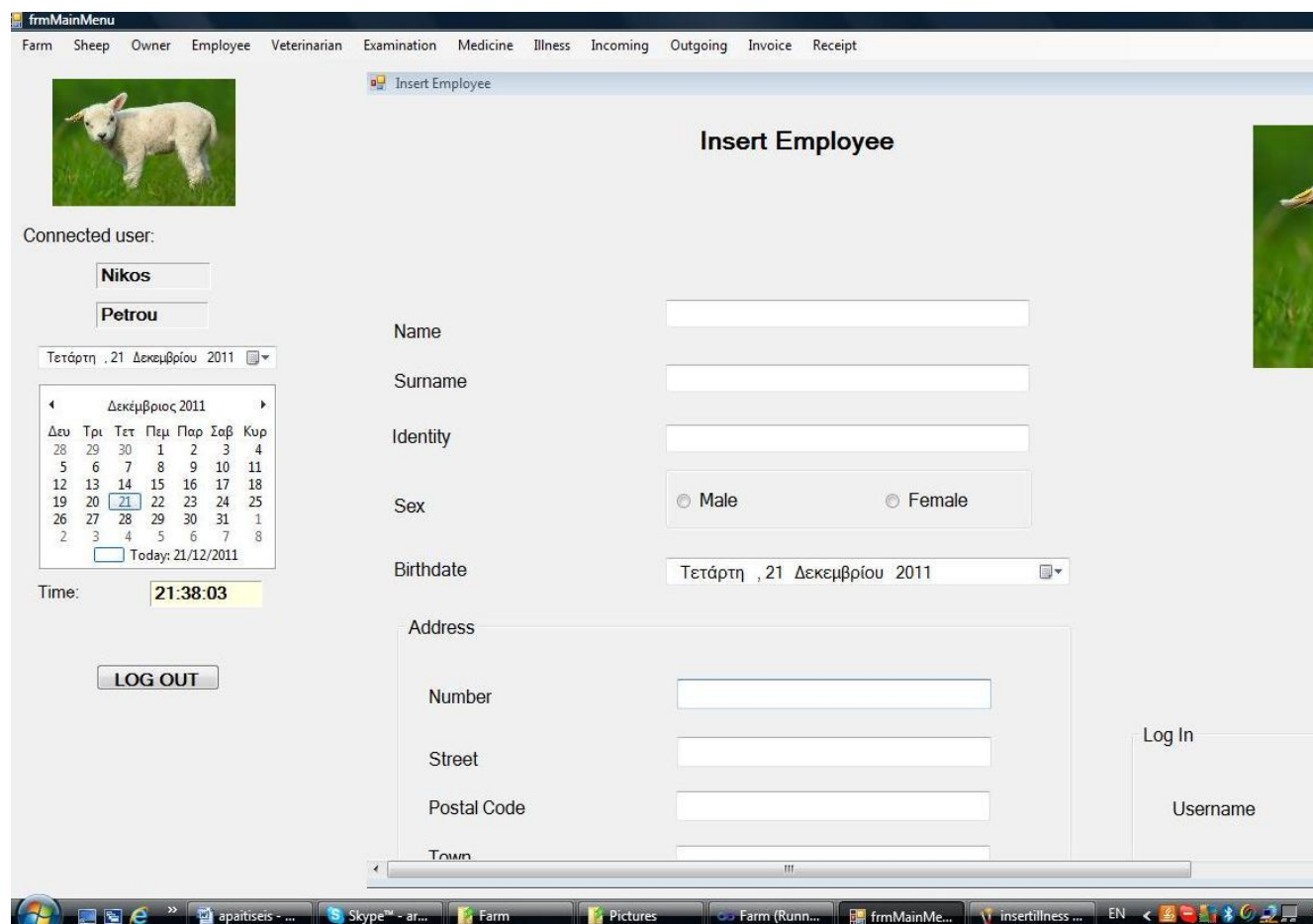


The screenshot shows a Windows-style window titled "frmLogIn". Inside the window, the word "FARM" is centered at the top. Below it, there are two input fields: "Username:" and "Password:". To the right of the "Username:" field is a small image of a white lamb standing in green grass. At the bottom of the window, there are three buttons: "LOG IN" on the left, "REGISTER NOW!" in the middle-right, and "EXIT" on the far right.

Εικόνα 2.1



Εικόνα 2.2



Εικόνα 2.3

frmMainMenu

Farm Sheep Owner Employee Veterinarian Examination Medicine Illness Incoming Outgoing Invoice Receipt

Insert Farm

Insert Farm

Connected user:

Nikos

Petrou

Τετάρτη, 21 Δεκεμβρίου 2011

Δεκέμβριος 2011

Δευ	Τρι	Τετ	Πεμ	Παρ	Σαβ	Κυρ
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Today: 21/12/2011

Time: **21:36:03**

LOG OUT

Name

Address

Number

Street

Postal code

Town

Country

Phone number

Mobile number

Fax

Email

Government License

Εικόνα 2.4

Κεφάλαιο 3. Ανάλυση Προδιαγραφών

Κεφάλαιο 3. Ανάλυση Προδιαγραφών	50
3.1 Εισαγωγή	50
3.2 Τεχνική Δομημένης Ανάλυσης Συστήματος	51
3.2.1 Δημιουργία ΔΡΔ	52
3.2.2 Μηχανογράφηση Τμημάτων.....	83
3.2.3 Εκλέπτυνση Ροών Δεδομένων	84
3.2.4 Προσδιορισμός Λογικής των διαδικασιών	100
3.2.5 Προσδιορισμός Αποθήκευσης Δεδομένων	103
3.2.6 Καθορισμός φυσικών πόρων	105
3.2.7 Ορισμός προδιαγραφών για Είσοδο/Εξοδο δεδομένων	112
3.2.8 Εκτίμηση μεγέθους	119
3.2.9 Απαιτήσεις σε hardware	120
3.3 Βήματα Αντικειμενοστραφούς Ανάλυσης.....	121
3.3.1 Εισαγωγή	121
3.4 Διαγράμματα Οντοτήτων.....	131

Κεφάλαιο 3. Ανάλυση Προδιαγραφών

3.1 Εισαγωγή

Σκοπός του εγγράφου

Σε αυτό το σημείο θα παρουσιάσω τις προδιαγραφές του συστήματος με βάση τις απαιτήσεις που ορίσαμε στο προηγούμενο στάδιο της φάσης ανάπτυξης του συστήματος. Το έγγραφο αυτό θεωρείται επίσημο έγγραφο και αποτελεί συμβόλαιο μεταξύ του ιδιοκτήτη της κτηνοτροφικής μονάδας και της ομάδας ανάπτυξης λογισμικού. Με το έγγραφο αυτό αποσκοπώ στην καλύτερη διατύπωση και ξεκάθαρη διαπίστωση των λειτουργιών που κλήθηκα υλοποιήσω. Επίσης το έγγραφο αυτό απευθύνεται προς τον πελάτη και προς το άτομο που αναλαμβάνει την ανάπτυξη του λογισμικού. Επιπρόσθετα το έγγραφο αυτό θα αποτελεί και τεκμηρίωση μετέπειτα για το στάδιο της συντήρησης.

Συντομογραφίες που έχω χρησιμοποιήσει:

ΒΔ -> Βάση δεδομένων

ΒΟ -> Back office System

ΦΟ -> Front Office System

Χρήστης -> ιδιοκτήτης κτηνοτροφικής μονάδας

Πηγές:

- Net beans(χρησιμοποιήθηκε για την δημιουργία του γρήγορου πρωτότυπου)
- Σημειώσεις μαθήματος «Τεχνολογία Λογισμικού» Χειμερινό εξάμηνο 2010-2011
- Σημειώσεις μαθήματος «Ομαδική Εργασία Τεχνολογίας Λογισμικού» Εαρινό εξάμηνο 2010-2011
- “Βασικές Αρχές Τεχνολογίας Λογισμικού”, Ι. Sommerville, Εκδ. Κλειδάριθμος, 8η έκδοση, Μάιος 2008.

Περιεχόμενα εγγράφου ανά τμήματα:

- Βήματα δομημένης ανάλυσης συστήματος .
- Βήματα αντικειμενοστραφούς ανάλυσης.
- Πλάνο διαχείρισης του έργου.
- Διαγράμματα οντοτήτων συσχετίσεων

- Gantt chart για το χρονοπρογραμματισμό των εργασιών για όλη τη διάρκεια της ανάπτυξης του συστήματος.

3.2 Τεχνική Δομημένης Ανάλυσης Συστήματος

Η τεχνική που έχει ακολουθηθεί για τη σύνταξη των προδιαγραφών ακολουθεί τα εξής βήματα(Gane Sarnsen):

1. Δημιουργία διαγραμμάτων ροής δεδομένων τα οποία θα αντιπροσωπεύουν τη λογική ροή των δεδομένων μέσα στο σύστημα.
2. Καταγραφή των αποφάσεων σχετικά με το ποια τμήματα θα μηχανογραφηθούν ή όχι. Οι αποφάσεις αυτές έχουν αιτιολογηθεί και επίσης έχει περιγραφεί πως θα μηχανογραφηθεί το κάθε τμήμα.
3. Εκλέπτυνση των ροών δεδομένων, στις οποίες καταγράφονται αναλυτικά τα στοιχεία που περνούν σε κάθε ροή. Επίσης γίνεται ανάλυση των στοιχείων αυτών μέχρι να παρουσιαστούν όλα με πλήρη λεπτομέρεια.
4. Ανάλυση κάθε διαδικασίας που υπάρχει στο διάγραμμα ροής δεδομένων και συγκεκριμένα ανάλυση του προσδιορισμού της λογικής της κάθε διεργασίας.
5. Προσδιορισμός αποθήκευσης δεδομένων στον οποίο ορίζονται τα ακριβή περιεχόμενα κάθε αποθήκευσης και η μορφή τους. Επίσης για το που απαιτείται άμεση πρόσβαση
6. Καθορισμός φυσικών πόρων που θα χρησιμοποιηθούν για την εφαρμογή και αναλύονται θέματα όπως το όνομα του κάθε αρχείου, η οργάνωση του καθώς και τα μέσα αποθήκευσης.
7. Ορισμός προδιαγραφών για εισόδους/εξόδους.
8. Εκτίμηση μεγέθους
9. Καθορισμός απαιτήσεων σε υλικό

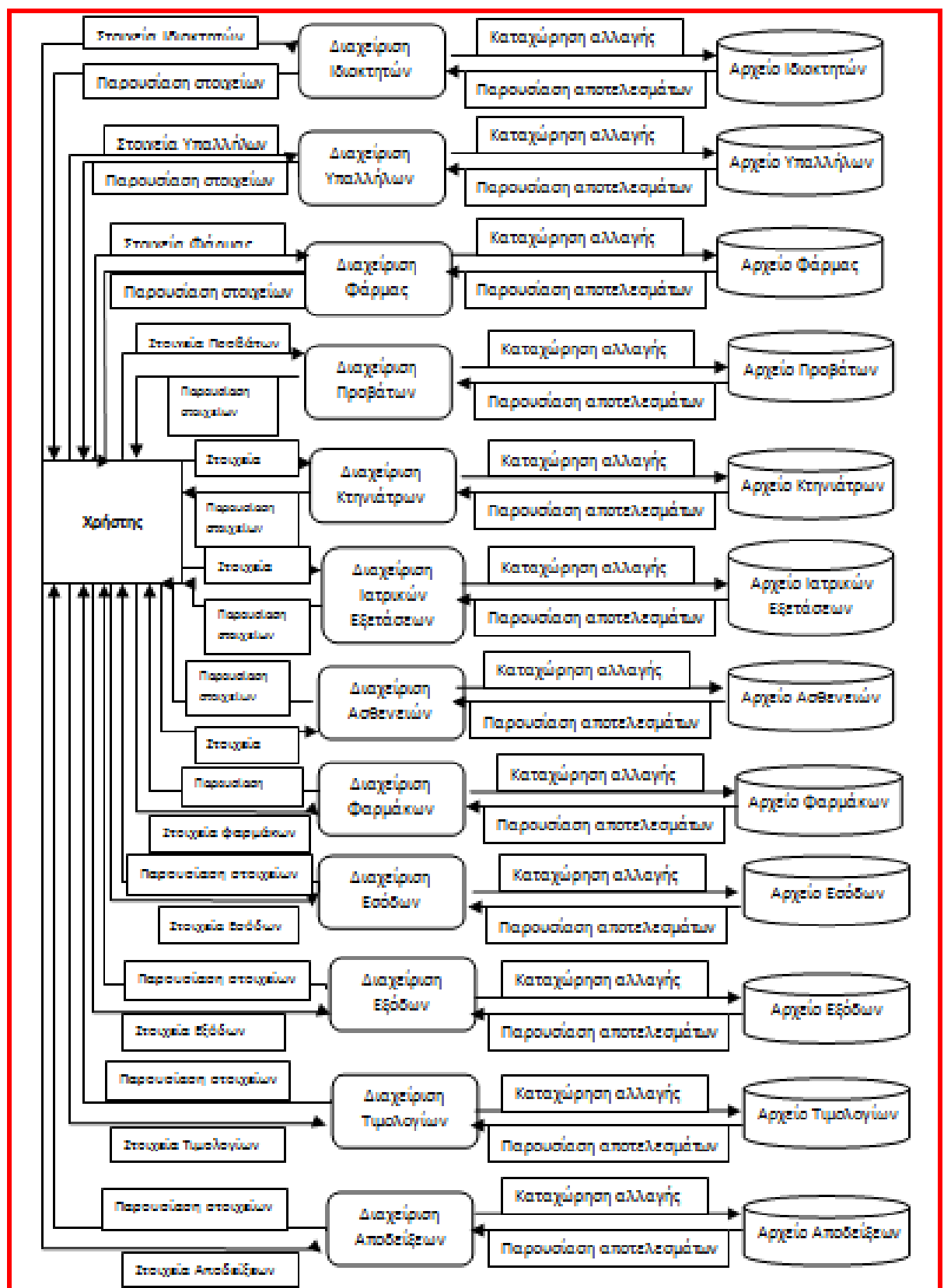
3.2.1 Δημιουργία ΔΡΔ

3.2.1.1 Δομικά στοιχεία ενός ΔΡΔ

Τα διαγράμματα ροής δεδομένων εκφράζουν τη λογική ροή δεδομένων του συστήματος, δηλαδή την αλληλεπίδραση που έχουν μεταξύ τους οι εξωτερικοί πράκτορες, οι διαδικασίες και οι αποθηκευτικές μονάδες. Αυτά είναι και τα δομικά στοιχεία του ΔΡΔ.

Ο εξωτερικός πράκτορας αναπαριστάται από ένα τετράγωνο με το όνομα του μέσα σε αυτό. Οι διαδικασίες με ένα στρογγυλοποιημένο ορθογώνιο με το όνομα τις διαδικασίας μέσα. Η αποθηκευτική μονάδα με ένα ορθογώνιο που του απολείπεται η δεξιά του πλευρά. Τέλος οι ροές δεδομένων αναπαριστώνται με ένα βέλος με το τύπο της ροής πάνω από το βέλος.

3.2.1.2 Διαγράμματα ροής δεδομένων- Επίπεδο 0

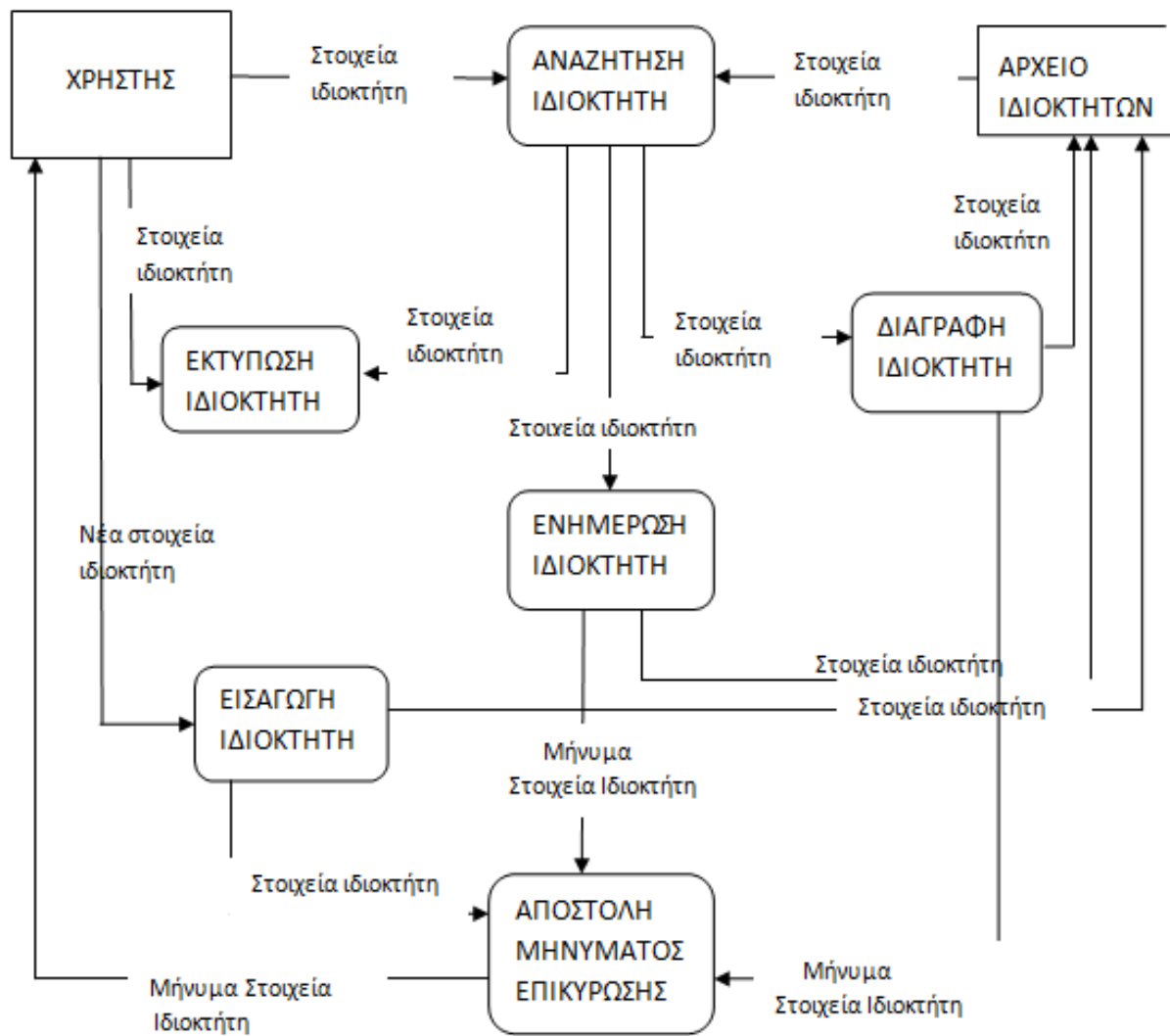


Διάγραμμα 3.1

Στο διάγραμμα ροής δεδομένων επιπέδου 0 έχουμε ένα εξωτερικό χρήστη, στον οποίο παρέχεται ένα σύνολο από λειτουργίες του συστήματος, μαζί με τις διάφορες αποθηκευτικές μονάδες.

3.2.1.3 Διαγράμματα ροής δεδομένων - Επίπεδο 1

Διαχείριση Ιδιοκτήτη



Διάγραμμα 3.2

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των ιδιοκτητών**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου ιδιοκτήτη στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για τον ιδιοκτήτη είναι το όνομα, το επίθετο, η ημερομηνία γέννησης, το φύλο, ο αριθμός ταυτότητας, η διεύθυνση και το τηλέφωνο του. Επίσης θα πρέπει να εισάγουμε το email του και το fax, καθώς και το συνθηματικό και τον κωδικό που θα του επιτρέπουν να έχει πρόσβαση στο σύστημα.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του ιδιοκτήτη.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του ιδιοκτήτη στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός ιδιοκτήτη στο σύστημα.

Για την ανεύρεση ενός ιδιοκτήτη για διαγραφή θα δίνεται ως δεδομένο εισόδου η ταυτότητα του, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου ιδιοκτήτη.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του ιδιοκτήτη από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των προσωπικών στοιχείων του ιδιοκτήτη στη βάση. Για παράδειγμα μπορεί το κινητό να αλλάξει, όπως και η διεύθυνση του ιδιοκτήτη.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου ιδιοκτήτη και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του ιδιοκτήτη.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου ιδιοκτήτη.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του ιδιοκτήτη.

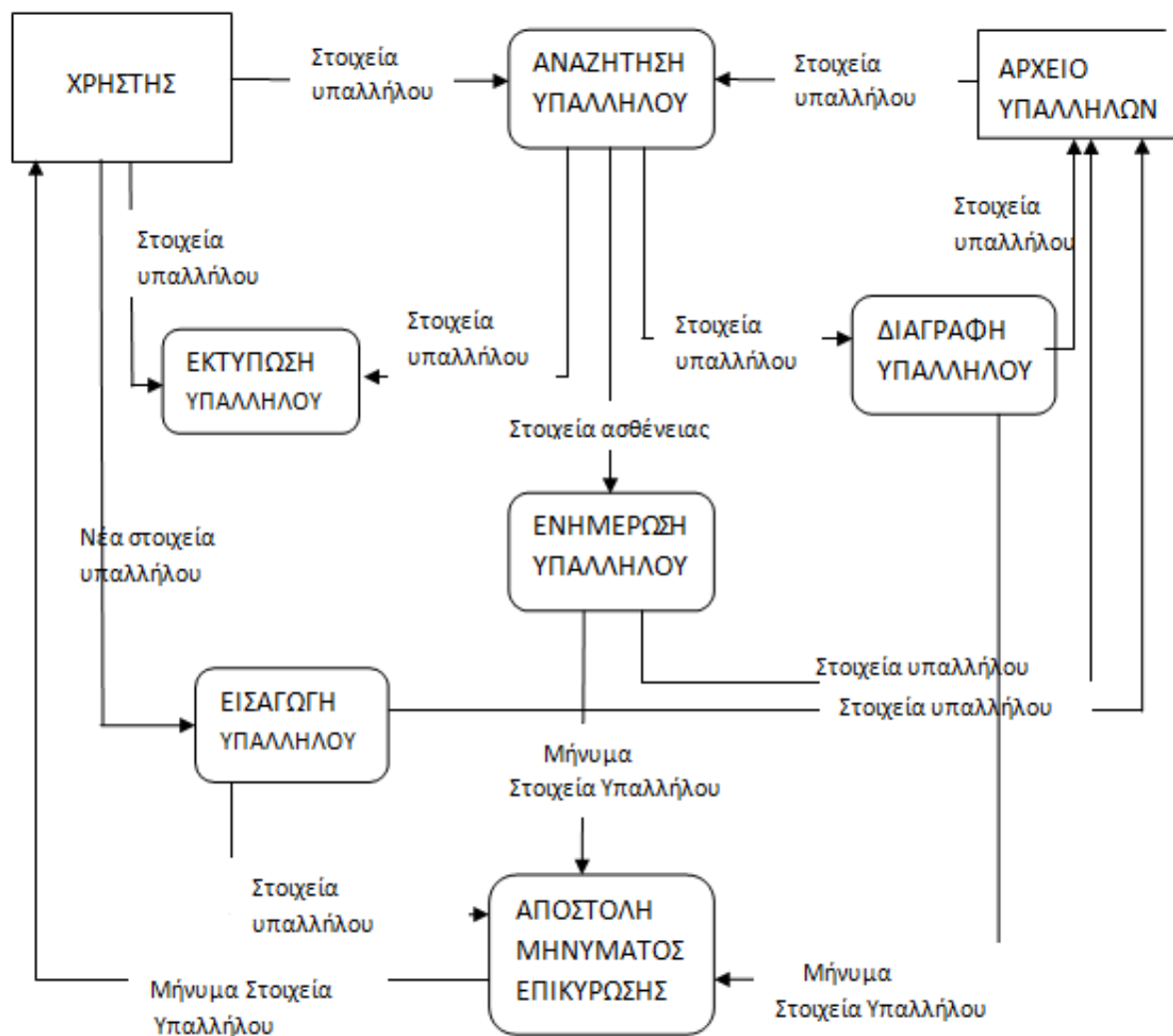
Θα υπάρχει δυνατότητα **αναζήτησης** ιδιοκτητών από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του ιδιοκτήτη, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τον ιδιοκτήτη, δηλαδή το τηλέφωνο ή το ονοματεπώνυμο του.

Εύρεση του συγκεκριμένου ιδιοκτήτη από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν τον ιδιοκτήτη που πληροί τα κριτήρια της αναζήτησης

Διαχείριση Υπαλλήλου



Διάγραμμα 3.3

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των υπαλλήλων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου υπάλληλου στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για τον υπάλληλο είναι το όνομα, το επίθετο, η ημερομηνία γέννησης, το φύλο, ο αριθμός ταυτότητας, η διεύθυνση και το τηλέφωνο του. Επίσης θα πρέπει να εισάγουμε το email του και το fax, καθώς και το συνθηματικό και τον κωδικό που θα του επιτρέπουν να έχει πρόσβαση στο σύστημα.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του υπαλλήλου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του υπαλλήλου στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός υπαλλήλου στο σύστημα.

Για την ανεύρεση ενός υπαλλήλου για διαγραφή θα δίνεται ως δεδομένο εισόδου η ταυτότητα του, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου υπαλλήλου.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του υπαλλήλου από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των προσωπικών στοιχείων του υπαλλήλου στη βάση. Για παράδειγμα μπορεί το κινητό να αλλάξει, όπως και η διεύθυνση του υπαλλήλου.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου υπαλλήλου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του υπαλλήλου.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου υπαλλήλου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του υπαλλήλου.

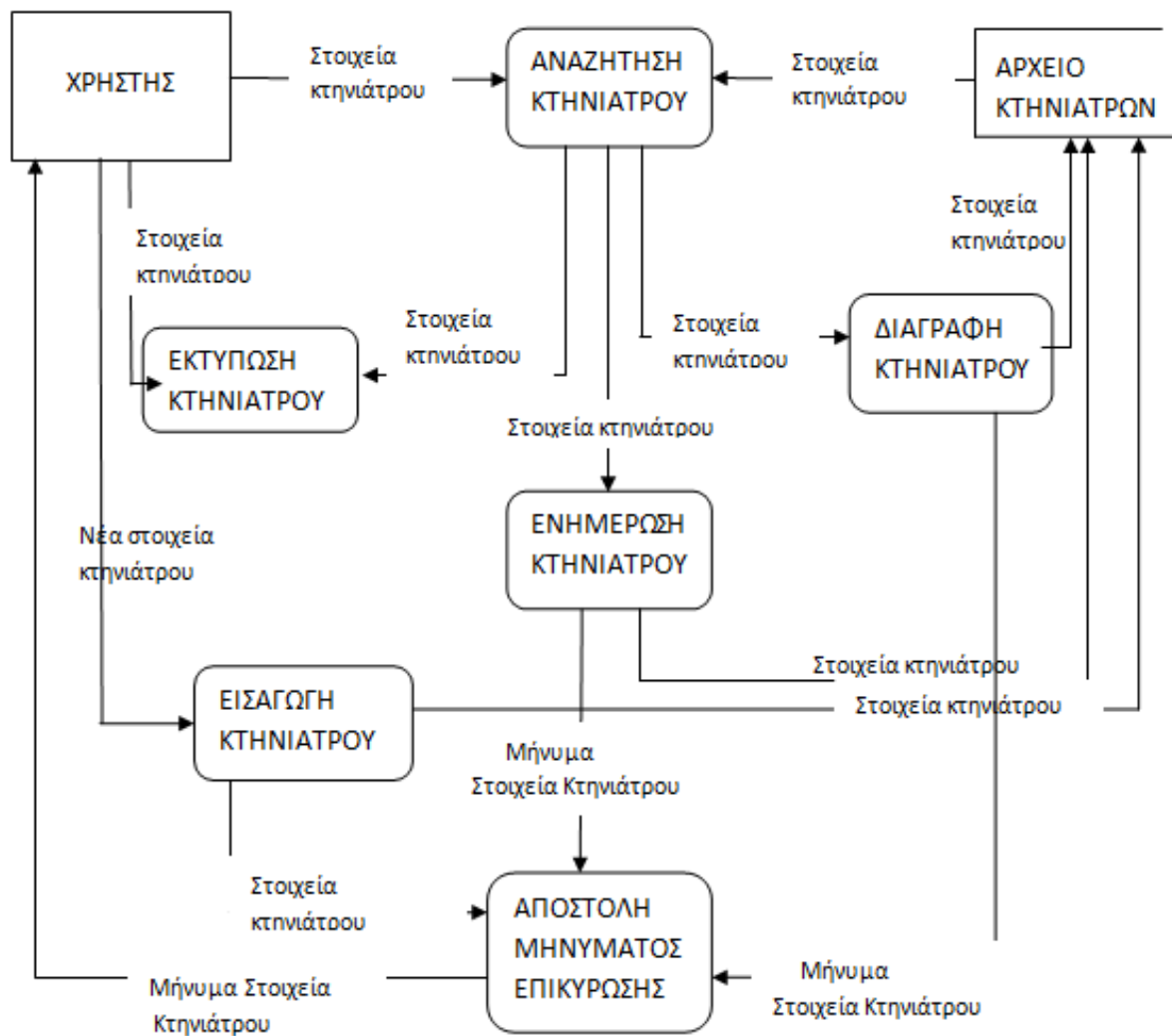
Θα υπάρχει δυνατότητα **αναζήτησης** υπαλλήλων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του υπαλλήλου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τον υπάλληλο, δηλαδή το τηλέφωνο ή το ονοματεπώνυμο του.

Εύρεση του συγκεκριμένου υπαλλήλου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν τον υπάλληλο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Κτηνιάτρου



Διάγραμμα 3.4

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των κτηνιάτρων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου κτηνιάτρου στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για τον κτηνίατρο είναι το όνομα, το επίθετο, η ημερομηνία γέννησης, το φύλο, ο αριθμός ταυτότητας, η διεύθυνση και το τηλέφωνο του. Επίσης θα πρέπει να εισάγουμε το email του και το fax, καθώς και την διεύθυνση του ιατρείου του.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του κτηνίατρου. Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του κτηνίατρου στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός κτηνίατρου στο σύστημα.

Για την ανεύρεση ενός κτηνίατρου για διαγραφή θα δίνεται ως δεδομένο εισόδου η ταυτότητα του, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου κτηνίατρου.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του κτηνίατρου από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των προσωπικών στοιχείων του κτηνίατρου στη βάση. Για παράδειγμα μπορεί το κινητό να αλλάξει, όπως και η διεύθυνση του υπαλλήλου.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου κτηνίατρου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του κτηνίατρου.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου κτηνίατρου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του κτηνίατρου.

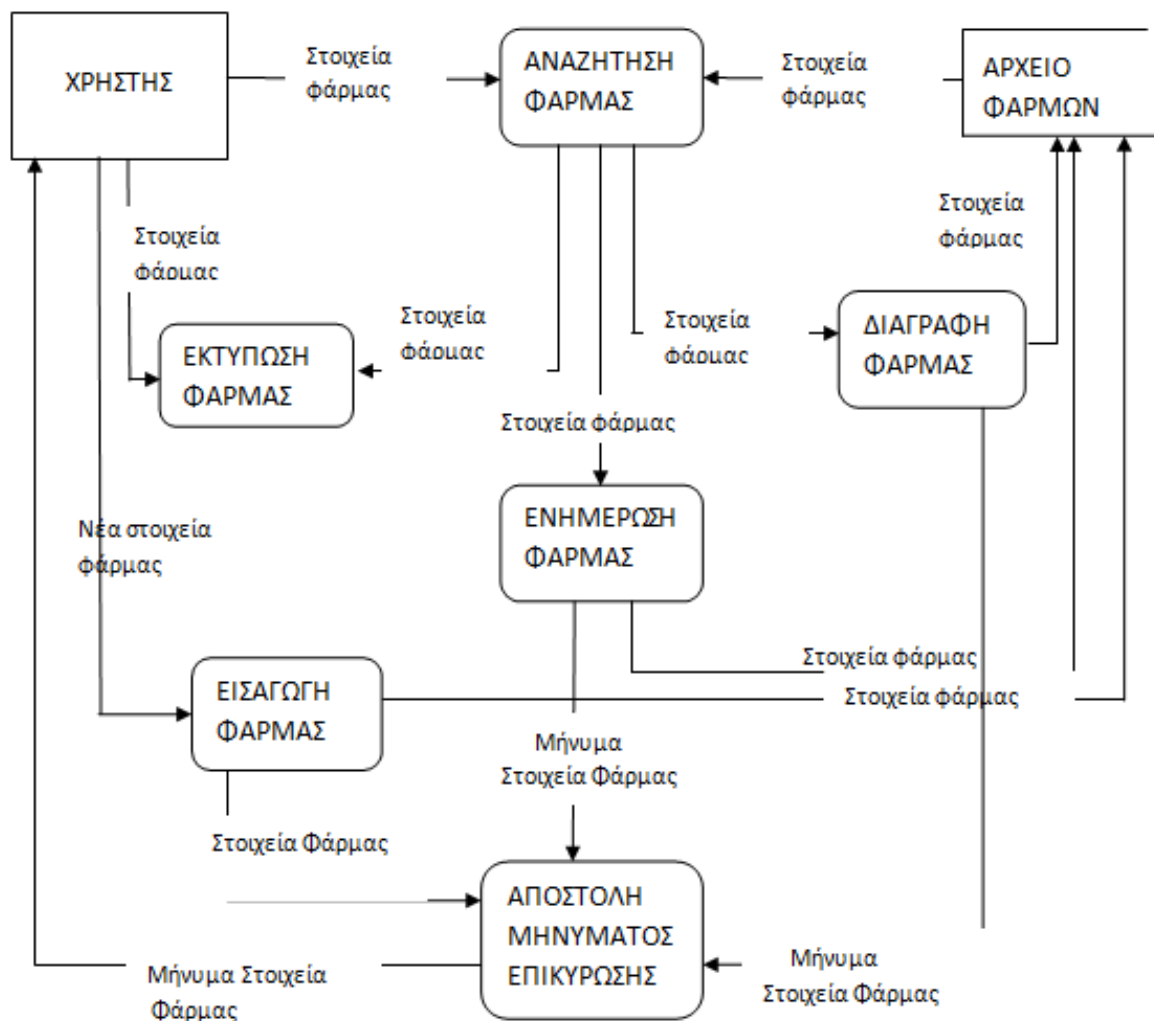
Θα υπάρχει δυνατότητα **αναζήτησης** κτηνίατρων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του κτηνίατρου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τον κτηνίατρο, δηλαδή το τηλέφωνο ή το ονοματεπώνυμο του.

Εύρεση του συγκεκριμένου κτηνίατρου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν τον κτηνίατρο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Φάρμας



Διάγραμμα 3.5

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία της φάρμας**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέας φάρμας στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την φάρμα είναι το όνομα, η διεύθυνση της φάρμας, τα τηλέφωνα επικοινωνίας καθώς και το email και fax της φάρμας. Επίσης, θα υπάρχει δυνατότητα καταχώρησης της κυβερνητικής άδειας της φάρμας.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της φάρμας. Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της φάρμας στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** μίας φάρμας στο σύστημα.

Για την ανεύρεση της φάρμας για διαγραφή θα δίνεται ως δεδομένο εισόδου το όνομα, αλλά θα υπάρχει και δυνατότητα αναζήτησης της με άλλο κριτήριο.

Εύρεση και διαγραφή της συγκεκριμένης φάρμας.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτής της φάρμας από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων της φάρμας στη βάση. Για παράδειγμα μπορεί το όνομα να αλλάξει, όπως και η κυβερνητική άδεια της φάρμας.

Αρχικά θα γίνεται η ανεύρεση της συγκεκριμένης φάρμας, και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή της φάρμας.

Εύρεση και ενημέρωση των στοιχείων της συγκεκριμένης φάρμας.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων της φάρμας.

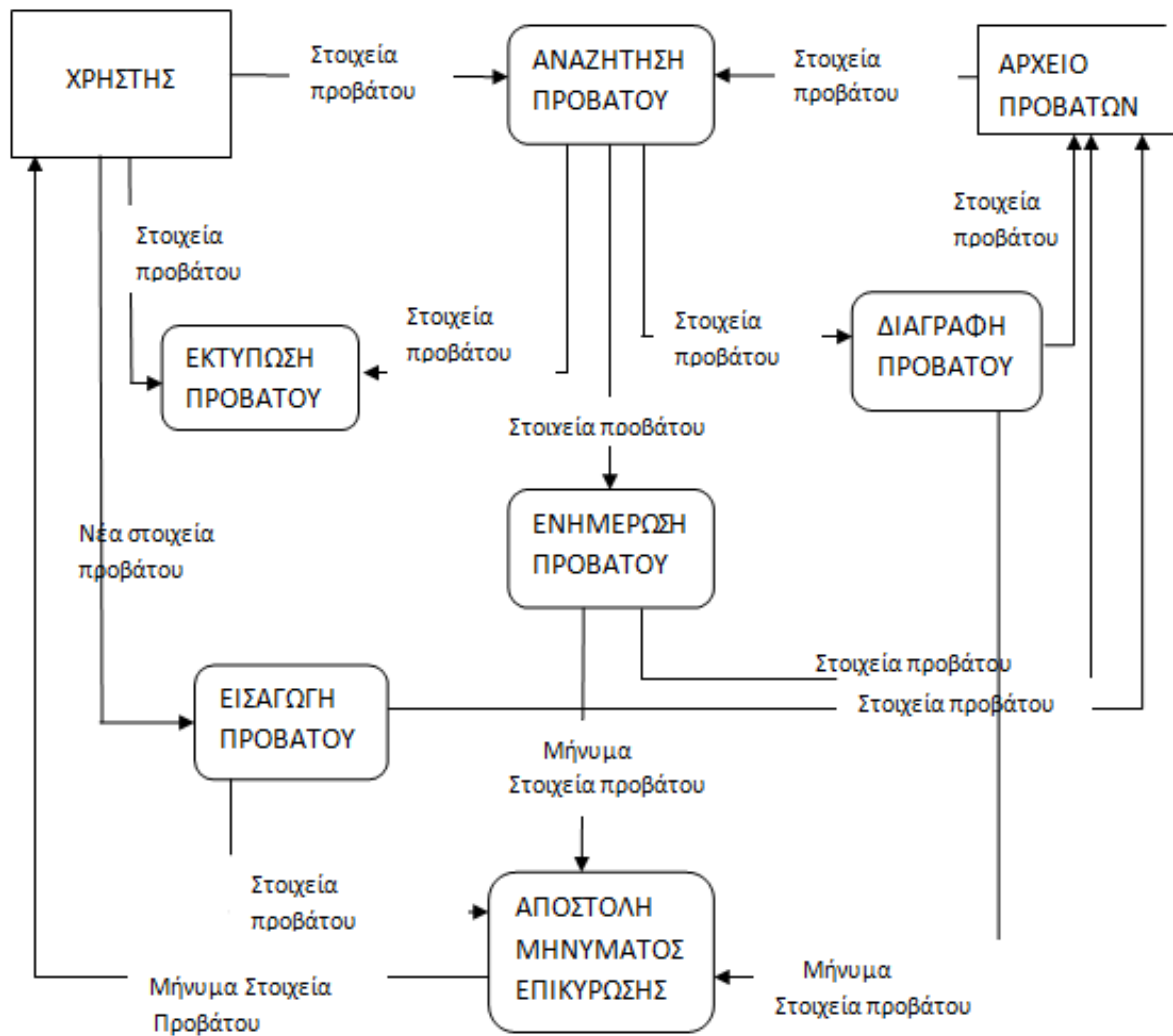
Θα υπάρχει δυνατότητα **αναζήτησης** της φάρμας από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση της φάρμας, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν τη φάρμα, δηλαδή το τηλέφωνο ή το όνομα της.

Εύρεση της συγκεκριμένης φάρμας από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την φάρμα που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Προβάτου



Διάγραμμα 3.6

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των προβάτων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου προβάτου στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για το πρόβατο είναι ο κωδικός του, το όνομα, η ημερομηνία γέννησης, το φύλο, ο γονότυπος, το βάρος του, το βάρος μετά από 110 ημέρες, η φάρμα στην οποία ανήκει, η κατανάλωση του μέχρι στιγμής και το αντίστοιχο κόστος, μία φωτογραφία του, ο κωδικός της μητέρας και του πατέρα του, καθώς και το είδος του, πρόβατο ή κατσίκι. Επίσης, αν το ζώο είναι θηλυκό τότε δίνεται η δυνατότητα εισαγωγής της ποσότητας γάλακτος που παρήγαγε και το αντίστοιχο κόστος του.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του προβάτου. Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του προβάτου στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός προβάτου από το σύστημα.

Για την ανεύρεση ενός προβάτου για διαγραφή θα δίνεται ως δεδομένο εισόδου ο κωδικός του, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου προβάτου.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του προβάτου από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων του προβάτου στη βάση.

Για παράδειγμα μπορεί το βάρος να αλλάξει, όπως και η παραγωγή του σε γάλα του προβάτου.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου προβάτου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του προβάτου.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου προβάτου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του προβάτου.

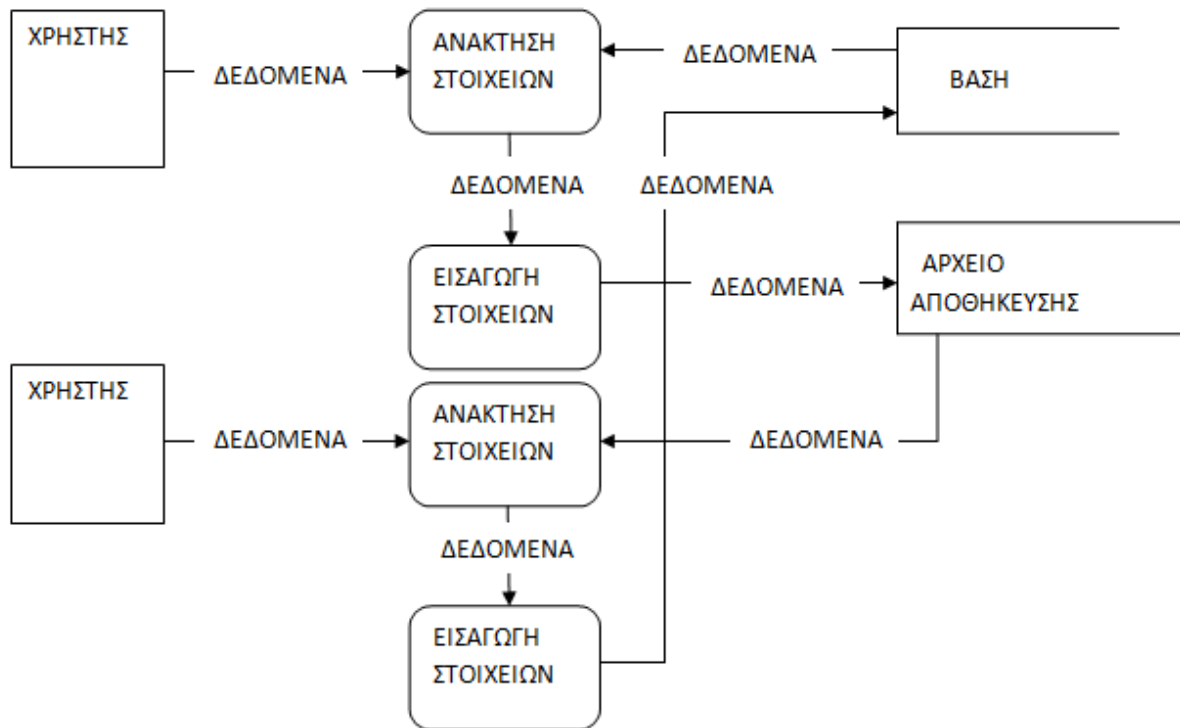
Θα υπάρχει δυνατότητα **αναζήτησης** προβάτων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του προβάτου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το πρόβατο, δηλαδή ο κωδικός ή το όνομα του.

Εύρεση του συγκεκριμένου προβάτου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το πρόβατο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση LOG FILE



Διάγραμμα 3.7

Κάθε φορά που ένας χρήστης κάνει μία αλλαγή που να επηρεάζει τη βάση του ΒΟ συστήματος, τότε το LOG FILE θα ενημερώνεται αποθηκεύοντας μία νέα εγγραφή σε αυτό, η οποία θα αντιπροσωπεύει την αλλαγή που έγινε. Οι αλλαγές αυτές αφορούν την εισαγωγή κάποιας νέας εγγραφής, τη διαγραφή της και την ενημέρωσή της.

Η λειτουργία αυτή θα εκτελείται αυτόματα κάθε φορά που ο χρήστης κάνει μία αλλαγή στο σύστημα. Σαν είσοδο θα παίρνει ένα αύξον αριθμό του χρήστη, την ημερομηνία σύνδεσης, την ώρα σύνδεσης, τη διάρκεια σύνδεσης καθώς και την ενέργεια που εκτελέστηκε.

Εισαγωγή στη βάση μίας νέας εγγραφής LOG FILE με το δικό της αύξον αριθμό.

Η λειτουργία αυτή δεν είναι ορατή προς το χρήστη. Έτσι δεν θα εμφανίζεται κάποιο μήνυμα στην οθόνη. Οι αλλαγές γίνονται μόνο στο ΒΟ.

Ο διαχειριστής του συστήματος θα μπορεί να βλέπει το LOG FILE και να πληροφορείται ποιες αλλαγές έγιναν από το σύστημα και από ποιους. Έτσι σε περίπτωση σφάλματος ή χάσιμο δεδομένων, τα δεδομένα θα μπορούν να ανακτηθούν.

Η μόνη είσοδος είναι το πάτημα του κουμπιού για το LOG FILE.

Φόρτωση LOG FILE

Εμφάνιση του LOG FILE στην οθόνη.

Προβολή πιο πρόσφατων εγγράφων LOG FILE που έχουν σχέση με το λογισμικό διαχείρισης της ΒΔ.

Ο διαχειριστής κατά την είσοδο του στο BO θα μπορεί να βλέπει τις πιο πρόσφατες εγγραφές που ενημερώθηκαν, και οι οποίες έγιναν από τη τελευταία φορά που είχε κάνει login στο BO

Η λειτουργία αυτή θα εκτελείται αυτόματα κάθε φορά που ο διαχειριστής κάνει login στο BO. Σαν είσοδο θα παίρνει την ημερομηνία σύνδεσης και την ώρα σύνδεσης, καθώς και την ημερομηνία που έγινε το τελευταίο logout του διαχειριστή. Επίσης σαν είσοδο θα παίρνει κάποιο flag ώστε να επιλέγονται οι κατάλληλες εγγραφές του log file για να παρουσιάζονται στο BO.

Επιλογή των κατάλληλων εγγραφών του log file με βάση τα κριτήρια των δεδομένων εισόδου.

Εμφάνιση των εγγραφών του log file που έχουν επιλεγεί στην οθόνη.

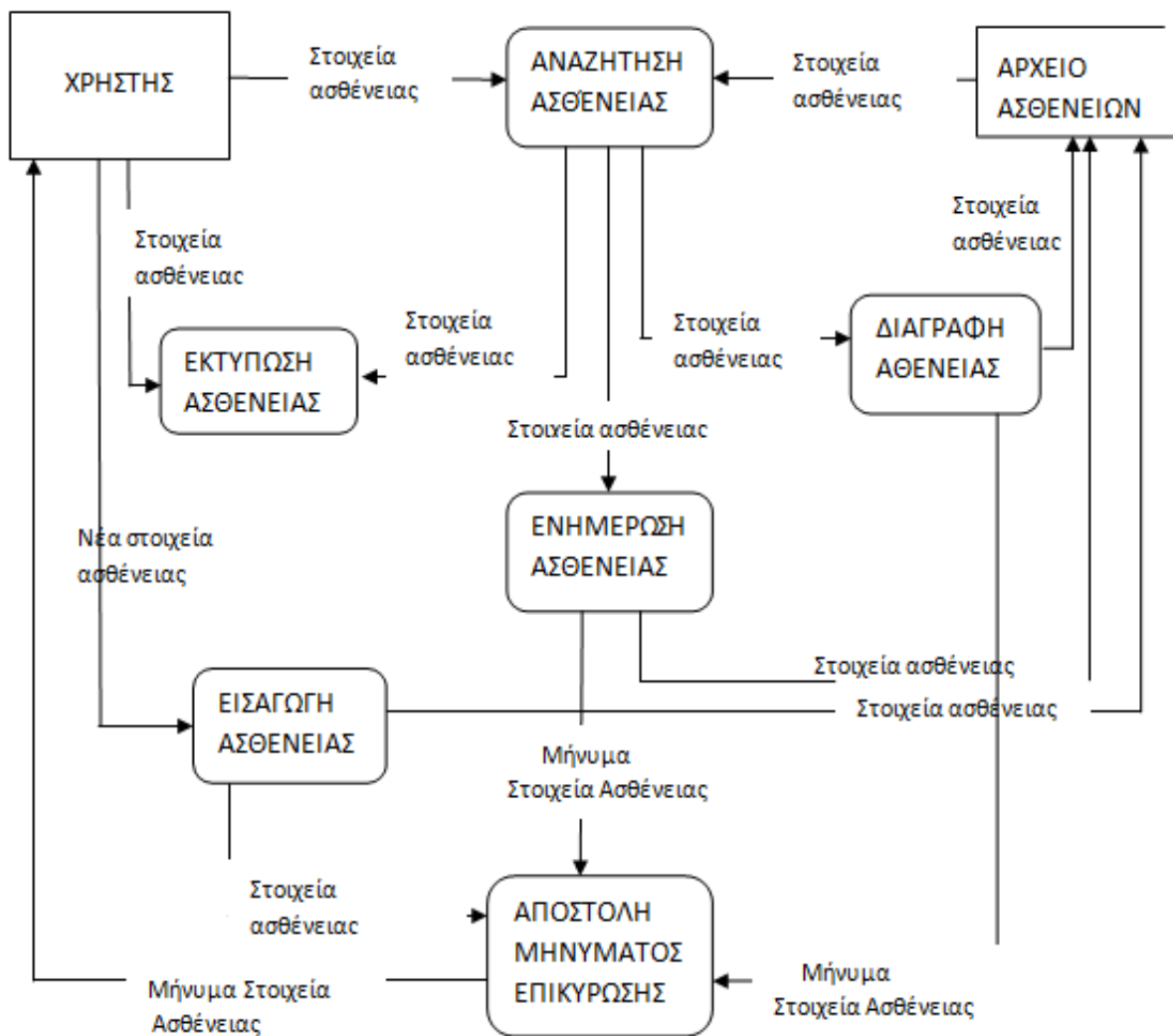
Ο διαχειριστής θα μπορεί να σβήνει πληροφορίες από το log file που είναι πλέον αχρείαστες. Αυτό βοηθάει σε θέματα ασφαλείας, αλλά και σε θέματα μνήμης, αφού απελευθερώνουμε χώρο από το log file.

Ο διαχειριστής του συστήματος θα έχει την ικανότητα να επιλέγει με κάποιο τρόπο τις παλιές και αχρείαστες εγγραφές και να μπορεί να τις διαγράψει. Επίσης θα μπορεί να διαγράψει όλες τις εγγραφές ή μία ομάδα αυτών. Ακόμα και ολική εκκαθάριση του log file.

Διαγραφή όλων των εγγραφών από το log file που επέλεξε ο χρήστης να διαγράψει.

Μήνυμα επιβεβαίωσης ότι οι επιλεγμένες εγγραφές έχουν διαγραφεί επιτυχώς από το log file, ή ότι έχει καθαρίσει ολόκληρο το log file.

Διαχείριση Ασθενειών



Διάγραμμα 3.8

Θα υπάρχει δυνατότητα **εισαγωγής νέας ασθένειας** στο σύστημα.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της ασθένειας. Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την πρόθεση της νέας ασθένειας στη ΒΔ.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων της ασθένειας στη βάση. Για παράδειγμα μπορεί το όνομα να αλλάξει, όπως και η περιγραφή του, καθώς και το βάρος/σημαντικότητα του.

Αρχικά θα γίνεται η ανεύρεση της συγκεκριμένης ασθένειας και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή της ασθένειας.

Εύρεση και ενημέρωση των στοιχείων της συγκεκριμένης ασθένειας.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων της ασθένειας.

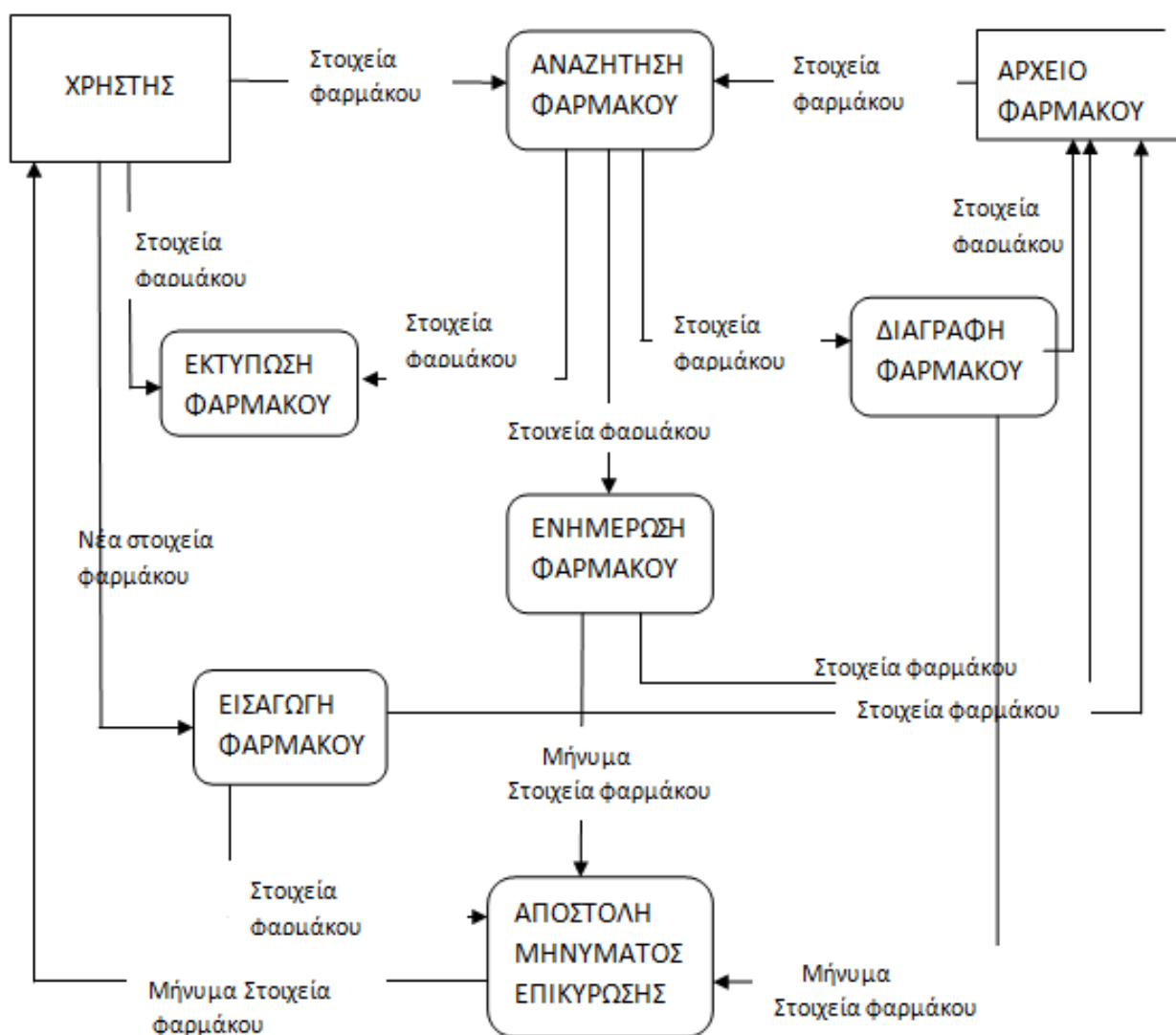
Θα υπάρχει δυνατότητα **αναζήτησης** των ασθενειών από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση της ασθένειας, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν την ασθένεια, δηλαδή το όνομα της.

Εύρεση της συγκεκριμένης ασθένειας από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την ασθένεια που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Φαρμάκων



Διάγραμμα 3.9

Θα υπάρχει δυνατότητα **εισαγωγής νέου φαρμάκου** στο σύστημα.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του φαρμάκου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την πρόθεση του νέου φαρμάκου στη ΒΔ.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων του φαρμάκου στη βάση.

Για παράδειγμα μπορεί το όνομα να αλλάξει, όπως και η περιγραφή του, καθώς και η τιμή του.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου φαρμάκου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του φαρμάκου.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου φαρμάκου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του φαρμάκου.

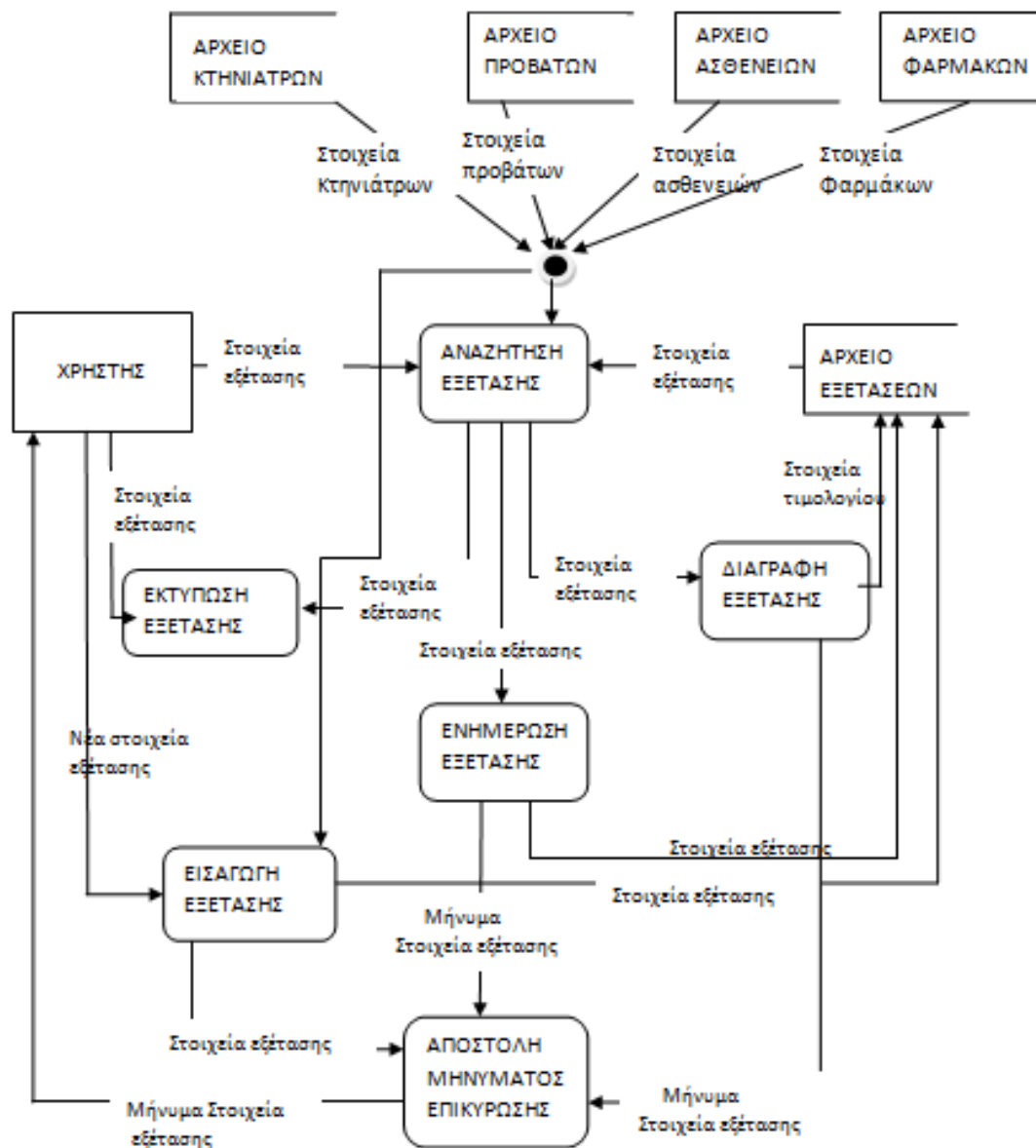
Θα υπάρχει δυνατότητα **αναζήτησης** των φαρμάκων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του φαρμάκου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το φάρμακο, δηλαδή το όνομα του.

Εύρεση του συγκεκριμένου φαρμάκου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το φάρμακο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Εξετάσεων



Διάγραμμα 3.10

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των εξετάσεων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέας εξέτασης στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την εξέταση είναι η ημερομηνία που έγινε η εξέταση, ο κτηνίατρος ο οποίος έκαμε την εξέταση, το πρόβατο το οποίο εξετάστηκε, η ασθένεια η οποία διαγνώστηκε, τα φάρμακα τα οποία χορηγήθηκαν και κάποια σχόλια από τον κτηνίατρο.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της εξέτασης.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της εξέτασης στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** μίας εξέτασης από το σύστημα.

Για την ανεύρεση μίας εξέτασης για διαγραφή θα δίνεται ως δεδομένο εισόδου το πρόβατο το οποίο την αφορά, αλλά θα υπάρχει και δυνατότητα αναζήτησης της με άλλο κριτήριο.

Εύρεση και διαγραφή της συγκεκριμένης εξέτασης.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτής της εξέτασης από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων της εξέτασης στη βάση.

Για παράδειγμα μπορεί το πρόβατο να αλλάξει, όπως και η ασθένεια η οποία διαγνώστηκε.

Αρχικά θα γίνεται η ανεύρεση της συγκεκριμένης απόδειξης και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του.

Εύρεση και ενημέρωση των στοιχείων της συγκεκριμένης εξέτασης.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων της εξέτασης.

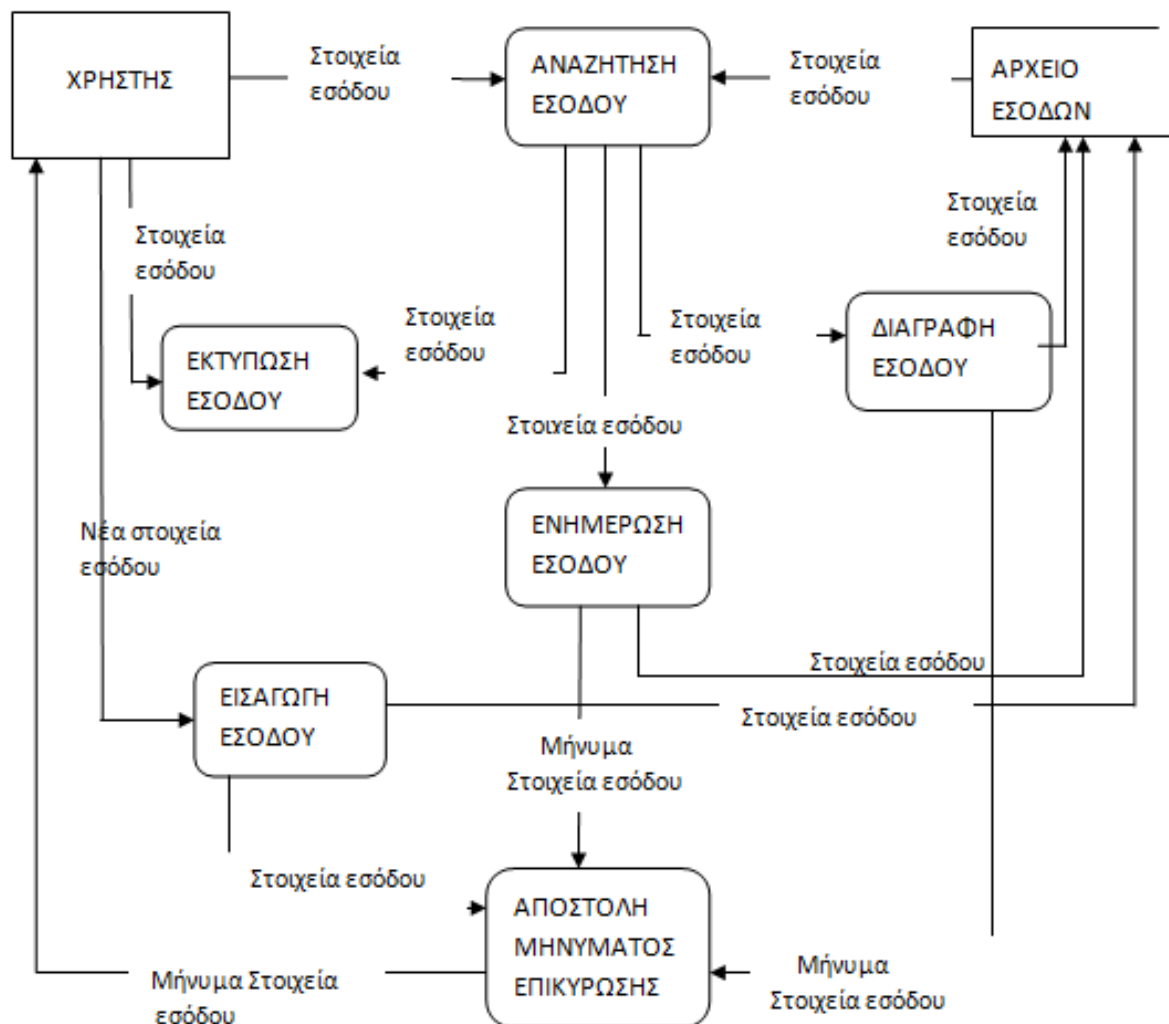
Θα υπάρχει δυνατότητα **αναζήτησης** εξετάσεων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση της εξέτασης, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν την εξέταση, δηλαδή το πρόβατο που αφορά η εξέταση όπως και η ημερομηνία που έγινε η εξέταση.

Εύρεση της συγκεκριμένης εξέτασης από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την εξέταση που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Εσόδων



Διάγραμμα 3. 11

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των εσόδων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου εσόδου στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για το έσοδο είναι η φάρμα στην οποία ανήκει, η ημερομηνία καταχώρησης του, η περιγραφή του καθώς και η μονάδα μέτρησης του εσόδου, και τέλος, η τιμή του ανά μονάδα μέτρησης.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του εσόδου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του εσόδου στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός εσόδου από το σύστημα.

Για την ανεύρεση ενός εσόδου για διαγραφή θα δίνεται ως δεδομένο εισόδου η περιγραφή του, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου εσόδου.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του εσόδου από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων του εσόδου στη βάση. Για παράδειγμα μπορεί η τιμή να αλλάξει, όπως και η φάρμα στην οποία ανήκει το έσοδο.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου εσόδου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του εσόδου.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου εσόδου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του εσόδου.

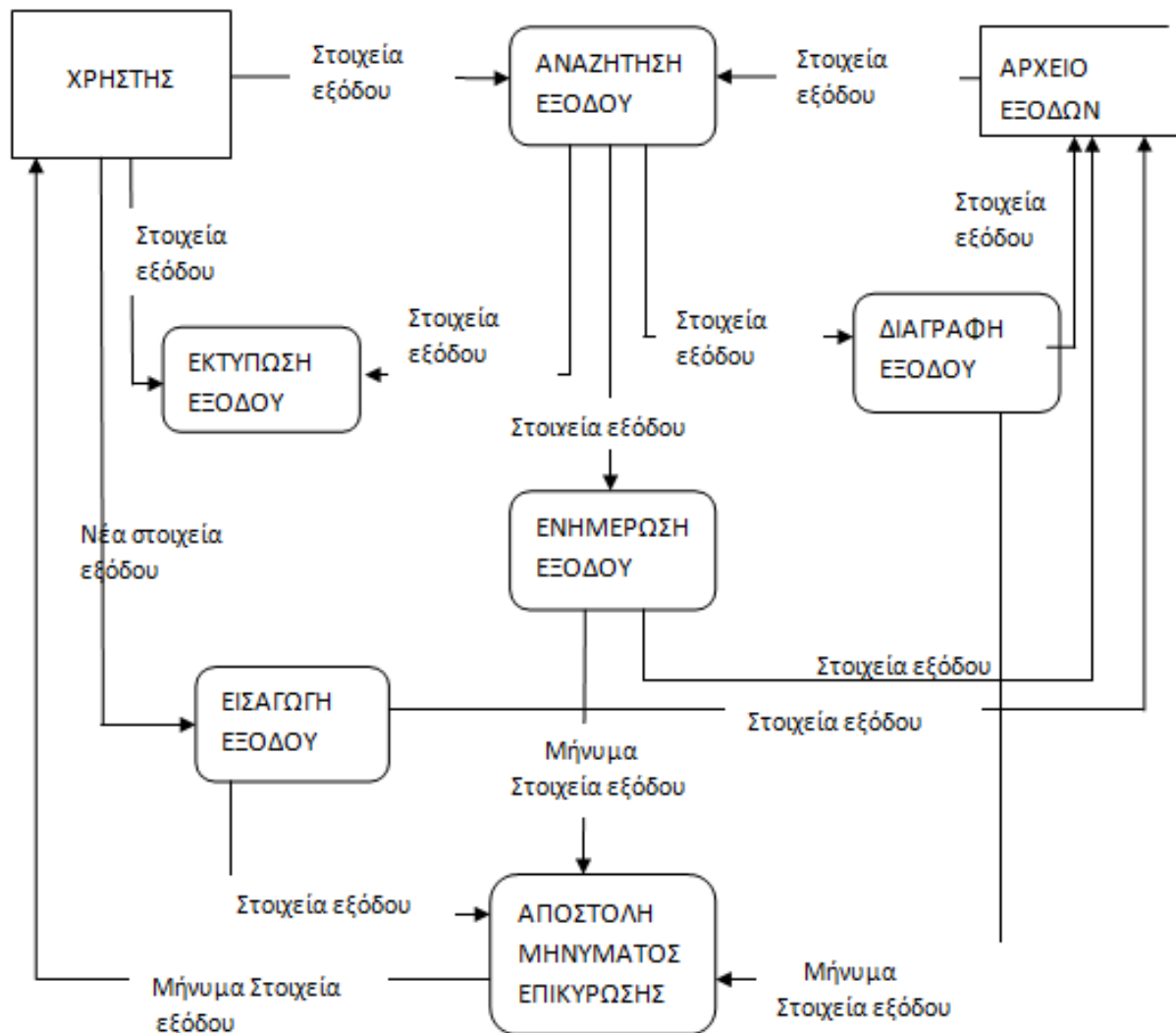
Θα υπάρχει δυνατότητα **αναζήτησης** εσόδων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του εσόδου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το έσοδο, δηλαδή η φάρμα στην οποία ανήκει και η περιγραφή του.

Εύρεση του συγκεκριμένου εσόδου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το έσοδο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Εξόδων



Διάγραμμα 3.12

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των εξόδων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου εξόδου στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για το έξοδο είναι η φάρμα στην οποία ανήκει, η ημερομηνία καταχώρησης του, η περιγραφή του καθώς και η μονάδα μέτρησης του εισόδου, και τέλος, η τιμή του ανά μονάδα μέτρησης. Επιπλέον, για κάθε έξοδο θα εισάγεται και ο τύπος του, δηλαδή αν είναι τρέχων ή σταθερό έξοδο.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του εξόδου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του εξόδου στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός εξόδου από το σύστημα.

Για την ανεύρεση ενός εξόδου για διαγραφή θα δίνεται ως δεδομένο εισόδου η περιγραφή του, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου εξόδου.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του εξόδου από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων του εξόδου στη βάση. Για παράδειγμα μπορεί η τιμή να αλλάξει, όπως και η φάρμα στην οποία ανήκει το έξοδο.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου εξόδου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου εξόδου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του εξόδου.

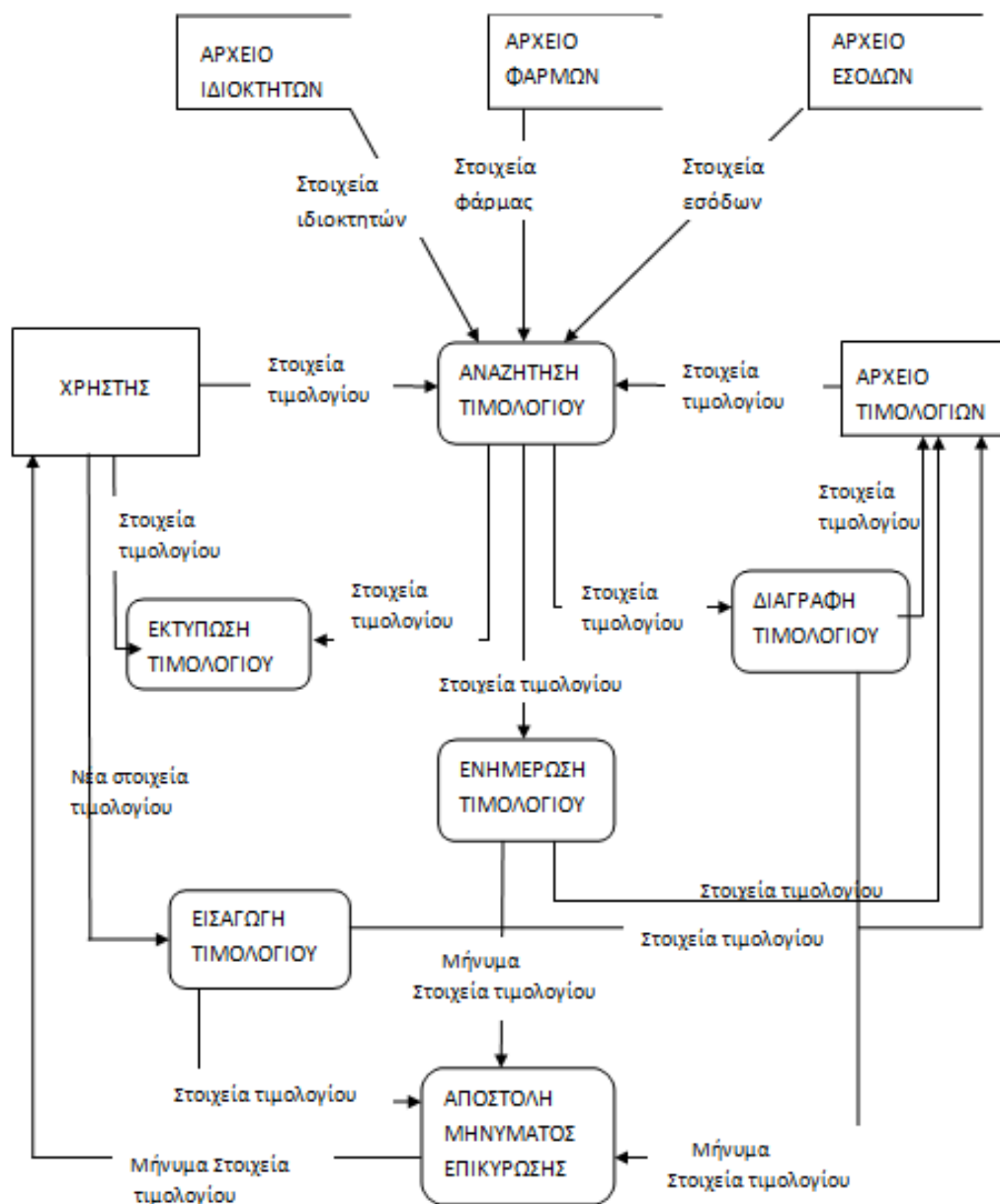
Θα υπάρχει δυνατότητα **αναζήτησης** εξόδων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του εξόδου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το έξοδο, δηλαδή η φάρμα στην οποία ανήκει και η περιγραφή του.

Εύρεση του συγκεκριμένου εξόδου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το έξοδο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Τιμολογίων



Διάγραμμα 3.13

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των τιμολογίων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέου τιμολογίου στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για το τιμολόγιο είναι ο ιδιοκτήτης της φάρμας τον οποίο αφορά το τιμολόγιο, η αντίστοιχη φάρμα, το έσοδο ή έξοδο για το οποίο δημιουργήθηκε το τιμολόγιο, η ημερομηνία καταχώρησης του καθώς και ο

αριθμός τεμαχίων του εισόδου/ εξόδου που επιλέχθηκε προηγουμένως. Ο χρήστης πατώντας το κουμπί «Υπολογισμός», θα υπολογίζεται αυτόματα το συνολικό ποσό του τιμολογίου και θα παρουσιάζεται στην φόρμα για τον χρήστη.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων του τιμολογίου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση του τιμολογίου στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** ενός τιμολογίου από το σύστημα.

Για την ανεύρεση ενός τιμολογίου για διαγραφή θα δίνεται ως δεδομένο εισόδου το έσοδο ή έξοδο το οποίο το αφορά, αλλά θα υπάρχει και δυνατότητα αναζήτησης του με άλλο κριτήριο.

Εύρεση και διαγραφή του συγκεκριμένου τιμολογίου.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτού του τιμολογίου από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων του τιμολογίου στη βάση.

Για παράδειγμα μπορεί ο αριθμός των τεμαχίων να αλλάξει, όπως και η φάρμα στην οποία ανήκει το τιμολόγιο.

Αρχικά θα γίνεται η ανεύρεση του συγκεκριμένου τιμολογίου και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή του.

Εύρεση και ενημέρωση των στοιχείων του συγκεκριμένου τιμολογίου.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων του τιμολογίου.

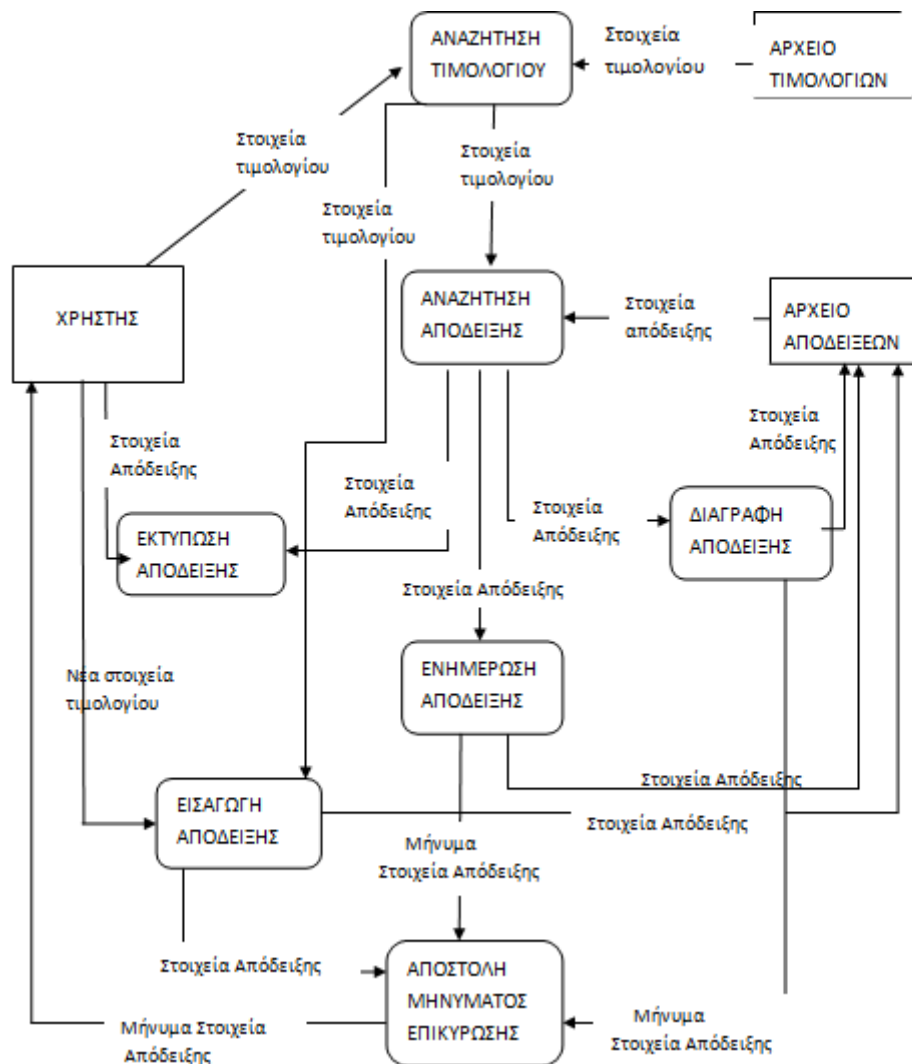
Θα υπάρχει δυνατότητα **αναζήτησης** τιμολογίων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση του τιμολογίου, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν το τιμολόγιο, δηλαδή η φάρμα στην οποία ανήκει και η περιγραφή του εισόδου/εξόδου του.

Εύρεση του συγκεκριμένου τιμολογίου από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν το τιμολόγιο που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Αποδείξεων



Διάγραμμα 3.14

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία των αποδείξεων**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέας απόδειξης στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την απόδειξη είναι το τιμολόγιο το οποίο αφορά η συγκεκριμένη απόδειξη, γι' αυτό και ο χρήστης θα πρέπει να έχει

προηγούμενως επιλέξει το αντίστοιχο τιμολόγιο, το ποσό που πρόκειται να πληρωθεί, η ημερομηνία της συναλλαγής και κάποια σχόλια που πιθανών να θέλει να καταχωρήσει ο ιδιοκτήτης.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της απόδειξης.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της απόδειξης στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **διαγραφής** μίας απόδειξης από το σύστημα.

Για την ανεύρεση μίας απόδειξης για διαγραφή θα δίνεται ως δεδομένο εισόδου το τιμολόγιο το οποίο την αφορά, αλλά θα υπάρχει και δυνατότητα αναζήτησης της με άλλο κριτήριο.

Εύρεση και διαγραφή της συγκεκριμένης απόδειξης.

Θα εκτυπώνεται στην οθόνη ένα μήνυμα που θα βεβαιώνει τη διαγραφή αυτής της απόδειξης από την βάση δεδομένων.

Θα υπάρχει δυνατότητα **ενημέρωσης** των στοιχείων της απόδειξης στη βάση.

Για παράδειγμα μπορεί το τιμολόγιο να αλλάξει, όπως και το ποσό το οποίο πρόκειται να πληρωθεί.

Αρχικά θα γίνεται η ανεύρεση της συγκεκριμένης απόδειξης και ακολούθως θα παίρνει ως δεδομένα εισόδου για ενημέρωση όλα τα δεδομένα τα οποία δίδονται αρχικά για την εγγραφή της.

Εύρεση και ενημέρωση των στοιχείων της συγκεκριμένης απόδειξης.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει την ενημέρωση των στοιχείων της απόδειξης.

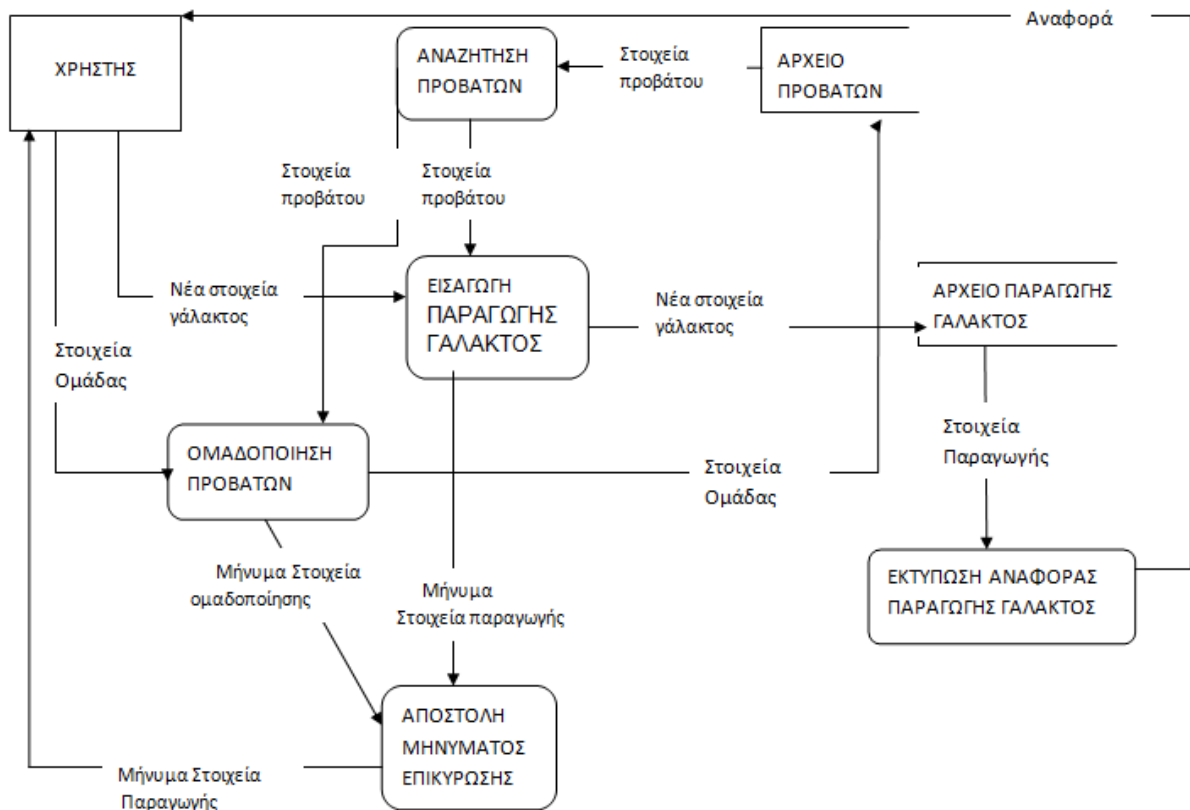
Θα υπάρχει δυνατότητα **αναζήτησης** αποδείξεων από τη βάση δεδομένων και παρουσίαση των στοιχείων τους σε μορφή Report.

Για την αναζήτηση της απόδειξης, τα δεδομένα εισόδου θα είναι οποιαδήποτε στοιχεία τα οποία ταυτοποιούν την απόδειξη, δηλαδή το τιμολόγιο που αφορά η απόδειξη όπως και το ποσό που έχει πληρωθεί.

Εύρεση της συγκεκριμένης απόδειξης από τη βάση.

Εκτύπωση στην οθόνη του υπολογιστή των πληροφοριών που αφορούν την απόδειξη που πληροί τα κριτήρια της αναζήτησης.

Διαχείριση Παραγωγής Γάλακτος



Διάγραμμα 3.15

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η λειτουργία της παραγωγής γάλακτος.

Θα υπάρχει δυνατότητα **εισαγωγής** νέας παραγωγής γάλακτος στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την παραγωγή γάλακτος είναι το πρόβατο το οποίο αφορά η συγκεκριμένη παραγωγή, γι' αυτό και ο χρήστης θα πρέπει να έχει προηγουμένως επιλέξει το αντίστοιχο πρόβατο, η ποσότητα γάλακτος για το αντίστοιχο πρόβατο και η ημερομηνία καταχώρησης της παραγωγής.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της παραγωγής γάλακτος.

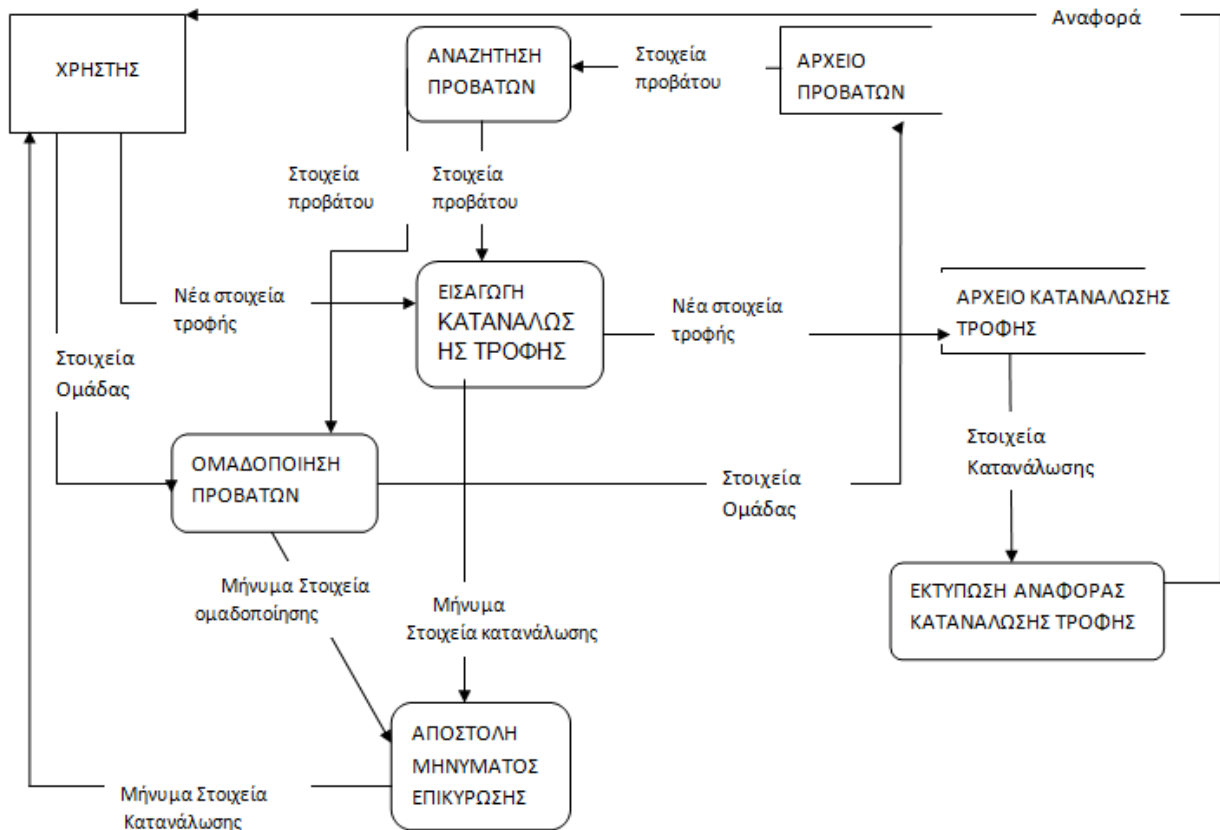
Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της παραγωγής στη βάση δεδομένων.

Θα υπάρχει δυνατότητα **ομαδοποίησης** των προβάτων, ανάλογα με την θέση που παίρνει το πρόβατο στο γαλευτήρι και θα παρουσιάζεται η δυνατότητα εισαγωγής του με γραφικό τρόπο, για να διευκολύνεται ο ιδιοκτήτης/υπάλληλος όταν εισάγει την ποσότητα γάλακτος του αντίστοιχου προβάτου.

Θα υπάρχει δυνατότητα **εκτύπωσης αναφοράς** παραγωγής γάλακτος του συστήματος διαχείρισης της κτηνοτροφικής μονάδας.

Ο χρήστης θα έχει την δυνατότητα επιλογής εκτύπωσης της αναφοράς για όλα τα πρόβατα μαζί (συλλογικά), και την δυνατότητα για κάθε πρόβατο ξεχωριστά έτσι ώστε να μπορεί να διακρίνει την πρόοδο της γαλακτοπαραγωγής.

Διαχείριση Κατανάλωσης Τροφής



Διάγραμμα 3.16

Στο πιο πάνω διάγραμμα ροής δεδομένων παρουσιάζεται η **λειτουργία της κατανάλωσης τροφής**.

Θα υπάρχει δυνατότητα **εισαγωγής** νέας κατανάλωσης τροφής στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την κατανάλωση τροφής είναι το πρόβατο το οποίο αφορά η συγκεκριμένη κατανάλωση, γι' αυτό και ο χρήστης θα πρέπει να έχει προηγουμένως επιλέξει το αντίστοιχο πρόβατο, η ποσότητα τροφής για το αντίστοιχο πρόβατο και η ημερομηνία καταχώρησης της κατανάλωσης.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της κατανάλωσης τροφής.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της κατανάλωσης στη βάση δεδομένων.

θα υπάρχει δυνατότητα **ομαδοποίησης** των προβάτων, ανάλογα με την θέση που παίρνει το πρόβατο στην φάρμα και θα παρουσιάζεται η δυνατότητα εισαγωγής του με γραφικό τρόπο, για να διευκολύνεται ο ιδιοκτήτης/υπάλληλος όταν εισάγει την κατανάλωση τροφής του αντίστοιχου προβάτου.

Θα υπάρχει δυνατότητα **εκτύπωσης αναφοράς** κατανάλωσης τροφής του συστήματος διαχείρισης της κτηνοτροφικής μονάδας.

Ο χρήστης θα έχει την δυνατότητα επιλογής εκτύπωσης της αναφοράς για όλα τα πρόβατα μαζί (συλλογικά), και την δυνατότητα για κάθε πρόβατο ξεχωριστά έτσι ώστε να μπορεί να διακρίνει την πρόοδο της κατανάλωσης.

3.2.2 Μηχανογράφηση Τμημάτων

Οι στόχοι και οι δραστηριότητες της κτηνοτροφικής μονάδας επικεντρώνονται ουσιαστικά σε ένα τμήμα, το οποίο αποτελείται από τις λειτουργίες :

1. Διαχείριση ιδιοκτητών
2. Διαχείριση υπαλλήλων
3. Διαχείριση κτηνιάτρων
4. Διαχείριση προβάτων
5. Διαχείριση φάρμας
6. Διαχείριση εφεδρικού αντιγράφου – LOG FILE
7. Διαχείριση ασθενειών
8. Διαχείριση φαρμάκων
9. Διαχείριση εξέτασης
10. Διαχείριση εσόδων
11. Διαχείριση εξόδων
12. Διαχείριση τιμολογίων
13. Διαχείριση αποδείξεων
14. Διαχείριση παραγωγής γάλακτος
15. Διαχείριση κατανάλωσης τροφής

Οι πιο πάνω λειτουργίες είναι αυτές που θα αποτελούν το νέο σύστημα για την διεκπεραίωση των αναγκών του χρήστη. Θα πραγματοποιηθούν όλες οι πιο πάνω λειτουργικότητες του νέου συστήματος που αναφέρθηκαν για την διεκπεραίωση των

αναγκών του χρήστη. Οι λειτουργικότητες έχουν προγραμματιστεί να διεκπεραιωθούν μέσα στα χρονικά πλαίσια που ορίστηκαν. Επιπλέον, με βάση αναλύσεις κόστους που έχουν γίνει για τις πιο πάνω λειτουργικότητες, έχει αποφασισθεί ότι βρισκόμαστε εντός προϋπολογισμού, δηλαδή εντός του budget που είναι διατεθειμένος να ξοδέψει ο πελάτης.

3.2.3 Εκλέπτυνση Ροών Δεδομένων

Παρακάτω θα παρουσιαστεί η εκλέπτυνση των ροών δεδομένων που εμφανίζονται στα ΔΡΔ. Με την εκλέπτυνση θα παρουσιάσουμε τον ακριβή καθορισμό των πληροφοριών που διακινούνται κατά την εκτέλεση των διάφορων διεργασιών.

Υπάρχουν έννοιες που υπάρχουν σε όλα τα ΔΡΔ όπως:

- **Αποτελέσματα:** Εμφάνιση των αποτελεσμάτων στην οθόνη μετά από αναζήτηση.
- **Λίστα:** Εμφάνιση των αποτελεσμάτων σε μορφή λίστας στην οθόνη μετά από αναζήτηση.
- **Μήνυμα Ορθότητας:** Πχ «Δώσατε λάθος στοιχεία».
- **Μήνυμα Επιτυχίας ή Αποτυχίας:**
Πχ «Τα δεδομένα αποθηκεύτηκαν» ή
«Τα δεδομένα δεν αποθηκεύτηκαν με επιτυχία».

3.2.3.1 Διαχείριση Ιδιοκτητών

Στοιχεία ιδιοκτήτη:

Κωδικός ιδιοκτήτη
Ονοματεπώνυμο
Φύλο
Ημ. Γέννησης
Αρ. Ταυτότητας
Τηλέφωνο Επικοινωνίας
Email
Fax
Διεύθυνση
Συνθηματικό

Κωδικός

Λειτουργίες:

Έλεγχος στοιχείων ιδιοκτήτη: Όλα τα στοιχεία του ιδιοκτήτη ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων ιδιοκτήτη: Όλα τα στοιχεία του ιδιοκτήτη εισάγονται στο αρχείο ιδιοκτητών και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων ιδιοκτήτη: Για την αναζήτηση, δίνονται κάποια στοιχεία του ιδιοκτήτη και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων ιδιοκτήτη: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία του ιδιοκτήτη ή και ολόκληρο τον ιδιοκτήτη αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων ιδιοκτήτη: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του ιδιοκτήτη, (αν πχ αλλάξει αρ. τηλεφώνου). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του ιδιοκτήτη και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του ιδιοκτήτη.

3.2.3.2 Διαχείριση Υπαλλήλων

Στοιχεία υπαλλήλου:

Κωδικός υπαλλήλου

Ονοματεπώνυμο

Φύλο

Ημερ. Γέννησης

Αρ. Ταυτότητας
Τηλέφωνο Επικοινωνίας
Email
Fax
Διεύθυνση
Συνθηματικό
Κωδικός

Λειτουργίες:

Έλεγχος στοιχείων υπαλλήλου: Όλα τα στοιχεία του υπαλλήλου ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων υπαλλήλου: Όλα τα στοιχεία του υπαλλήλου εισάγονται στο αρχείο υπαλλήλων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων υπαλλήλου: Για την αναζήτηση, δίνονται κάποια στοιχεία του υπαλλήλου και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων υπαλλήλου: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία του υπαλλήλου ή και ολόκληρο τον υπάλληλο αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων υπαλλήλου: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του υπαλλήλου, (αν πχ αλλάξει αρ. τηλεφώνου). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του υπαλλήλου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του υπαλλήλου.

3.2.3.3 Διαχείριση Κτηνιάτρων

Στοιχεία κτηνιάτρου:

Κωδικός υπαλλήλου

Ονοματεπώνυμο
Φύλο
Ημερ. Γέννησης
Αρ. Ταυτότητας
Τηλέφωνο Επικοινωνίας
Email
Fax
Διεύθυνση
Διεύθυνση Ιατρείου/Γραφείου

Λειτουργίες:

Έλεγχος στοιχείων κτηνιάτρου: Όλα τα στοιχεία του κτηνιάτρου ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων κτηνιάτρου: Όλα τα στοιχεία του κτηνιάτρου εισάγονται στο αρχείο κτηνιάτρων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων κτηνιάτρου: Για την αναζήτηση, δίνονται κάποια στοιχεία του κτηνιάτρου και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων κτηνιάτρου: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία του κτηνιάτρου ή και ολόκληρο τον κτηνίατρο αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων κτηνιάτρου: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του κτηνιάτρου, (αν πχ αλλάξει αρ. τηλεφώνου). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του κτηνιάτρου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του κτηνιάτρου.

3.2.3.4 Διαχείριση Προβάτων

Στοιχεία προβάτου:

Κωδικός προβάτων
Αριθμός προβάτου
Όνομα
Ημερ. Γέννησης
Φύλο
Γονότυπος
Βάρος
Βάρος μετά από 110 ημέρες
Κωδικός Φάρμας
Βάρος Κατανάλωσης τροφής
Κόστος Τροφής
Μετρήσεις Τροφής
Φωτογραφία
Κωδικός Μητέρας
Κωδικός Πατέρα
Είδος
Θηλυκά Ζώα: Παραγωγή Γάλακτος
Κόστος Γάλακτος
Αριθμός Γεννών
Μετρήσεις Γάλακτος

Λειτουργίες:

Έλεγχος στοιχείων προβάτου: Όλα τα στοιχεία του προβάτου ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων προβάτου: Όλα τα στοιχεία του προβάτου εισάγονται στο αρχείο προβάτων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων προβάτου: Για την αναζήτηση, δίνονται κάποια στοιχεία του προβάτου και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων προβάτου: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία του προβάτου ή και ολόκληρο το πρόβατο αν αυτό είναι απαραίτητο. Αρχικά θα

εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων προβάτου: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του προβάτου, (αν πχ αλλάξει αριθμό του προβάτου). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του προβάτου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του προβάτου.

3.2.3.5 Διαχείριση Φάρμας

Στοιχεία φάρμας:

Κωδικός φάρμας

Όνομα

Διεύθυνση

Τηλέφωνα Επικοινωνίας

Email

Fax

Κυβερνητική Άδεια

Λειτουργίες:

Έλεγχος στοιχείων φάρμας: Όλα τα στοιχεία της φάρμας ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων φάρμας: Όλα τα στοιχεία της φάρμας εισάγονται στο αρχείο φαρμών και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων φάρμας: Για την αναζήτηση, δίνονται κάποια στοιχεία της φάρμας και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων φάρμας: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία της φάρμας ή και ολόκληρο την φάρμα αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων φάρμας: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία της φάρμας, (αν πχ αλλάξει αρ.τηλεφώνου). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία της φάρμας και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης της φάρμας.

3.2.3.6 Διαχείριση Log File

Στοιχεία:

Όλα τα δεδομένα των πιο πάνω.

Λειτουργίες:

Εισαγωγή εγγραφής στο LOG FILE: Εισαγωγή στη βάση μίας νέας εγγραφής LOG FILE με το δικό της αύξον αριθμό.

Προβολή LOG FILE:

Φόρτωση LOG FILE

Ο διαχειριστής του συστήματος θα μπορεί να βλέπει το LOG FILE και να πληροφορείται ποιες αλλαγές έγιναν από το σύστημα και από ποιους. Έτσι σε περίπτωση σφάλματος ή χάσιμο δεδομένων, τα δεδομένα θα μπορούν να ανακτηθούν

Προβολή πιο πρόσφατων εγγράφων LOG FILE που έχουν σχέση με το λογισμικό διαχείρισης της ΒΔ: Ο διαχειριστής κατά την είσοδο του στο ΒΟ θα μπορεί να βλέπει τις πιο πρόσφατες εγγραφές που ενημερώθηκαν, και οι οποίες έγιναν από τη τελευταία φορά που είχε κάνει login στο ΒΟ

Επιλογή των κατάλληλων εγγράφων του log file με βάση τα κριτήρια των δεδομένων εισόδου.

3.2.3.7 Διαχείριση Ασθενειών

Στοιχεία Ασθένειας:

Κωδικός Ασθένειας

όνομα

Περιγραφή

Βάρος

Λειτουργίες:

Έλεγχος στοιχείων ασθένειας: Όλα τα στοιχεία της ασθένειας ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων ασθένειας: Όλα τα στοιχεία της ασθένειας εισάγονται στο αρχείο ασθενειών και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων ασθένειας: Για την αναζήτηση, δίνονται κάποια στοιχεία της ασθένειας και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Ενημέρωση στοιχείων ασθένειας: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία της ασθένειας, (αν πχ αλλάξει το βάρος). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία της ασθένειας και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης της ασθένειας.

3.2.3.8 Διαχείριση Φαρμάκων

Στοιχεία φαρμάκου:

Κωδικός φαρμάκου

Όνομα

Περιγραφή

Τιμή

Λειτουργίες:

Έλεγχος στοιχείων φαρμάκου: Όλα τα στοιχεία του φαρμάκου ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων φαρμάκου: Όλα τα στοιχεία του φαρμάκου εισάγονται στο αρχείο φαρμάκων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων φαρμάκου: Για την αναζήτηση, δίνονται κάποια στοιχεία του φαρμάκου και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Ενημέρωση στοιχείων φαρμάκου: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του φαρμάκου, (αν πχ αλλάξει η τιμή). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του φαρμάκου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του φαρμάκου.

3.2.3.9 Διαχείριση Εξετάσεων

Στοιχεία εξέτασης:

Κωδικός Εξέτασης
Ημερομηνία Εξέτασης
Κωδικός κτηνιάτρου
Κωδικός Ασθένειας
Κωδικός φαρμάκων
Σχόλια

Λειτουργίες:

Έλεγχος στοιχείων εξέτασης: Όλα τα στοιχεία της εξέτασης ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων εξέτασης: Όλα τα στοιχεία της εξέτασης εισάγονται στο αρχείο Εξετάσεων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων εξέτασης: Για την αναζήτηση, δίνονται κάποια στοιχεία της εξέτασης και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων εξέτασης: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία της εξέτασης ή και ολόκληρη την εξέταση αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων εξέτασης: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία της εξέτασης, (αν πχ αλλάξει ο κωδικός του κτηνιάτρου). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία της εξέτασης και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης της εξέτασης.

3.2.3.10 Διαχείριση Εσόδων

Στοιχεία εσόδου:

Κωδικός Εσόδου
Κωδικός Φάρμας
Ημερομηνία Καταχώρησης
Περιγραφή
Μονάδα Μέτρησης
Τιμή ανά μονάδα μέτρησης

Λειτουργίες:

Έλεγχος στοιχείων εσόδου: Όλα τα στοιχεία του εσόδου ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων εσόδου: Όλα τα στοιχεία του εσόδου εισάγονται στο αρχείο Εσόδων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων εσόδου: Για την αναζήτηση, δίνονται κάποια στοιχεία του εσόδου και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων εσόδου: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία του εσόδου ή και ολόκληρο το έσοδο αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων εσόδου: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του εσόδου, (αν πχ αλλάξει ο κωδικός της φάρμας). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του εσόδου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του εσόδου.

3.2.3.11 Διαχείριση Εξόδων

Στοιχεία εξόδου:

Κωδικός Εξόδου
Κωδικός Φάρμας
Ημερομηνία Καταχώρησης
Περιγραφή
Μονάδα Μέτρησης
Τιμή ανά μονάδα μέτρησης
Κατάσταση

Λειτουργίες:

Έλεγχος στοιχείων εξόδου: Όλα τα στοιχεία του εξόδου ελέγχονται. Στη συνέχεια βγάζει μήνυμα ορθότητας.

Εισαγωγή στοιχείων εξόδου: Όλα τα στοιχεία του εξόδου εισάγονται στο αρχείο Εξόδων και ακολουθεί ένα μήνυμα επιτυχίας ή αποτυχίας στον χρήστη.

Αναζήτηση στοιχείων εξόδου: Για την αναζήτηση, δίνονται κάποια στοιχεία του εξόδου και στην συνέχεια εμφανίζονται στην οθόνη τα αποτελέσματα.

Διαγραφή στοιχείων εξόδου: Ο χρήστης μπορεί να διαγράψει κάποια στοιχεία του εξόδου ή και ολόκληρο το έξοδο αν αυτό είναι απαραίτητο. Αρχικά θα εμφανίζεται μήνυμα προειδοποίησης και στην συνέχεια μήνυμα επιτυχίας της διαγραφής.

Ενημέρωση στοιχείων εξόδου: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει τα στοιχεία του εξόδου, (αν πχ αλλάξει ο κωδικός της φάρμας). Μετά τη τροποποίηση εμφανίζεται μήνυμα ότι η πράξη έγινε με επιτυχία.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του εξόδου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του εξόδου.

3.2.3.12 Διαχείριση Τιμολογίων

Στοιχεία τιμολογίων:

Κωδικός Τιμολογίου
Κωδικός Ιδιοκτήτη
Κωδικός Φάρμας
Κωδικός Εσόδου/Εξόδου
Ημερομηνία Συναλλαγής
Αριθμός Τεμαχίων
Συνολικό Ποσό

Λειτουργίες:

Αναζήτηση τιμολογίου: κατά την αναζήτηση του τιμολογίου, ο χρήστης δίνει τα μερικά ή όλα τα στοιχεία του τιμολογίου. Όλες οι πληροφορίες σχετικά με το συγκεκριμένο τιμολόγιο παίρνονται από το αρχείο τιμολογίων.

Εκτύπωση Τιμολογίων: κατά την εκτύπωση των τιμολογίων, δίνεται σαν είσοδος τα στοιχεία του τιμολογίου και τότε το τιμολόγιο δίνεται εκτυπωμένο στον χρήστη.

Αποστολή Μηνύματος Επικύρωσης: κατά την αποστολή μηνύματος επικύρωσης, δίνεται σαν είσοδος η λειτουργία η οποία προηγήθηκε και τα στοιχεία του τιμολογίου. Το μήνυμα αποστέλλεται στον χρήστη.

Διαγραφή Τιμολογίου: κατά την διαγραφή του τιμολογίου, μέσω της αναζήτησης το σύστημα παίρνει τα στοιχεία του τιμολογίου και διαγράφει τα στοιχεία από το Αρχείο Τιμολογίων.

Ενημέρωση Τιμολογίου: κατά την ενημέρωση του τιμολογίου, μέσω της αναζήτησης το σύστημα βρίσκει τα στοιχεία του τιμολογίου, παίρνει τα νέα στοιχεία του τιμολογίου, κάνει τις απαραίτητες αλλαγές και ενημερώνει τα στοιχεία από το Αρχείο Τιμολογίων.

Εισαγωγή Τιμολογίου: κατά την εισαγωγή νέου τιμολογίου, ο χρήστης δίνει τα νέα στοιχεία για το νέο τιμολόγιο και ενημερώνει το αρχείο τιμολογίων με το νέο τιμολόγιο.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία του τιμολογίου και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης του τιμολογίου.

3.2.3.13 Διαχείριση Αποδείξεων

Στοιχεία αποδείξεων:

Κωδικός Απόδειξης
Κωδικός Τιμολογίου
Πληρωτέο Ποσό
Ημερομηνία Συναλλαγής
Σχόλια

Λειτουργίες:

Αναζήτηση στοιχείων αποδείξεων: κατά την αναζήτηση των στοιχείων της απόδειξης , δίνεται σαν είσοδος ο κωδικός του τιμολογίου και με βάση αυτόν γίνεται αναζήτηση των στοιχείων της απόδειξης από το αρχείο Αποδείξεων.

Εκτύπωση Απόδειξης: κατά την εκτύπωση της απόδειξης, δίνεται σαν είσοδος τα στοιχεία της απόδειξης και τότε η απόδειξη δίνεται εκτυπωμένη στον χρήστη.

Αποστολή Μηνύματος Επικύρωσης: κατά την αποστολή μηνύματος επικύρωσης, δίνεται σαν είσοδος η λειτουργία η οποία προηγήθηκε και τα στοιχεία της απόδειξης. Το μήνυμα αποστέλλεται στον χρήστη.

Διαγραφή Απόδειξης: κατά την διαγραφή της απόδειξης, μέσω της αναζήτησης το σύστημα παίρνει τα στοιχεία της απόδειξης και διαγράφει τα στοιχεία από το Αρχείο Αποδείξεων.

Ενημέρωση Απόδειξης: κατά την ενημέρωση της απόδειξης, μέσω της αναζήτησης το σύστημα βρίσκει τα στοιχεία της απόδειξης, παίρνει τα νέα στοιχεία της απόδειξης , κάνει τις απαραίτητες αλλαγές και ενημερώνει τα στοιχεία από το Αρχείο Αποδείξεων.

Εισαγωγή Απόδειξης: κατά την εισαγωγή νέας απόδειξης, ο χρήστης δίνει τα νέα στοιχεία για την νέα απόδειξη και ενημερώνει το αρχείο αποδείξεων με την νέα απόδειξη.

Δημιουργία καταστατικού: κατά την δημιουργία καταστατικού, δίνεται σαν είσοδος τα στοιχεία της απόδειξης και αυτά δίνονται σαν έξοδος στην διαδικασία εκτύπωσης της απόδειξης.

3.2.3.14 Διαχείριση Παραγωγής Γάλακτος

Στοιχεία παραγωγής γάλακτος:

Κωδικός Προβάτου

Ποσότητα Γάλακτος

Ημερομηνία καταγραφής

Λειτουργίες:

Θα υπάρχει δυνατότητα εισαγωγής νέας παραγωγής γάλακτος στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την παραγωγή γάλακτος είναι το πρόβατο το οποίο αφορά η συγκεκριμένη παραγωγή, γι' αυτό και ο χρήστης θα πρέπει να έχει προηγουμένως επιλέξει το αντίστοιχο πρόβατο, η ποσότητα γάλακτος για το αντίστοιχο πρόβατο και η ημερομηνία καταχώρησης της παραγωγής.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της παραγωγής γάλακτος.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της παραγωγής στη βάση δεδομένων.

θα υπάρχει δυνατότητα ομαδοποίησης των προβάτων, ανάλογα με την θέση που παίρνει το πρόβατο στο γαλευτήρι και θα παρουσιάζεται η δυνατότητα εισαγωγής του με γραφικό τρόπο, για να διευκολύνεται ο ιδιοκτήτης/υπάλληλος όταν εισάγει την ποσότητα γάλακτος του αντίστοιχου προβάτου.

Θα υπάρχει δυνατότητα εκτύπωσης αναφοράς παραγωγής γάλακτος του συστήματος διαχείρισης της κτηνοτροφικής μονάδας.

Ο χρήστης θα έχει την δυνατότητα επιλογής εκτύπωσης της αναφοράς για όλα τα πρόβατα μαζί (συλλογικά), και την δυνατότητα για κάθε πρόβατο ξεχωριστά έτσι ώστε να μπορεί να διακρίνει την πρόοδο της γαλακτοπαραγωγής.

3.2.3.15 Διαχείριση Κατανάλωσης Τροφής

Στοιχεία Κατανάλωσης Τροφής:

Κωδικός Προβάτου

Ποσότητα Τροφής

Ημερομηνία καταγραφής

Λειτουργίες:

Θα υπάρχει δυνατότητα εισαγωγής νέας κατανάλωσης τροφής στο σύστημα διαχείρισης της κτηνοτροφικής μονάδας.

Οι πληροφορίες που πρέπει να εισάγονται για την κατανάλωση τροφής είναι το πρόβατο το οποίο αφορά η συγκεκριμένη κατανάλωση, γι' αυτό και ο χρήστης θα πρέπει να έχει προηγουμένως επιλέξει το αντίστοιχο πρόβατο, η ποσότητα τροφής για το αντίστοιχο πρόβατο και η ημερομηνία καταχώρησης της κατανάλωσης.

Τα δεδομένα που εισάγονται θα ελέγχονται αν είναι έγκυρα. Αν τα δεδομένα είναι έγκυρα τότε θα γίνεται εισαγωγή στη βάση δεδομένων των στοιχείων της κατανάλωσης τροφής.

Εκτύπωση στην οθόνη του υπολογιστή ενός μηνύματος που θα βεβαιώνει τη πρόσθεση της κατανάλωσης στη βάση δεδομένων.

θα υπάρχει δυνατότητα ομαδοποίησης των προβάτων, ανάλογα με την θέση που παίρνει το πρόβατο στην φάρμα και θα παρουσιάζεται η δυνατότητα εισαγωγής του με γραφικό τρόπο, για να διευκολύνεται ο ιδιοκτήτης/υπάλληλος όταν εισάγει την κατανάλωση τροφής του αντίστοιχου προβάτου.

Θα υπάρχει δυνατότητα εκτύπωσης αναφοράς κατανάλωσης τροφής του συστήματος διαχείρισης της κτηνοτροφικής μονάδας.

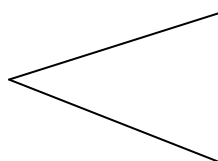
Ο χρήστης θα έχει την δυνατότητα επιλογής εκτύπωσης της αναφοράς για όλα τα πρόβατα μαζί (συλλογικά), και την δυνατότητα για κάθε πρόβατο ξεχωριστά έτσι ώστε να μπορεί να διακρίνει την πρόοδο της κατανάλωσης.

3.2.4 Προσδιορισμός Λογικής των διαδικασιών

Υπάρχουν κάποιες λειτουργίες στο σύστημα που πρέπει να επικοινωνούν με τη βάση. Ο λόγος που χρειάζεται ο προσδιορισμός της λογικής είναι επειδή θα πρέπει να μας εμφανίζονται μηνύματα αποτυχίας ή επιτυχίας για τις λειτουργίες αυτές. Παρακάτω παρουσιάζεται η λογική των λειτουργιών που χρειάζονται να επικοινωνούν με τη βάση.

Λειτουργίες:

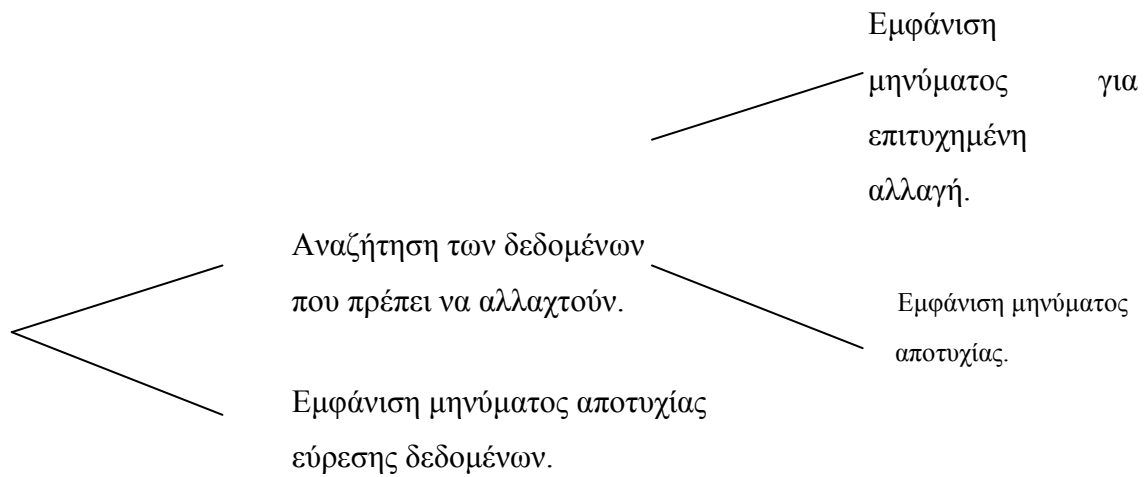
Εισαγωγή δεδομένων στο σύστημα:



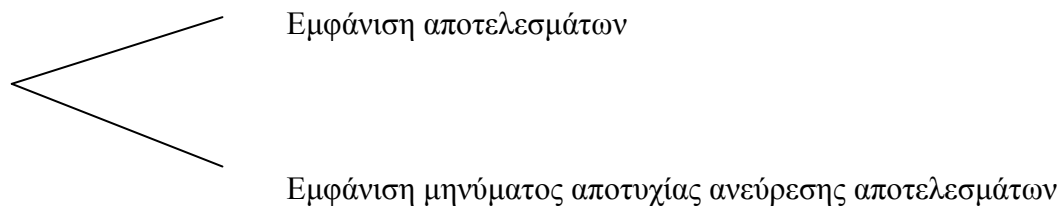
Εμφάνιση μηνύματος για επιτυχημένη εισαγωγή στο σύστημα μετά από έλεγχο που έγινε στη βάση.

Εμφάνιση μηνύματος για αποτυχημένη εισαγωγή στο σύστημα μετά από έλεγχο που έγινε στη βάση.

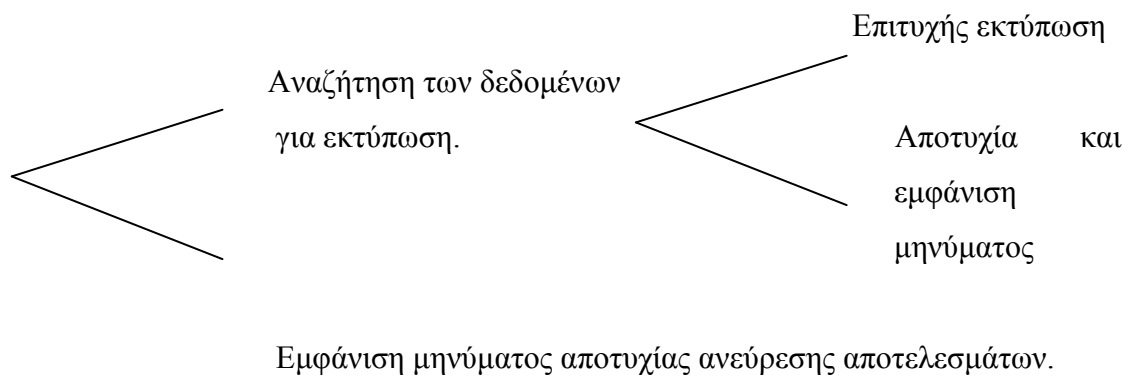
Αλλαγή δεδομένων του χρήστη του συστήματος:



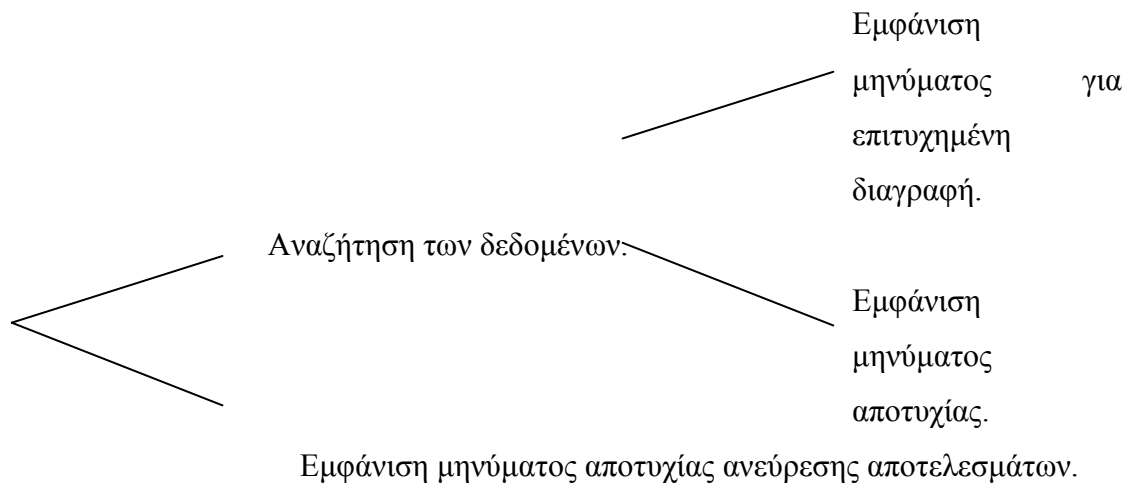
Αναζήτηση δεδομένων του συστήματος:



Εκτύπωση δεδομένων του συστήματος:



Διαγραφή δεδομένων του συστήματος:



3.2.5 Προσδιορισμός Αποθήκευσης Δεδομένων

Στην ενότητα αυτή θα παρουσιάσουμε τα ακριβή περιεχόμενα κάθε αποθηκευτικής μονάδας που βρίσκεται στην βάση δεδομένων του συστήματος που θα αναπτυχθεί σε μορφή πινάκων. Θα καθορίσουμε επίσης και το που απαιτείται άμεση πρόσβαση.

EXAMINATION	
	examination_id
	Condition
	examin_date
	vet_id
	sheep_id
	comments

MEDICINE	
	medicine_id
	Name
	Description
	price

ILLNESS	
	illness_id
	Name
	Description
	ill_weight

FARM	
	farm_id
	condition
	name
	AddressNum
	AddressStreet
	PostalCode
	AddressTown
	AddressCountry
	Phone
	Mobile
	Fax
	Email
	gov_licence

INVOICE_IN	
	invoice_id
	owner_id
	farm_id
	incoming_id
	inv_date
	num_of_units
	total_amount
	amount_paid
	amount_remaining
	Condition

INVOICE_OUT	
	invoice_id
	owner_id
	farm_id
	outgoing_id
	inv_date
	num_of_units
	total_amount
	amount_paid
	amount_remaining
	Condition

RECEIPT_OUT	
	receipt_id
	invoice_id
	amount_cur_given
	description
	rec_date
	Condition

RECEIPT_IN	
	receipt_id
	invoice_id
	amount_cur_given
	description
	rec_date
	Condition

INCOMING	
	incoming_id
	farm_id
	incom_date
	incom_description
	incom_code
	incom_unit
	Condition
	incom_price_per_unit

OUTGOING	
	outgoing_id
	farm_id
	outg_date
	outg_description
	outg_code
	outg_unit
	outg_price_per_unit
	Condition
	outg_state

SHEEP	
	sheep_id
	animal_num
	Name
	Birthdate
	Sex
	Genotype
	WeightInKilos
	WeightInKilos2
	farm_id
	food_consumption
	food_cost
	Condition
	sheep_img
	mother_id
	father_id
	Team
	src_img
	kind
	food_count

VETERINARIAN	
	vet_id
	Condition
	ID
	Fname
	Lname
	Sex
	BirthDate
	AddressNum
	AddressStreet
	PostalCode
	AddressTown
	AddressCountry
	Phone
	Mobile
	Fax
	Email
	OfficeAddressNum
	OfficeAddressStreet
	OfficePostalCode
	OfficeAddressTown
	OfficeAddressCountry

OWNERS	
	owner_id
	Condition
	ID
	Fname
	Lname
	Sex
	BirthDate
	AddressNum
	AddressStreet
	PostalCode
	AddressTown
	AddressCountry
	Phone
	Mobile
	Fax
	Email

FEMALE_SHEEP	
	sheep_id
	milk_production
	milk_cost
	num_of_birth
	milk_count

DAILY_FOOD_CONSUMPTION	
	daily_date
	sheep_id
	food

SHEEP_BIRTH	
	sheep_birth_id
	sheep_id
	num_of_child
	birth_date
	Condition

DAILY_MILK_PRODUCTION	
	daily_date
	sheep_id
	milk

EMPLOYEE	
	employee_id
	Condition
	ID
	Fname
	Lname
	Sex
	BirthDate
	AddressNum
	AddressStreet
	PostalCode
	AddressTown
	AddressCountry
	Phone
	Mobile
	Fax
	Email

3.2.6 Καθορισμός φυσικών πόρων

Όλα τα δεδομένα του νέου συστήματος θα αποθηκεύονται στην βάση δεδομένων. Συγκεκριμένα, το μέσο αποθήκευσης είναι ο σκληρός δίσκος του ηλεκτρονικού υπολογιστή που θα χρησιμοποιείται, αφού δεν υπάρχει δίκτυο για να κριθεί αναγκαία η χρήση εξυπηρετητή δικτύου (server) για την βάση δεδομένων. Τα δεδομένα αυτά θα αποθηκεύονται σε κατάλληλους πίνακες που θα υπάρχουν στην βάση, ανάλογα με το τι αντιπροσωπεύουν.

Η βάση δεδομένων θα αναπτυχθεί με το λογισμικό Microsoft SQL Server 2008 Management Studio. Πιο κάτω φαίνονται τα μεγέθη των τύπων που θα χρησιμοποιηθούν.

Πίνακας που δείχνει το μέγεθος κάθε τύπου:

Πίνακας 3.1

Τύπος Δεδομένων	Απαιτούμενη Μνήμη (bytes)
bit	1
datetime	8
int	4
Decimal(p, s)	5
Nvarchar(n)	2n+2
image	$2^{31} - 1$
Varbinary(max)	L+2
char	1
Varchar(n)	n+1

Πιο κάτω παρουσιάζονται οι πίνακες μαζί με τα πρωτεύοντα κλειδιά τους. Το μέγεθος που παρουσιάζεται είναι το μέγιστο που μπορεί να χρησιμοποιηθεί σε περίπτωση Nvarchar(n). Σε κάθε πίνακα υπάρχουν ακόμη δύο επιπρόσθετες μεταβλητές για τον έλεγχο ασφάλειας.

Πίνακας 3.2: LOG IN

Όνομα αρχείου: LOG_IN				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός χρήστη	integer	4		Not null
Όνομα χρήστη	Nvarchar(15)	32		Not null
Συνθηματικό χρήστη	Nvarchar(15)	32	Primary key	Not null
Σύνολο		68 bytes		

Πίνακας 3.3: Ιδιοκτήτης

Όνομα Αρχείου: Ιδιοκτήτης				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/not null
Κωδικός ιδιοκτήτη	integer	4	primary	Not null
Κατάσταση	Char(1)	1		Not null
Ταυτότητα	Nvarchar(20)	42	unique	Not null
Όνομα	Nvarchar(50)	102		Not null
Επίθετο	Nvarchar(50)	102		Not null
Φύλο	Char(1)	1		null
Ημερ. Γέννησης	smalldatetime	4		null
Αρ. διεύθυνσης	Nvarchar(10)	22		null
Οδός διεύθυνσης	Nvarchar(100)	202		null
Ταχυδρομικός κώδικας	Nvarchar(20)	42		null
Πόλη διεύθυνσης	Nvarchar(30)	62		null
Χώρα διεύθυνσης	Nvarchar(30)	62		null
Σταθερό Τηλέφωνο	varchar(20)	21	unique	Not null
Κινητό τηλέφωνο	Varchar(20)	21		Not null
Fax	varchar(20)	21		null
Email	Nvarchar(150)	302		null
Σύνολο	1011 bytes			

Πίνακας 3.4: Εργοδοτούμενος

Όνομα Αρχείου: Εργοδοτούμενος				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/not null
Κωδικός ιδιοκτήτη	integer	4	primary	Not null
Κατάσταση	Char(1)	1		Not null
Ταυτότητα	Nvarchar(20)	42	unique	Not null
Όνομα	Nvarchar(50)	102		Not null
Επίθετο	Nvarchar(50)	102		Not null
Φύλο	Char(1)	1		null
Ημερ. Γέννησης	smalldatetime	4		null
Αρ. διεύθυνσης	Nvarchar(10)	22		null
Οδός διεύθυνσης	Nvarchar(100)	202		null
Ταχυδρομικός κώδικας	Nvarchar(20)	42		null
Πόλη διεύθυνσης	Nvarchar(30)	62		null
Χώρα διεύθυνσης	Nvarchar(30)	62		null
Σταθερό Τηλέφωνο	varchar(20)	21	unique	Not null
Κινητό τηλέφωνο	Varchar(20)	21		Not null
Fax	varchar(20)	21		null
Email	Nvarchar(150)	302		null
Σύνολο	1011 bytes			

Πίνακας 3.5: Κτηνίατρος

Όνομα Αρχείου: Κτηνίατρος				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/not null
Κωδικός ιδιοκτήτη	integer	4	primary	Not null
Κατάσταση	Char(1)	1		Not null
Ταυτότητα	Nvarchar(20)	42	unique	Not null
Όνομα	Nvarchar(50)	102		Not null
Επίθετο	Nvarchar(50)	102		Not null
Φύλο	Char(1)	1		null
Ημερ. Γέννησης	smalldatetime	4		null
Αρ. διεύθυνσης	Nvarchar(10)	22		null
Οδός διεύθυνσης	Nvarchar(100)	202		null
Ταχυδρομικός κώδικας	Nvarchar(20)	42		null
Πόλη διεύθυνσης	Nvarchar(30)	62		null
Χώρα διεύθυνσης	Nvarchar(30)	62		null
Σταθερό Τηλέφωνο	varchar(20)	21	unique	Not null
Κινητό τηλέφωνο	Varchar(20)	21		Not null
Fax	varchar(20)	21		null
Email	Nvarchar(150)	302		null
Διεύθυνση Ιατρείου				
Αρ. διεύθυνσης	Nvarchar(10)	22		null
Οδός διεύθυνσης	Nvarchar(100)	202		null
Ταχυδρομικός κώδικας	Nvarchar(20)	42		null
Πόλη διεύθυνσης	Nvarchar(30)	62		null
Χώρα διεύθυνσης	Nvarchar(30)	62		null
Σύνολο		1401 bytes		

Πίνακας 3.6: Ημερήσια παραγωγή γάλακτος

Όνομα αρχείου: Ημερήσια Παραγωγή Γάλακτος				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Ημερομηνία	smalldatetime	4	Primary key	Not null
Κωδικός Προβάτου	int	4	Primary key	Not null
Ποσότητα Γάλακτος	Decimal(18,0)	5		Not null
Σύνολο		13 bytes		

Πίνακας 3.7: Ημερήσια κατανάλωση τροφής

Όνομα αρχείου: Ημερήσια Κατανάλωση Τροφής				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Ημερομηνία	smalldatetime	4	Primary key	Not null
Κωδικός Προβάτου	int	4	Primary key	Not null
Ποσότητα Τροφής	Decimal(18,0)	5		Not null
Σύνολο	13 bytes			

Πίνακας 3.8: Εξέταση

Όνομα αρχείου: Εξέταση				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Εξέτασης	int	4	Primary key	Not null
Κατάσταση	Char(1)	1		Not null
Ημερομηνία	smalldatetime	4		null
Κωδικός Κτηνιάτρου	Int	4	Foreign key	Not null
Κωδικός Προβάτου	Int	4	Foreign key	Not null
Σχόλια	Nvarchar(400)	802		null
Σύνολο	819 bytes			

Πίνακας 3.9: Φάρμα

Όνομα Αρχείου: Φάρμα				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/not null
Κωδικός φάρμας	integer	4	primary	Not null
Κατάσταση	Char(1)	1		Not null
Όνομα	Nvarchar(30)	102		null
Αρ. διεύθυνσης	Nvarchar(10)	22		null
Οδός διεύθυνσης	Nvarchar(100)	202		null
Ταχυδρομικός κώδικας	Nvarchar(20)	42		null
Πόλη διεύθυνσης	Nvarchar(30)	62		null
Χώρα διεύθυνσης	Nvarchar(30)	62		null
Σταθερό Τηλέφωνο	varchar(20)	21	unique	null
Κινητό τηλέφωνο	Varchar(20)	21		null
Fax	varchar(20)	21		null
Email	Nvarchar(150)	302		null
Κυβερνητική άδεια	Nvarchar(150)	302		null
Σύνολο	1164 bytes			

Πίνακας 3.10: θηλυκό πρόβατο

Όνομα αρχείου: Θηλυκό Πρόβατο				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Προβάτου	int	4	Primary key	Not null
Παραγωγή Γάλακτος	Decimal(10,3)	5		null
Κόστος Γάλακτος	Decimal(10,3)	5		null
Αριθμός Γεννών	int	4		null
Μετρήσεις Γάλακτος	int	4		null
Σύνολο	22 bytes			

Πίνακας 3.11: Ασθένεια

Όνομα αρχείου: Ασθένεια				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Ασθένειας	int	4	Primary key	Not null
Όνομα	Nvarchar(50)	102		Not null
Περιγραφή	Nvarchar(400)	802		null
Βάρος	int	4		null
Σύνολο	912 bytes			

Πίνακας 3.12: Φάρμακο

Όνομα αρχείου: Φάρμακο				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Φαρμάκου	int	4	Primary key	Not null
Όνομα	Nvarchar(50)	102		Not null
Περιγραφή	Nvarchar(400)	802		null
Τιμή	real	4		null
Σύνολο	912 bytes			

Πίνακας 3.13: Έσοδο

Όνομα αρχείου: Έσοδο				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Εσόδου	int	4	Primary key	Not null
Κωδικός Φάρμας	int	4	Foreign key	Not null
Ημερομηνία	smalldatetime	4		null
Περιγραφή	Nvarchar(100)	202		null
Μονάδα Μέτρησης	Nvarchar(10)	22		null
Κατάσταση	Char(1)	1		Not null
Τιμή ανά μονάδα μέτρησης	Decimal(6, 2)	5		null
Σύνολο	243 bytes			

Πίνακας 3.14: Έξοδο

Όνομα αρχείου: Έξοδο				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Εξόδου	int	4	Primary key	Not null
Κωδικός Φάρμας	int	4	Foreign key	Not null
Ημερομηνία	smalldatetime	4		null
Περιγραφή	Nvarchar(100)	202		null
Μονάδα Μέτρησης	Nvarchar(10)	22		null
Κατάσταση	Char(1)	1		Not null
Τιμή ανά μονάδα μέτρησης	Decimal(6, 2)	5		null
Είδος	smallint	2		null
Σύνολο	245 bytes			

Πίνακας 3.15: Τιμολόγιο

Όνομα αρχείου: Τιμολόγιο				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Τιμολογίου	Int	4	Primary Key	Not null
Κωδικός Ιδιοκτήτη	Int	4	Foreign Key	Not null
Κωδικός Φάρμας	Int	4	Foreign Key	Not null
Κωδικός Εσόδου/Εξόδου	Int	4	Foreign Key	Not null
Ημερομηνία	Smalldatetime	4		Not null
Αριθμός Τεμαχίων	Int	4		null
Συνολικό Ποσό	Decimal(6,2)	5		Not null
Πληρωμένο Ποσό	Decimal(6,2)	5		null
Υπολειπόμενο Ποσό	Decimal(6,2)	5		null
Κατάσταση	Char(1)	1		Not null
Σύνολο		40 bytes		

Πίνακας 3.16: Απόδειξη

Όνομα αρχείου: Απόδειξη				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Απόδειξης	int	4	Primary key	Not null
Κωδικός Τιμολογίου	int	4	Foreign key	Not null
Ημερομηνία	smalldatetime	4		null
Περιγραφή	Nvarchar(400)	802		null
Πληρωτέο Ποσό	Decimal(6,2)	5		Not null
Κατάσταση	Char(1)	1		Not null
Σύνολο		820 bytes		

Πίνακας 3.17: Πρόβατο

Όνομα Αρχείου: Πρόβατο				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/not null
Κωδικός προβάτου	int	4	Primary key	Not null
Αριθμός Προβάτου	int	4	unique	Not null
Όνομα	Nvarchar(35)	72		Not null
Ημερ. Γέννησης	smalldatetime	4		Not null
Φύλο	Char(1)	1		Not null
Γονότυπος	Char(2)	2		null
Βάρος	Decimal(7,3)	5		null
Βάρος μετά από 100 μέρες	Decimal(7,3)	5		null
Κωδικός Φάρμας	int	4	Foreign key	null
Κατανάλωση Τροφής	Decimal(10,3)	5		null
Κόστος Τροφής	Decimal(10,3)	5		null
Κατάσταση	Char(1)	1		Not null
Θέση Εικόνας	nvarchar(150)	302	unique	Not null
Κωδικός μητέρας	int	4	Foreign key	Not null
Κωδικός πατέρα	int	4	Foreign key	Not null
Ομάδα	Char(1)	1		null
Εικόνα	Varbinary(max)	L+2		null
Είδος	Nvarchar(5)	12		null
Μετρήσεις τροφής	int	4		null
Σύνολο	441+L bytes			

Πίνακας 3.18: Γέννες Προβάτου

Όνομα αρχείου: Γέννες Προβάτου				
Όνομα πεδίου	Τύπος	Μέγεθος	Κλειδί	Null/Not Null
Κωδικός Γέννας	int	4	Primary key	Not null
Κωδικός Προβάτου	int	4	Foreign key	Not null
Ημερομηνία Γέννησης	smalldatetime	4		null
Αριθμός Παιδιών	int	802		Not null
Κατάσταση	Char(1)	1		Not null
Σύνολο	815 bytes			

3.2.7 Ορισμός προδιαγραφών για Είσοδο/Εξοδο δεδομένων

Η δημιουργία ενός απλού, εύχρηστου και φιλικού περιβάλλοντος είναι ένας από τους πρωταρχικούς στόχους για την ανάπτυξη ενός λογισμικού συστήματος. Έχοντας αυτό υπόψη, γίνεται η ανάπτυξη της διεπιφάνειας των χρηστών βασισμένη στις

προδιαγραφές και λειτουργίες του υπό-ανάπτυξη συστήματος. Παρακάτω παρουσιάζονται μερικές από τις οθόνες διεπιφάνειας των χρηστών για είσοδο και έξοδο δεδομένων από το λογισμικό σύστημα.

3.2.7.1 Φόρμα Διαχείρισης Ιδιοκτητών

3.2.7.1.1 Εισαγωγή Ιδιοκτήτη

Περιγραφή	Ο χρήστης μπορεί να εισάγει νέο ιδιοκτήτη στο σύστημα δίνοντας τα στοιχεία του.
Είσοδος	Κωδικός ιδιοκτήτη, Ονοματεπώνυμο, Φύλο, Ημ. Γέννησης, Αρ. Ταυτότητας, Τηλέφωνο Επικοινωνίας, Email, Fax, Διεύθυνση, Συνθηματικό, Κωδικός
Έξοδος	Αν όλα τα στοιχεία εισήχθησαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.1.2 Αναζήτηση Ιδιοκτήτη

Περιγραφή	Ο χρήστης έχει την δυνατότητα να αναζητήσει έναν υπάρχων ιδιοκτήτη από το σύστημα.
Είσοδος	Μπορεί να το αναζητήσει με οποιοδήποτε από τα πιο πάνω στοιχεία.
Έξοδος	Αποτελέσματα αναζήτησης

3.2.7.1.3 Ενημέρωση Ιδιοκτήτη

Περιγραφή	Ο χρήστης έχει την δυνατότητα να ενημερώσει τα στοιχεία από έναν υπάρχων ιδιοκτήτη από το σύστημα.
Είσοδος	Μπορεί να τροποποιήσει μερικά ή όλα από τα πιο πάνω στοιχεία.
Έξοδος	Αν όλα τα στοιχεία ενημερώθηκαν σωστά τότε εμφανίζεται ένα μήνυμα

που ενημερώνει για την επιτυχή τροποποίηση και αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.1.4 Διαγραφή Ιδιοκτήτη

Περιγραφή	Ο χρήστης έχει την δυνατότητα να διαγράψει τα στοιχεία από έναν υπάρχων ιδιοκτήτη του συστήματος.
Είσοδος	Ο ιδιοκτήτης για διαγραφή θα βρίσκεται ήδη επιλεγμένος πάνω στην φόρμα αφού θα έχει προηγηθεί η αναζήτηση του.
Έξοδος	Αν όλα στοιχεία διαγράφηκαν με επιτυχία τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή διαγραφή των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.2 Φόρμα Διαχείρισης Υπαλλήλων

3.2.7.2.1 Εισαγωγή Υπαλλήλου

Περιγραφή	Ο χρήστης μπορεί να εισάγει νέο υπάλληλο στο σύστημα δίνοντας τα στοιχεία του.
Είσοδος	Κωδικός υπαλλήλου, Ονοματεπώνυμο, Φύλο, Ημ. Γέννησης, Αρ. Ταυτότητας, Τηλέφωνο Επικοινωνίας, Email, Fax, Διεύθυνση, Συνθηματικό, Κωδικός
Έξοδος	Αν όλα τα στοιχεία εισήχθησαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.2.2 Αναζήτηση Υπαλλήλου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να αναζητήσει έναν υπάρχων υπάλληλο από το σύστημα.
Είσοδος	Μπορεί να το αναζητήσει με οποιοδήποτε από τα πιο πάνω στοιχεία.

Έξοδος	Αποτελέσματα αναζήτησης
--------	-------------------------

3.2.7.2.3 Ενημέρωση Υπαλλήλου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να ενημερώσει τα στοιχεία από έναν υπάρχων υπάλληλου από το σύστημα.
Είσοδος	Μπορεί να τροποποιήσει μερικά ή όλα από τα πιο πάνω στοιχεία.
Έξοδος	Αν όλα τα στοιχεία ενημερώθηκαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή τροποποίηση και αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.2.4 Διαγραφή Υπαλλήλου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να διαγράψει τα στοιχεία από έναν υπάρχων υπάλληλο του συστήματος.
Είσοδος	Ο υπάλληλος για διαγραφή θα βρίσκεται ήδη επιλεγμένος πάνω στην φόρμα αφού θα έχει προηγηθεί η αναζήτηση του.
Έξοδος	Αν όλα στοιχεία διαγράφηκαν με επιτυχία τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή διαγραφή των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.3 Φόρμα Διαχείρισης Προβάτων

3.2.7.3.1 Εισαγωγή Προβάτου

Περιγραφή	Ο χρήστης μπορεί να εισάγει νέο πρόβατο στο σύστημα δίνοντας τα στοιχεία του.
Είσοδος	Κωδικός προβάτου, αριθμός προβάτου, όνομα, Φύλο, Ημ. Γέννησης, Γονότυπος, βάρος, βάρος 110 ημερών, κωδικός φάρμας, κατανάλωση

	τροφής, κόστος τροφής, μετρήσεις τροφής, εικόνα, κωδικός μητέρας, κωδικός πατέρα, είδος, (θηλυκά ζώα: παραγωγή γάλακτος, κόστος γάλακτος, αριθμός γεννών, μετρήσεις γάλακτος)
Έξοδος	Αν όλα τα στοιχεία εισήχθησαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.3.2 Αναζήτηση Προβάτου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να αναζητήσει έναν υπάρχων πρόβατο από το σύστημα.
Είσοδος	Μπορεί να το αναζητήσει με οποιοδήποτε από τα πιο πάνω στοιχεία.
Έξοδος	Αποτελέσματα αναζήτησης

3.2.7.3.3 Ενημέρωση Προβάτου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να ενημερώσει τα στοιχεία από ένα υπάρχων πρόβατο από το σύστημα.
Είσοδος	Μπορεί να τροποποιήσει μερικά ή όλα από τα πιο πάνω στοιχεία.
Έξοδος	Αν όλα τα στοιχεία ενημερώθηκαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή τροποποίηση και αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.3.4 Διαγραφή Προβάτου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να διαγράψει τα στοιχεία από ένα υπάρχων πρόβατο του συστήματος.
Είσοδος	Το πρόβατο για διαγραφή θα βρίσκεται ήδη επιλεγμένο πάνω στην φόρμα αφού θα έχει προηγηθεί η αναζήτηση του.
Έξοδος	Αν όλα στοιχεία διαγράφηκαν με επιτυχία τότε εμφανίζεται ένα μήνυμα

που ενημερώνει για την επιτυχή διαγραφή των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.4 Φόρμα Διαχείρισης Τιμολογίων

3.2.7.4.1 Εισαγωγή Τιμολογίου

Περιγραφή	Ο χρήστης μπορεί να εισάγει νέο τιμολόγιο στο σύστημα δίνοντας τα στοιχεία του.
Είσοδος	Κωδικός τιμολογίου, κωδικός ιδιοκτήτη, κωδικός φάρμας, κωδικός εσόδου/εξόδου, ημερομηνία συναλλαγής, αριθμός τεμαχίων, συνολικό ποσό
Έξοδος	Αν όλα τα στοιχεία εισήχθησαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.4.2 Αναζήτηση Τιμολογίου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να αναζητήσει ένα υπάρχων τιμολόγιο από το σύστημα.
Είσοδος	Μπορεί να το αναζητήσει με οποιοδήποτε από τα πιο πάνω στοιχεία.
Έξοδος	Αποτελέσματα αναζήτησης

3.2.7.4.3 Ενημέρωση Τιμολογίου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να ενημερώσει τα στοιχεία από ένα υπάρχων τιμολόγιο από το σύστημα.
Είσοδος	Μπορεί να τροποποιήσει μερικά ή όλα από τα πιο πάνω στοιχεία.
Έξοδος	Αν όλα τα στοιχεία ενημερώθηκαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή τροποποίηση και αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.4.4 Διαγραφή Τιμολογίου

Περιγραφή	Ο χρήστης έχει την δυνατότητα να διαγράψει τα στοιχεία από ένα υπάρχων τιμολόγιο του συστήματος.
Είσοδος	Το τιμολόγιο για διαγραφή θα βρίσκεται ήδη επιλεγμένο πάνω στην φόρμα αφού θα έχει προηγηθεί η αναζήτηση του.
Έξοδος	Αν όλα στοιχεία διαγράφηκαν με επιτυχία τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή διαγραφή των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.5 Φόρμα Διαχείρισης Απόδειξεων

3.2.7.5.1 Εισαγωγή Απόδειξης

Περιγραφή	Ο χρήστης μπορεί να εισάγει νέα απόδειξη στο σύστημα δίνοντας τα στοιχεία του.
Είσοδος	Κωδικός απόδειξης, κωδικός τιμολογίου, , ημερομηνία συναλλαγής, πληρωτέο ποσό, σχόλια
Έξοδος	Αν όλα τα στοιχεία εισήχθησαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.7.5.2 Αναζήτηση Απόδειξης

Περιγραφή	Ο χρήστης έχει την δυνατότητα να αναζητήσει μία υπάρχουσα απόδειξη από το σύστημα.
Είσοδος	Μπορεί να το αναζητήσει με οποιοδήποτε από τα πιο πάνω στοιχεία.
Έξοδος	Αποτελέσματα αναζήτησης

3.2.7.5.3 Ενημέρωση Απόδειξης

Περιγραφή	Ο χρήστης έχει την δυνατότητα να ενημερώσει τα στοιχεία από μία υπάρχουσα απόδειξη του συστήματος.
Είσοδος	Μπορεί να τροποποιήσει μερικά ή όλα από τα πιο πάνω στοιχεία.

Έξοδος	Αν όλα τα στοιχεία ενημερώθηκαν σωστά τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή τροποποίηση και αποθήκευση των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.
--------	--

3.2.7.5.4 Διαγραφή Απόδειξης

Περιγραφή	Ο χρήστης έχει την δυνατότητα να διαγράψει τα στοιχεία από μία υπάρχουσα απόδειξη του συστήματος.
Είσοδος	Η απόδειξη για διαγραφή θα βρίσκεται ήδη επιλεγμένη πάνω στην φόρμα αφού θα έχει προηγηθεί η αναζήτηση της.
Έξοδος	Αν όλα στοιχεία διαγράφηκαν με επιτυχία τότε εμφανίζεται ένα μήνυμα που ενημερώνει για την επιτυχή διαγραφή των στοιχείων. Διαφορετικά εμφανίζεται ένα μήνυμα αποτυχίας.

3.2.8 Εκτίμηση μεγέθους

Στο στάδιο αυτό υπολογίζεται ο όγκος δεδομένων του κάθε πίνακα, καθώς και ο όγκος δεδομένων εισόδου την πρώτη φορά που θα λειτουργήσει το σύστημα.

Πιο κάτω παρουσιάζονται τα βασικά μεγέθη που αφορούν τα δεδομένα της βάσης, όπως αυτά έχουν υπολογιστεί πιο πάνω, συγκεκριμένα στην παράγραφο 2.6, με σκοπό να παρουσιαστεί μία πιθανή ένδειξη της ποσότητας μνήμης που θα χρησιμοποιηθεί από την βάση δεδομένων.

Πίνακας 3.19

A/A	Όνομα Πίνακα	Μνήμη ανά εγγραφή (bytes)
1	LOG IN	68
2	Ιδιοκτήτης	1011
3	Υπάλληλος	1011
4	Κτηνίατρος	1401
5	Παραγωγή Γάλακτος	13
6	Κατανάλωση Τροφής	13
7	Εξέταση	819
8	Φάρμα	1164
9	Θηλυκό Πρόβατο	22
10	Ασθένεια	912
11	Φάρμακο	912
12	Έσοδο	243
13	Έξοδο	243
14	Τιμολόγιο	40
15	Απόδειξη	820
16	Πρόβατο	441+L
17	Γέννες προβάτων	815
Σύνολο		9948

Από τον πιο πάνω πίνακα φαίνεται ότι το συνολικό μέγεθος των δεδομένων για μία εγγραφή σε κάθε αρχείο είναι περίπου 9.95Kb.

3.2.9 Απαιτήσεις σε hardware

3.2.9.1 Αποθηκευτικές μονάδες

Το σύστημα που έχει αναπτυχθεί, υλοποιεί μια desktop εφαρμογή που διαχειρίζεται μια βάση δεδομένων. Για την αποθήκευση και την ανάκτηση των δεδομένων της βάσης, καθώς επίσης και για την αποθήκευση των ίδιων δεδομένων στο backup (LOG FILE), είναι απαραίτητη η χρήση ενός desktop PC με σκληρό δίσκο ελάχιστης χωρητικότητας 5GB. Είναι επίσης απαραίτητο για την εισαγωγή των αρχικών δεδομένων, την εγκατάσταση του προγράμματος που αναπτύσσουμε και τον Microsoft SQL Server 2008.

3.2.9.2 Συσκευές εισόδου δεδομένων

Οι συσκευές εισόδου του συστήματος θα είναι το ποντίκι και το πληκτρολόγιο, τα οποία θα χρησιμοποιούνται για την εισαγωγή των δεδομένων που αφορούν τα πρόβατα, τους ιδιοκτήτες και τα τιμολόγια, καθώς επίσης και για πλοήγηση στην επιφάνεια εργασίας του προγράμματος.

3.2.9.3 Συσκευές εξόδου δεδομένων

Οι συσκευές εξόδου του συστήματος θα είναι η οθόνη για την παρουσίαση των δεδομένων, ένας συνηθισμένος εκτυπωτής για την εκτύπωση των αναφορών και άλλων εγγράφων που αφορούν την κτηνοτροφική μονάδα.

3.2.9.4 Επιπλέον εξοπλισμός

Επιπλέον εξοπλισμός θα χρειαστεί αν υπάρχει ανάγκη για επιπλέον μνήμη, όπου θα χρειαστεί ακόμη ένας σκληρός δίσκος, πιθανόν εξωτερικός, για την φύλαξη των backup αρχείων της βάσης δεδομένων. Έτσι θα εξασφαλιστεί μεγαλύτερη ασφάλεια στο σύστημα αφού θα μπορεί ο χρήστης να ανακτήσει τα δεδομένα σε περίπτωση που υπάρξει κάποια βλάβη στον δίσκο.

3.3 Βήματα Αντικειμενοστραφούς Ανάλυσης

3.3.1 Εισαγωγή

Απαραίτητο είναι ένα ιδεατό μοντέλο των διεργασιών του συστήματος που θα λειτουργήσουν στο λογισμικό, για τη καλύτερη κατανόηση , προσδιορισμό και την έκφραση των αναγκών του λογισμικού. Τα μοντέλα που χρησιμοποιούνται για την αντικειμενοστραφή ανάλυση είναι οι ακόλουθες τεχνικές παραστάσεις:

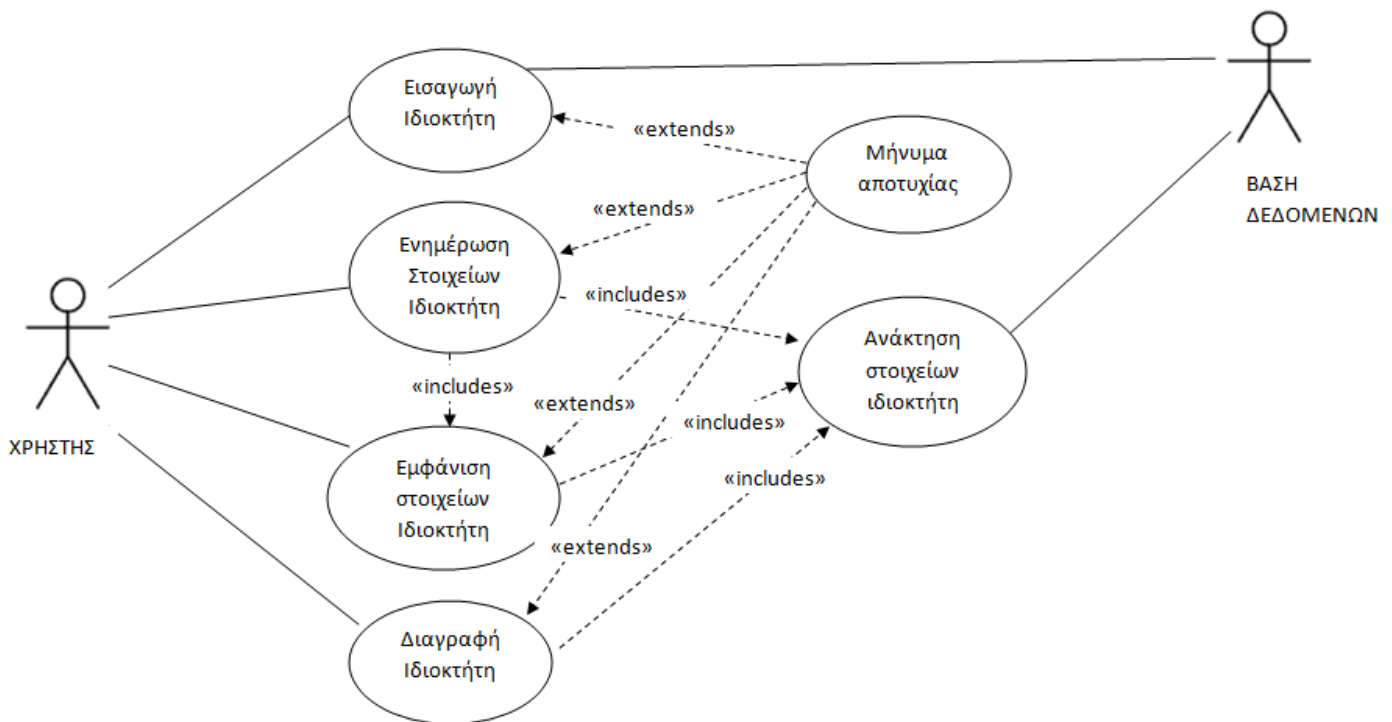
- Διαγράμματα Χρήσης / Use-Case Diagrams
- Διαγράμματα Κλάσεων / Analysis Class Diagrams

Η τεχνική που έχει χρησιμοποιηθεί για τη αναπαράσταση των ιδεατών μοντέλων είναι τα Use Case Diagrams.

3.3.1.1 Use Case Diagrams

Τα διαγράμματα χρήσης είναι ένα σύνολο σεναρίων τα οποία συνδέονται για τους σκοπούς που θέλει να εκπληρώσει ο χρήστης. Συνήθως περιλαμβάνει ένα πρωταρχικό σενάριο σαν μία ακολουθία από βήματα, και τις εναλλακτικές περιπτώσεις σαν παραλλαγές της κύριας ακολουθίας βημάτων.

Διαχείριση Ιδιοκτητών

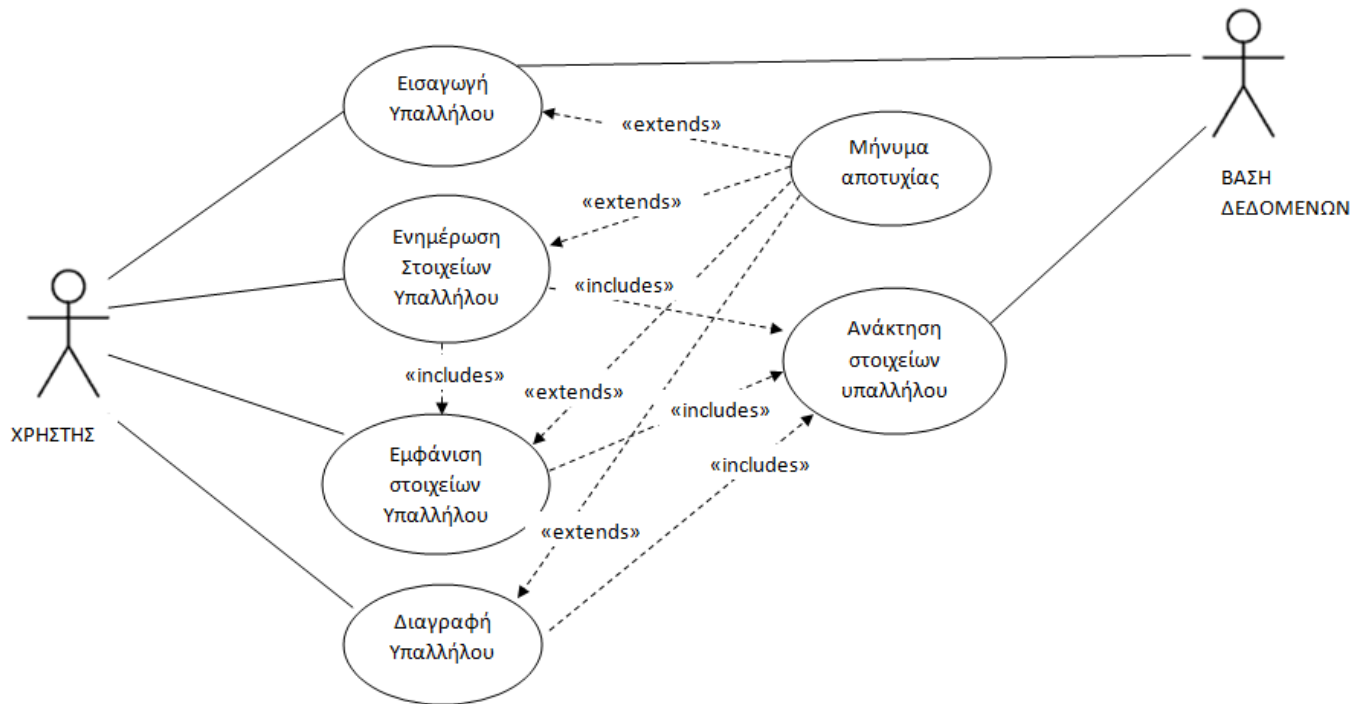


Διάγραμμα 3.17

Κατά την διαχείριση των ιδιοκτητών, ο χρήστης μπορεί να:

- Προσθέσει ένα νέο ιδιοκτήτη: σε περίπτωση σφάλματος, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία του νέου ιδιοκτήτη αποθηκεύονται στην βάση δεδομένων και ο χρήστης λαμβάνει μήνυμα επιτυχίας.
- Αναζητήσει τα στοιχεία κάποιου υπάρχων ιδιοκτήτη: αν δεν βρεθούν τα στοιχεία του ιδιοκτήτη, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και παρουσιάζονται στον χρήστη.
- Ενημερώσει τα στοιχεία κάποιου υπάρχων ιδιοκτήτη: αν δεν βρεθούν τα στοιχεία του ιδιοκτήτη, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και ενημερώνονται από τον χρήστη.
- Διαγράψει τα στοιχεία κάποιου υπάρχων ιδιοκτήτη: όταν τα στοιχεία ανακτηθούν από την βάση δεδομένων, ο ιδιοκτήτης διαγράφεται.

Διαχείριση Υπαλλήλων

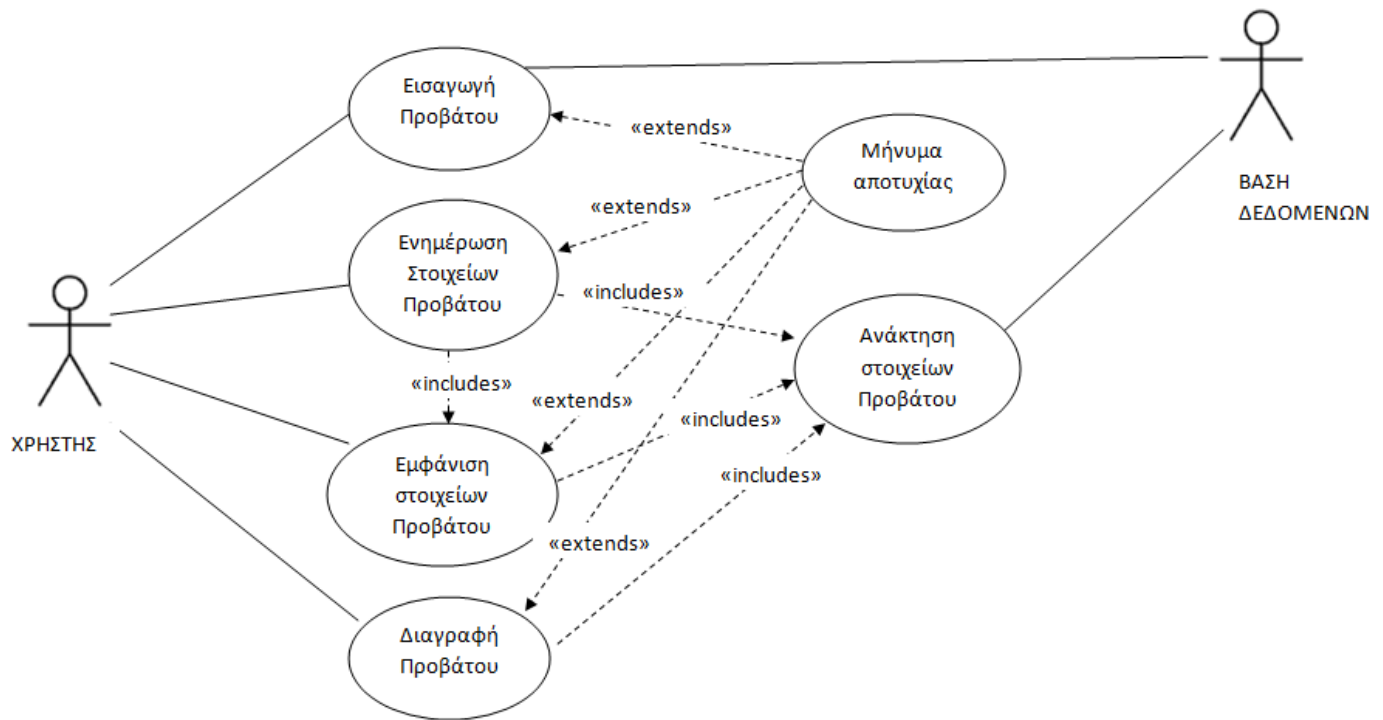


Διάγραμμα 3.18

Κατά την διαχείριση των υπαλλήλων, ο χρήστης μπορεί να:

- Προσθέσει ένα νέο υπάλληλο: σε περίπτωση σφάλματος, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία του νέου υπαλλήλου αποθηκεύονται στην βάση δεδομένων και ο χρήστης λαμβάνει μήνυμα επιτυχίας.
- Αναζητήσει τα στοιχεία κάποιου υπάρχων υπαλλήλου: αν δεν βρεθούν τα στοιχεία του υπαλλήλου, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και παρουσιάζονται στον χρήστη.
- Ενημερώσει τα στοιχεία κάποιου υπάρχων υπαλλήλου: αν δεν βρεθούν τα στοιχεία του υπαλλήλου, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και ενημερώνονται από τον χρήστη.
- Διαγράψει τα στοιχεία κάποιου υπάρχων υπαλλήλου: όταν τα στοιχεία ανακτηθούν από την βάση δεδομένων, ο υπάλληλος διαγράφεται.

Διαχείριση Προβάτων

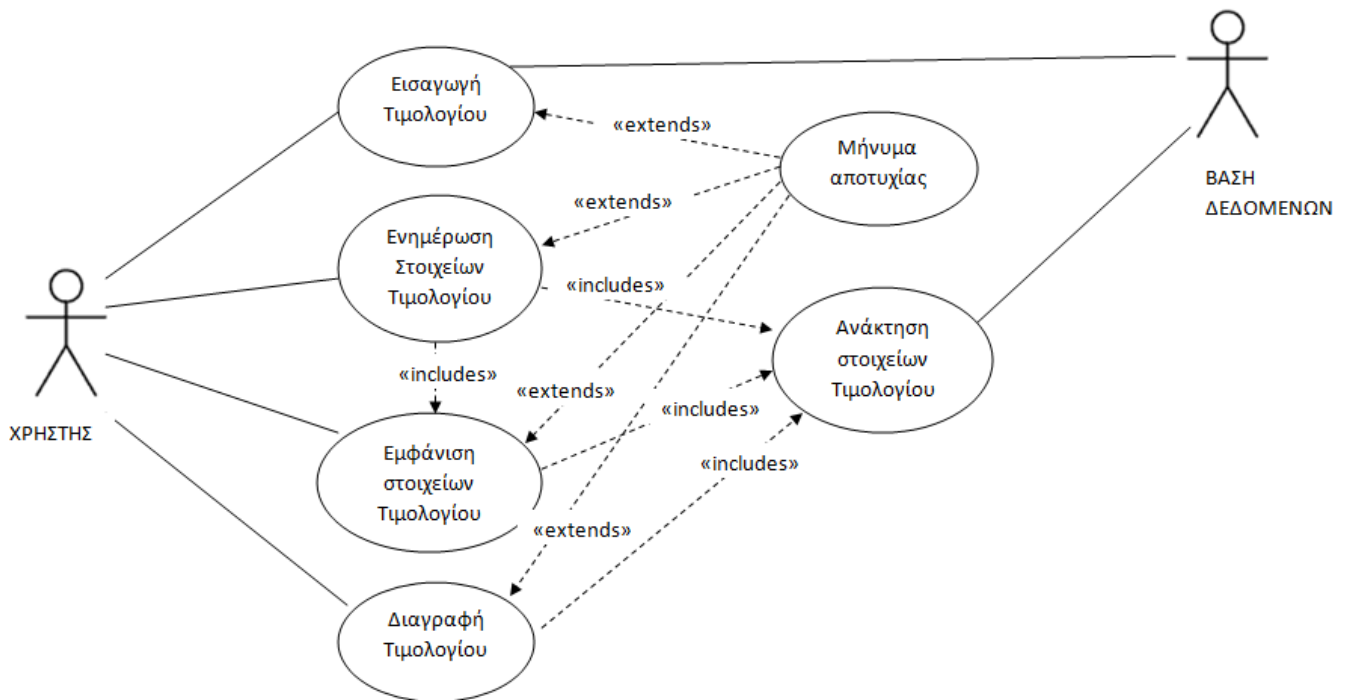


Διάγραμμα 3.19

Κατά την διαχείριση των προβάτων, ο χρήστης μπορεί να:

- Προσθέσει ένα νέο πρόβατο: σε περίπτωση σφάλματος, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία του νέου προβάτου αποθηκεύονται στην βάση δεδομένων και ο χρήστης λαμβάνει μήνυμα επιτυχίας.
- Αναζητήσει τα στοιχεία κάποιου υπάρχων προβάτου: αν δεν βρεθούν τα στοιχεία του προβάτου, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται σε μορφή λίστας και παρουσιάζονται στον χρήστη.
- Ενημερώσει τα στοιχεία κάποιου υπάρχων προβάτου: αν δεν βρεθούν τα στοιχεία του προβάτου, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και ενημερώνονται από τον χρήστη.
- Διαγράψει τα στοιχεία κάποιου υπάρχων προβάτου: όταν τα στοιχεία ανακτηθούν από την βάση δεδομένων, το πρόβατο διαγράφεται.

Διαχείριση Τιμολογίων

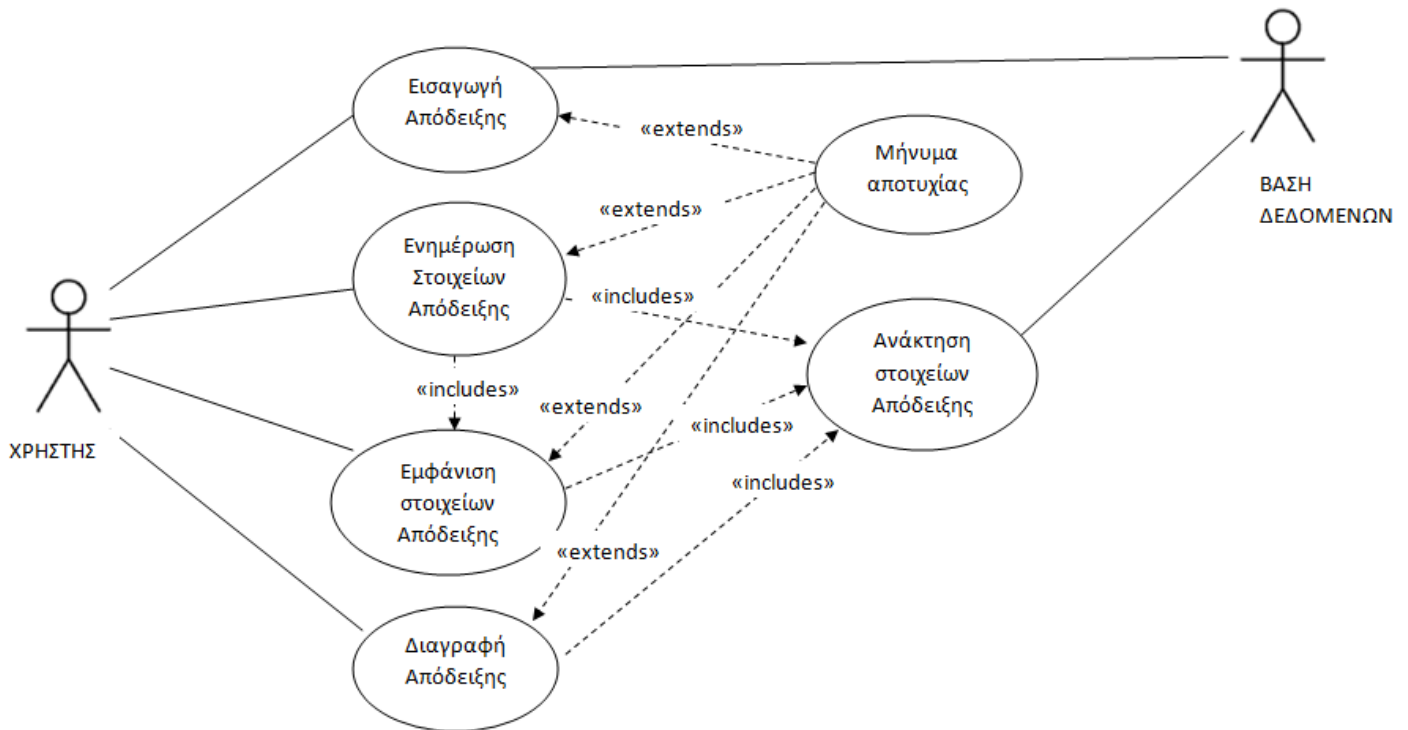


Διάγραμμα 3.20

Κατά την διαχείριση των τιμολογίων, ο χρήστης μπορεί να:

- Προσθέσει ένα νέο τιμολόγιο: σε περίπτωση σφάλματος, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία του νέου τιμολογίου αποθηκεύονται στην βάση δεδομένων και ο χρήστης λαμβάνει μήνυμα επιτυχίας.
- Αναζητήσει τα στοιχεία κάποιου υπάρχων τιμολογίου: αν δεν βρεθούν τα στοιχεία του τιμολογίου, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται σε μορφή λίστας και παρουσιάζονται στον χρήστη.
- Ενημερώσει τα στοιχεία κάποιου υπάρχων τιμολογίου: αν δεν βρεθούν τα στοιχεία του τιμολογίου, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και ενημερώνονται από τον χρήστη.
- Διαγράψει τα στοιχεία κάποιου υπάρχων τιμολογίου: όταν τα στοιχεία ανακτηθούν από την βάση δεδομένων, το τιμολόγιο διαγράφεται.

Διαχείριση Απόδειξεων



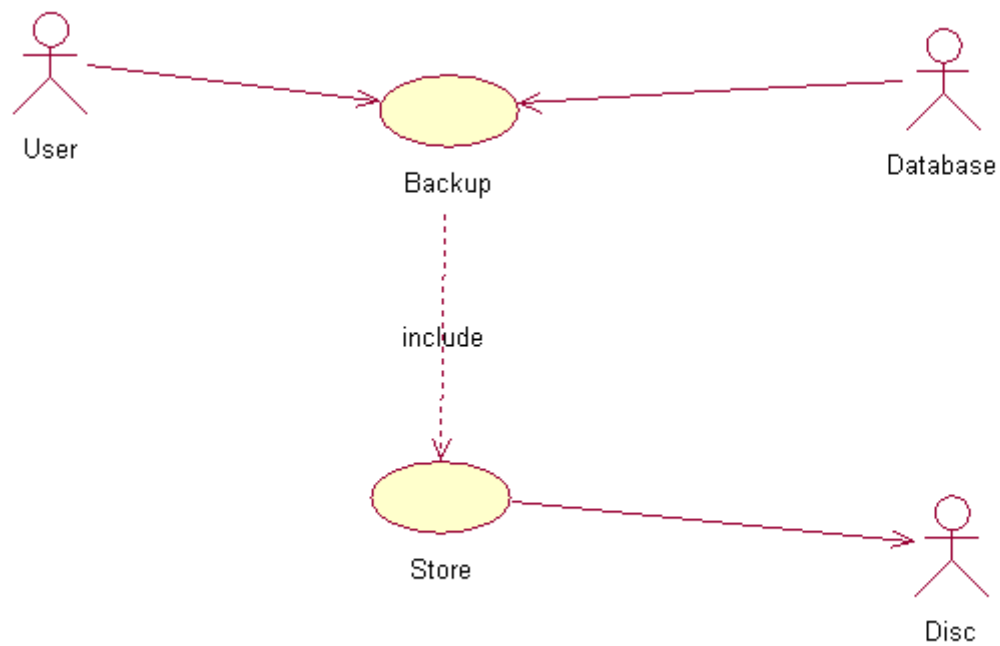
Διάγραμμα 3.21

Κατά την διαχείριση των αποδείξεων, ο χρήστης μπορεί να:

- Προσθέσει μία νέα απόδειξη: σε περίπτωση σφάλματος, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία της νέας απόδειξης αποθηκεύονται στην βάση δεδομένων και ο χρήστης λαμβάνει μήνυμα επιτυχίας.
- Αναζητήσει τα στοιχεία κάποιας υπάρχουσας απόδειξης: αν δεν βρεθούν τα στοιχεία της απόδειξης, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται σε μορφή λίστας και παρουσιάζονται στον χρήστη.
- Ενημερώσει τα στοιχεία κάποιας υπάρχουσας απόδειξης: αν δεν βρεθούν τα στοιχεία της απόδειξης, ο χρήστης λαμβάνει μήνυμα αποτυχίας. Διαφορετικά, τα στοιχεία ανακτώνται και ενημερώνονται από τον χρήστη.
- Διαγράψει τα στοιχεία κάποιας υπάρχουσας απόδειξης: όταν τα στοιχεία ανακτηθούν από την βάση δεδομένων, η απόδειξη διαγράφεται.

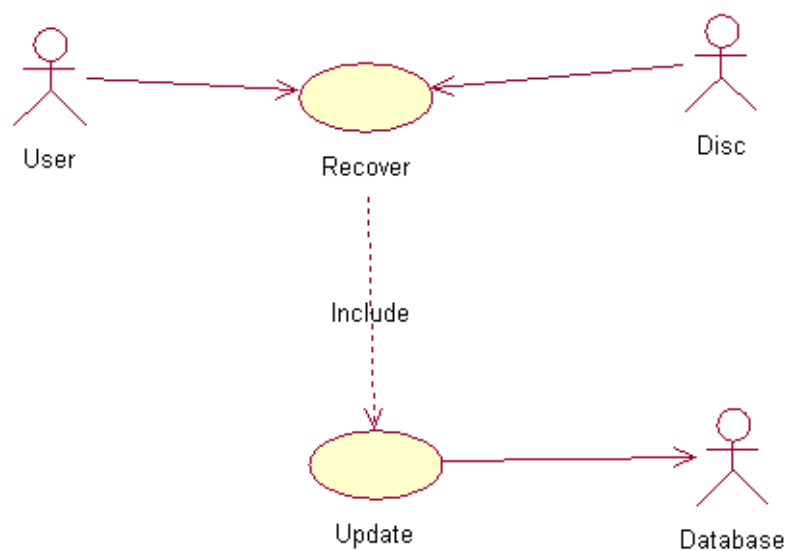
Log File_(Πρόσβαση μόνο από τον administrator)

- Αποθήκευση δεδομένων:



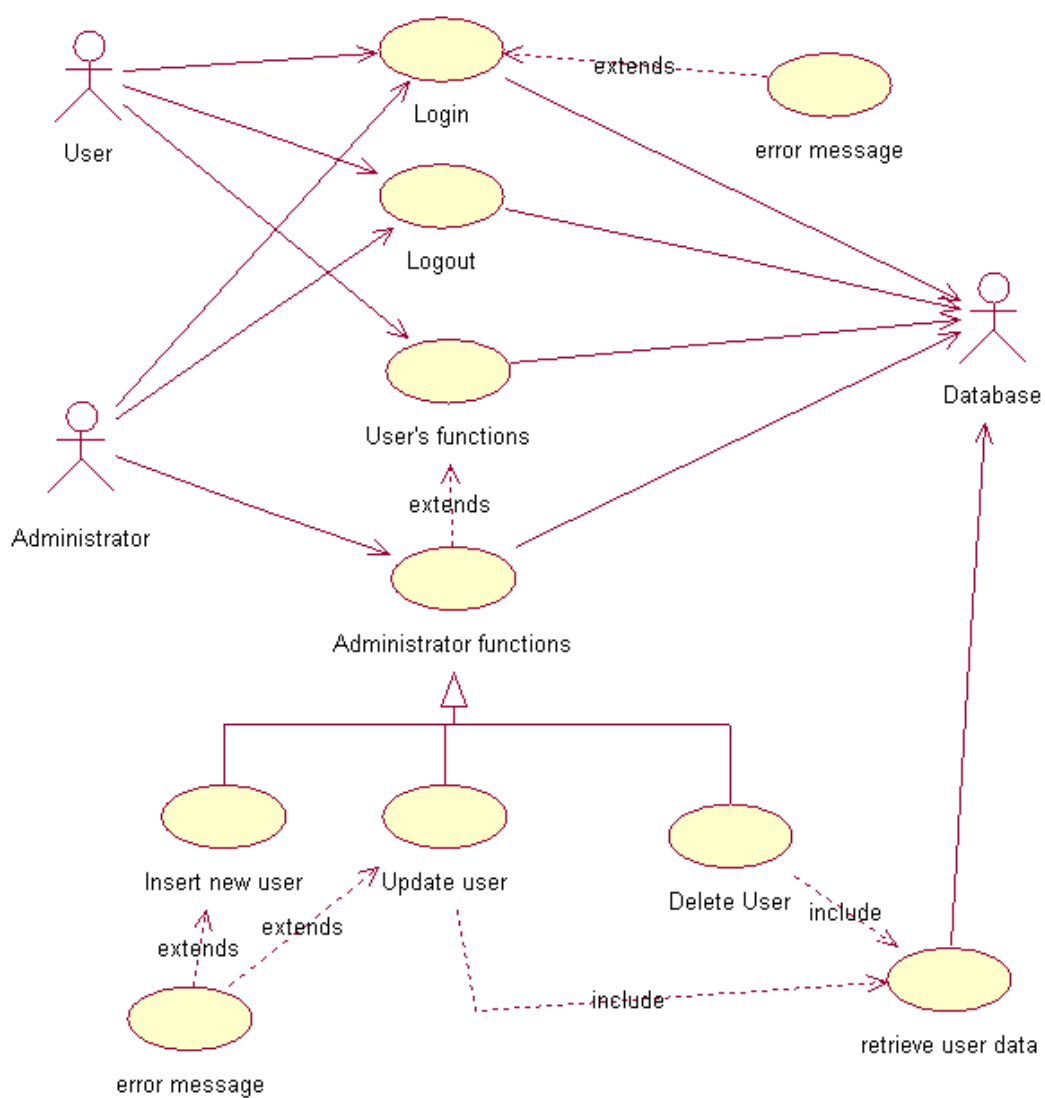
Διάγραμμα 3.22

- Επαναφορά δεδομένων:



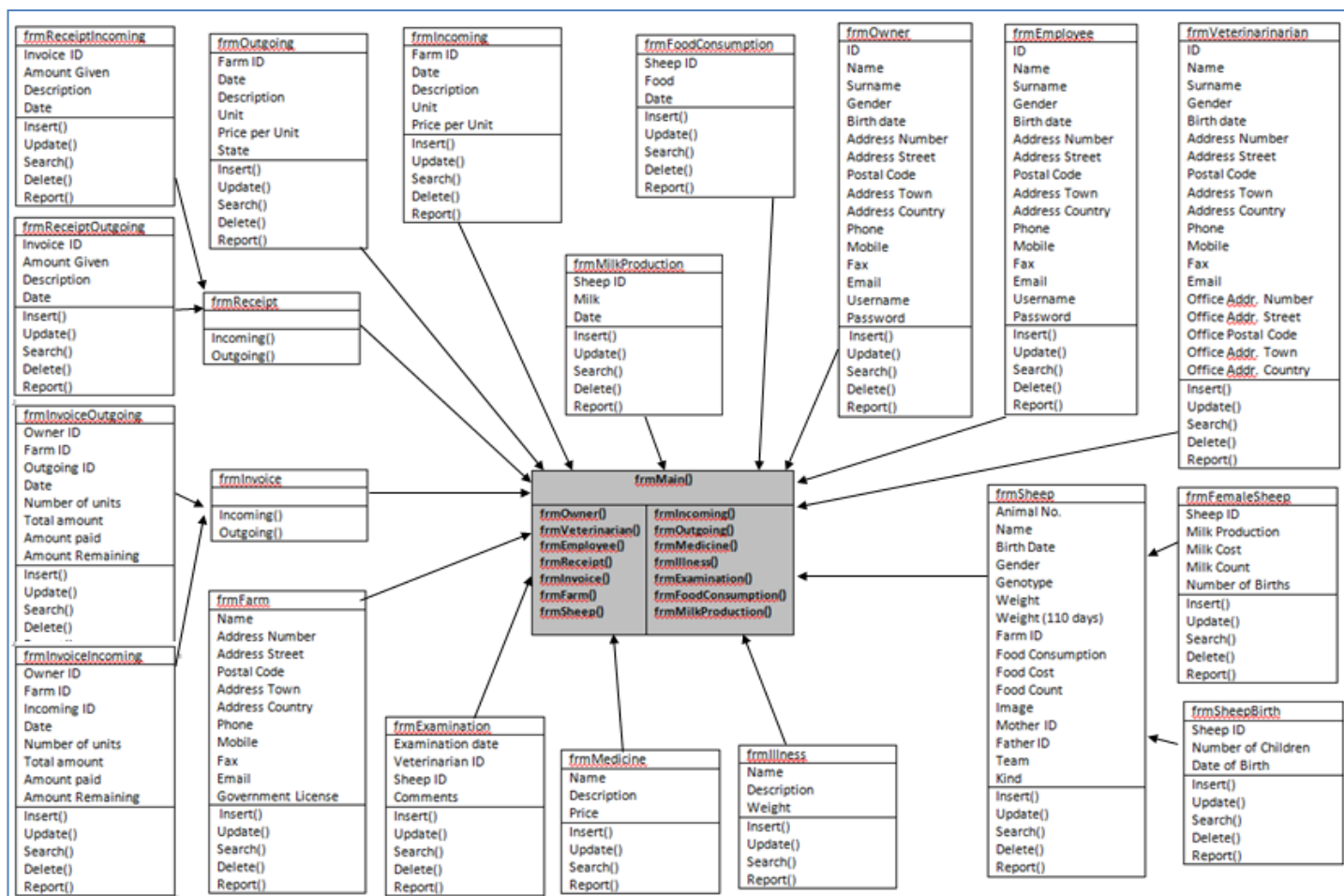
Διάγραμμα 3.23

Login



Διάγραμμα 3.24

3.3.1.2 Analysis Class Diagrams



Διάγραμμα 3.25

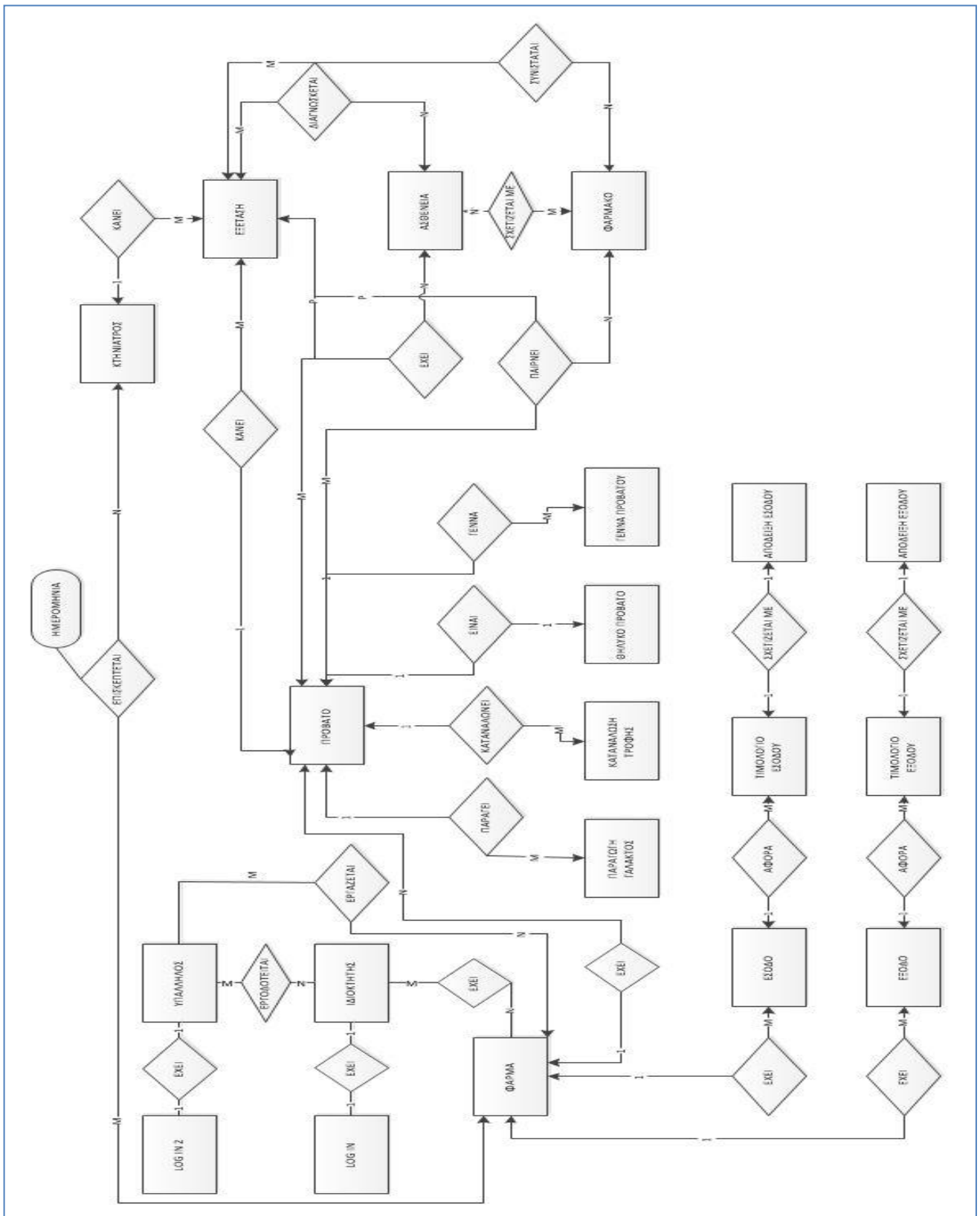
Η Main (κλάση με γκρίζο χρώμα) είναι η κλάση που ελέγχει όλες τις υπόλοιπες κλάσεις του συστήματος. Από εδώ μπορεί να γίνει κλήση στις βασικές κλάσεις του συστήματος (κλάσεις που είναι απευθείας συνδεδεμένες με την Main). Μέσω της κλάσης frmSheep μπορεί να γίνει εισαγωγή, ενημέρωση, αναζήτηση, εκτύπωση και διαγραφή ενός προβάτου. Παρόμοια, μέσω των κλάσεων frmOwner, frmEmployee, frmVeterinarian, frmExamination, frmIncoming, frmOutgoing μπορούν να εισαχθούν, αναζητηθούν, ενημερωθούν, εκτυπωθούν και διαγραφούν ιδιοκτήτες, υπαλλήλοι, κτηνιάτροι, εξετάσεις, έσοδα και έξοδα, αντίστοιχα.

Επιπλέον, ο χρήστης, μέσω των κλάσεων frmMilkProduction(), frmFoodConsumption(), μπορεί να εισάγει την ημερήσια παραγωγή γάλακτος για κάθε πρόβατο ξεχωριστά και την ημερήσια κατανάλωση τροφής για ομάδες προβάτων. Μπορεί επίσης, να δει τις αναφορές που παράγονται για κάθε πρόβατο ξεχωριστά, αλλά

και συνολικά έτσι ώστε να δει την πρόοδο της γαλακτοπαραγωγής για κάθε πρόβατο και να παρατηρήσει τυχόν προβλήματα όσο αφορά την διατροφή του.

Μέσω της κλάσης `frmSheep()`, ο χρήστης μπορεί να εισάγει, ενημερώσει, αναζητήσει και διαγράψει την συνολική παραγωγή σε γάλα μέσω της κλάσης `frmFemaleSheep`, αλλά και στοιχεία όσο αφορά τις γέννες κάποιων προβάτων μέσω της κλάσης `frmSheepBirth`.

3.4 Διαγράμματα Οντοτήτων



Διάγραμμα 3.26

Στο πιο πάνω διάγραμμα οντοτήτων-συσχετίσεων φαίνονται όλες οι σχέσεις που διέπουν τους πίνακες που απαρτίζουν την βάση δεδομένων του συστήματος.

Στην βάση δεδομένων γίνεται ένας διαχωρισμός ανάμεσα στα έσοδα και τα έξοδα που έχει η κάθε φάρμα, και τότε ανάλογα με τα έσοδα και έξοδα της κάθε φάρμας ο χρήστης μπορεί να δημιουργήσει τα αντίστοιχα τιμολόγια και στην συνέχεια τις ανάλογες αποδείξεις. Ο χρήστης μπορεί να ενημερώσει και να διαγράψει τιμολόγια και αποδείξεις τα οποία δεν επιθυμεί. Με την εισαγωγή κάποιας απόδειξης, ενημερώνεται απευθείας το τιμολόγιο το οποίο την αφορά, δηλαδή μειώνεται το υπολειπόμενο ποσό και αυξάνεται αντίστοιχα το πληρωμένο ποσό του σχετικού τιμολογίου.

Επιπλέον, στην βάση υπάρχουν πίνακες για τους ιδιοκτήτες και τους υπαλλήλους της φάρμας και στον κάθε ένα από αυτούς ανατίθεται ένα συνθηματικό όνομα και κωδικός με τα οποία θα τους επιτρέπεται η είσοδος στο σύστημα της κτηνοτροφικής μονάδας. Το συνθηματικό όνομα και ο κωδικός και για τον κάθε ιδιοκτήτη βρίσκεται στον πίνακα LOG IN της βάσης δεδομένων, ενώ το αντίστοιχο συνθηματικό και κωδικός των υπαλλήλων βρίσκονται στον πίνακα LOG IN 2 της βάσης.

Ο χρήστης έχει την δυνατότητα να εισάγει κάποιες ασθένειες των ζώων, καθώς και φάρμακα τα οποία χορηγούνται στα ζώα κατόπιν συνταγών των κτηνιάτρων, στους πίνακες ILLNESS και MEDICINE, αντίστοιχα. Με βάση τις πληροφορίες των 2 αυτών πινάκων, ο χρήστης τις αξιοποιεί ώστε να εισάγει μία εξέταση του ζώου στο σύστημα με ολοκληρωμένα στοιχεία. Επιπρόσθετα, γνωρίζοντας τις ασθένειες και τα φάρμακα τα οποία χορηγήθηκαν στο κάθε πρόβατο, ο χρήστης θα βοηθηθεί στην διαδικασία της επιλογής των ζώων για την επόμενη περίοδο, δηλαδή θα τον βοηθήσουν στην απόφαση του για το ποια ζώα θα απορριφθούν από την φάρμα.

Μία από τις λειτουργίες του συστήματος που προσφέρονται στον χρήστη, είναι η διαδικασία της καταχώρησης της ημερήσιας παραγωγής γάλακτος, καθώς και της ημερήσιας κατανάλωσης τροφής, η οποία είναι δυνατή μέσω γραφικών. Οι πληροφορίες αυτές αποθηκεύονται στους πίνακες MILK_PRODUCTION και FOOD_CONSUMPTION, αντίστοιχα. Στο τέλος κάθε μήνα, τα στοιχεία των πινάκων αυτών γίνονται export σε Excel αρχεία ώστε να μπορεί ο χρήστης να έχει αποθηκευμένα τα στοιχεία αφού είναι πάρα πολλά για να βρίσκονται αποθηκευμένα

στον πίνακα της βάσης δεδομένων. Ακολούθως, μετά την διαδικασία export σε excel, τα στοιχεία των πινάκων αυτών διαγράφονται ώστε να αποθηκεύονται κάθε φορά η παραγωγή γάλακτος και κατανάλωση τροφής των προβάτων για κάθε μήνα.

Επίσης, οποιαδήποτε δεδομένα τα οποία είναι βοηθητικά για λειτουργίες και υπολογισμούς της κτηνοτροφικής μονάδας, αποθηκεύονται στην βάση δεδομένων, έτσι ώστε να αποφεύγεται η επανάληψη και η συνεχής πληκτρολόγηση τους από τον χρήστη του συστήματος.

Τα χαρακτηριστικά των οντοτήτων φαίνονται στην παράγραφο 3.2.5- *Προσδιορισμός αποθήκευσης δεδομένων*.

Κεφάλαιο 4: Ανάλυση Σχεδίασης

Κεφάλαιο 4: Ανάλυση Σχεδίασης.....	135
4.1 Εισαγωγή.....	135
4.1.1 Σκοπός	135
4.1.2 Επισκόπηση Φάσης Σχεδίασης.....	136
4.2 Αρχιτεκτονική Σχεδίαση.....	137
4.2.1. Ανάλυση Ροών Δεδομένων.....	137
4.3 Αναλυτική Σχεδίαση.....	160
4.3.1 Οντότητες με κοινές λειτουργίες	160
4.4 Graphical User Interface	183

Κεφάλαιο 4: Ανάλυση Σχεδίασης

4.1 Εισαγωγή

4.1.1 Σκοπός

Το έγγραφο της σχεδίασης έχει σκοπό να μειώσει την συνολική πολυπλοκότητα του συστήματος και κατά συνέπεια το κόστος υλοποίησης και το ενδεχόμενο λαθών μέσα από την τμηματικότητα του σχεδίου του συστήματος. Μια καλή μέθοδος σχεδιασμού πρέπει να επιτρέπει:

1. Την αποσύνθεση σε τμήματα (modular decomposability)
2. Την σύνθεση τμημάτων (modular compos ability)
3. Την χωριστή κατανόηση τμημάτων (modular understandability)
4. Τη συνέχεια των τμημάτων (modular continuity) σε αλλαγές: θα πρέπει να επηρεάζονται λίγα τμήματα
5. Την προστασία των τμημάτων (modular protection) : λάθος σε ένα τμήμα δε θα πρέπει να δημιουργεί προβλήματα σε άλλα

Ο καλός σχεδιασμός πρέπει να φέρνει κοντά τμήματα που εμφανίζουν υψηλή συνεκτικότητα. Διακρίνονται τα παρακάτω αυξανόμενα επίπεδα συνεκτικότητας:

1. Συμπτωματική συνεκτικότητα (coincidental cohesion)
2. Λογική συνεκτικότητα (logical cohesion)
3. Χρονική συνεκτικότητα (temporal cohesion)
4. Διαδικαστική συνεκτικότητα (procedural cohesion)
5. Επικοινωνιακή συνεκτικότητα (communicational cohesion)
6. Ακολουθιακή συνεκτικότητα (sequential cohesion)
7. Λειτουργική συνεκτικότητα (functional cohesion)
8. Σύζευξη (coupling)

Ο καλός σχεδιασμός πρέπει κατά το δυνατόν να αποφεύγει τη σύζευξη μεταξύ των τμημάτων. Διακρίνονται τα παρακάτω αυξανόμενα επίπεδα σύζευξης:

1. Σύζευξη δεδομένων (data coupling)
2. Σύζευξη δομής δεδομένων (stamp coupling) (ή αντιγράφου)

3. Σύζευξη ελέγχου (control coupling)
4. Σύζευξη κοινών δεδομένων (common coupling) (ή από κοινού σύνδεση)
5. Σύζευξη εξωτερικών δεδομένων (external coupling)
6. Σύζευξη περιεχομένων (content coupling)

Επιπρόσθετα, στο έγγραφο της σχεδίασης θα μελετηθεί η αρχιτεκτονική και η αναλυτική σχεδίαση των διεργασιών του συστήματος. Το έγγραφο αυτό προορίζεται για τους προγραμματιστές του συστήματος. Επομένως, η αρχιτεκτονική και η αναλυτική σχεδίαση των διεργασιών θα είναι λεπτομερής και με σαφήνεια, έτσι ώστε να μπορούν να χρησιμοποιηθούν σαν αναφορά στη φάση της Υλοποίησης, δηλαδή οι προγραμματιστές να δημιουργήσουν ορθά την κάθε διεργασία και να διευκολυνθούν στην όλη υλοποίηση.

Στην αρχιτεκτονική σχεδίαση καθορίζονται τα τμήματα και οι λειτουργικότητες του συστήματος από τα οποία απαρτίζεται το κάθε τμήμα, με βάση τις προδιαγραφές που καθορίστηκαν στην φάση των Προδιαγραφών. Στην αναλυτική σχεδίαση, γίνεται λεπτομερής σχεδίαση του κάθε τμήματος και διεργασίας με λεπτομέρεια και σαφήνεια, για να αποφευχθούν τα λάθη στην φάση της υλοποίησης τα οποία σίγουρα θα καθυστερούσαν την φάση αυτή.

4.1.2 Επισκόπηση Φάσης Σχεδίασης

- Το δεύτερο βήμα, περιλαμβάνει την αρχιτεκτονική σχεδίαση. Το τμήμα αυτό ασχολείται με διαγράμματα ροής δεδομένων και διαγράμματα ανάλυσης ροής δεδομένων.
- Το τρίτο βήμα περιλαμβάνει την αναλυτική σχεδίαση η οποία περιλαμβάνει την ανάλυση των οντοτήτων που έχουν εντοπιστεί στα προηγούμενα βήματα.
- Τέλος θα ασχοληθούμε με το Grafical User Interface, δηλαδή θα παρουσιάσουμε την διεπαφή μεταξύ του χρήστη και του λογισμικού.

4.2 Αρχιτεκτονική Σχεδίαση

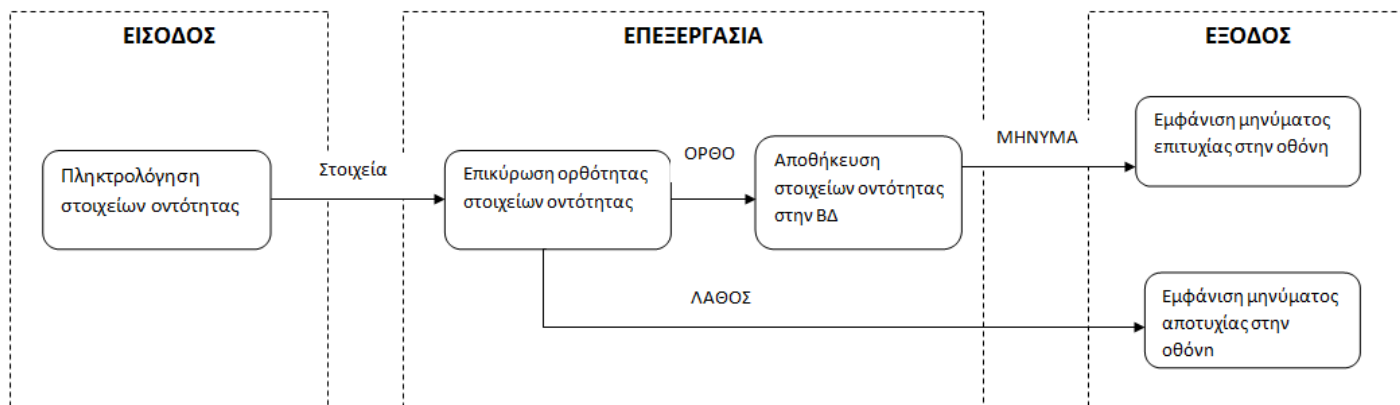
Κατά την αρχιτεκτονική σχεδίαση, γίνεται προσδιορισμός των μονάδων που αποτελούν το λογισμικό και πώς σχετίζονται μεταξύ τους. Επίσης, στην αρχιτεκτονική σχεδίαση, η εργασία αυτή γίνεται λαμβάνοντας ως είσοδο τα διαγράμματα ροής δεδομένων. Σε αυτά εφαρμόζεται ένα σύνολο κανόνων και καθορίζονται οι μονάδες λογισμικού που είναι αναγκαίες για την υλοποίηση των μετασχηματισμών και των ροών δεδομένων.

4.2.1. Ανάλυση Ροών Δεδομένων

4.2.1.1. Διαγράμματα ροής δεδομένων

4.2.1.1.1. Οντότητες με κοινές λειτουργίες:

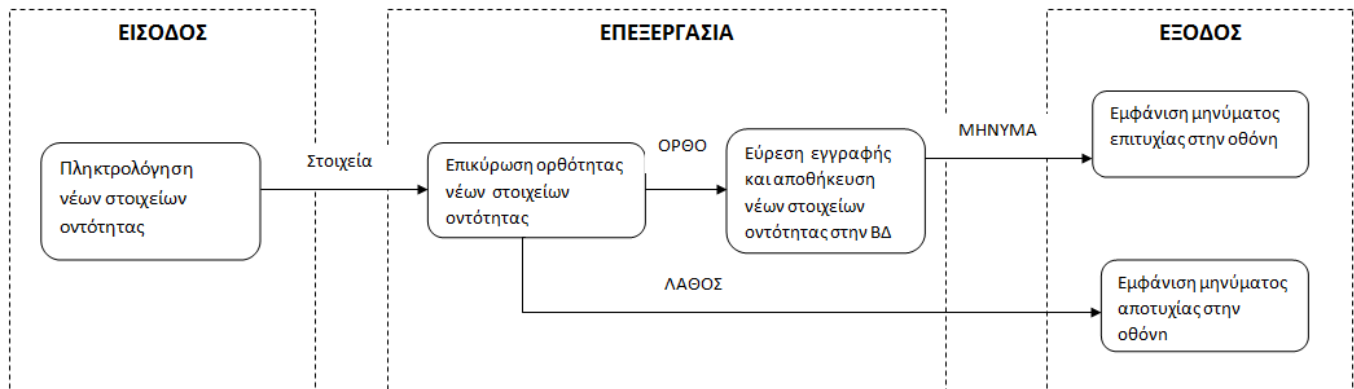
Εισαγωγή οντότητας



Διάγραμμα 4.1

Η λειτουργία της εισαγωγής οντότητας περιλαμβάνει εισαγωγή και αποθήκευση των στοιχείων της οντότητας τα οποία συμπληρώνονται από τον χρήστη μέσω κάποιας φόρμας. Στη συνέχεια γίνεται έλεγχος αν εισάχθηκαν τα ορθά δεδομένα, δηλαδή αν έχουν τον σωστό τύπο δεδομένων. Αν τα στοιχεία είναι σωστά και επικυρώνονται τότε αποθηκεύονται στη βάση και εμφανίζεται μήνυμα ορθότητας, αλλιώς εμφανίζεται μήνυμα λάθους.

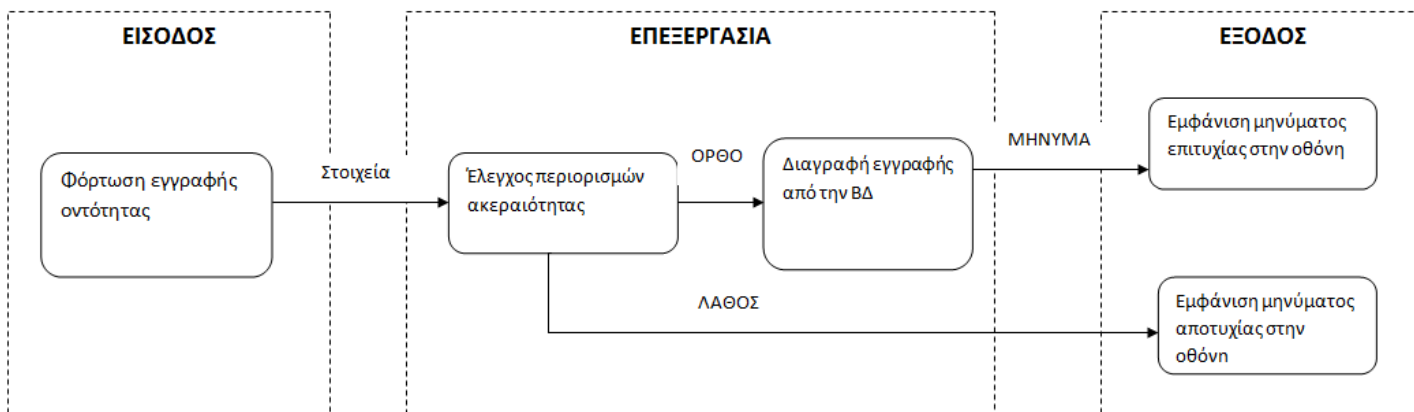
Τροποποίηση Οντότητας



Διάγραμμα 4.2

Η λειτουργία της τροποποίησης, η οποία γίνεται μετά από αναζήτηση, είναι υπεύθυνη για την ενημέρωση των πεδίων υπάρχουσών οντοτήτων της βάσης δεδομένων. Αφού γίνει η εισαγωγή των τροποποιημένων στοιχείων από τον χρήστη, στη συνέχεια ελέγχονται αν τα στοιχεία αυτά είναι σωστά. Αν είναι σωστά αποθηκεύονται στη βάση και εμφανίζεται μήνυμα ορθότητας. Αν όχι εμφανίζεται μήνυμα λάθους.

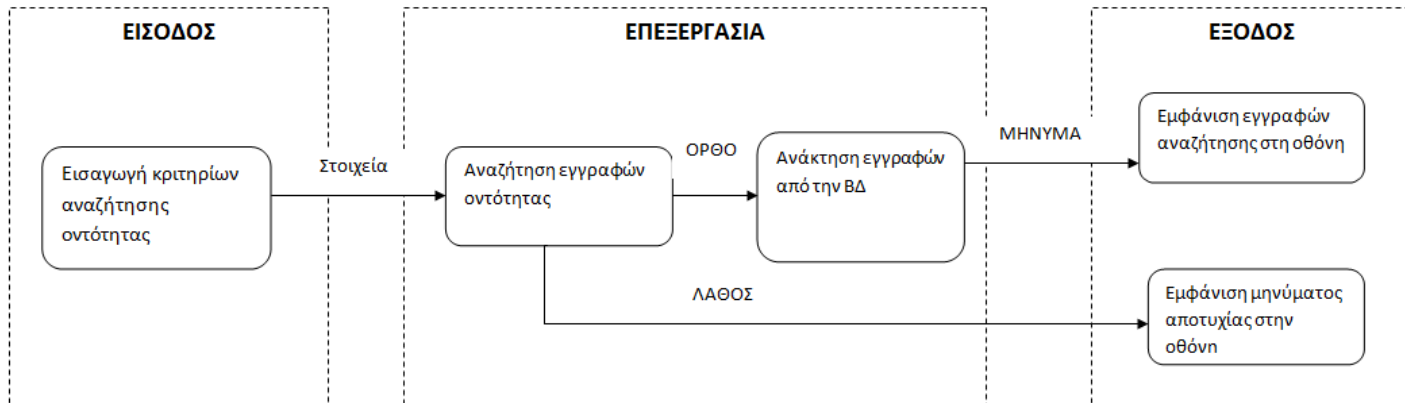
Διαγραφή Οντότητας



Διάγραμμα 4.3

Η λειτουργία της διαγραφής είναι υπεύθυνη για την ενημέρωση των πεδίων υπάρχουσών εγγραφών στην βάση δεδομένων. Μετά από επαναφορά της εγγραφής ελέγχουμε για τους περιορισμούς ακεραιότητας. Αν είναι ορθά διαγράφεται από τη ΒΔ αλλιώς εμφανίζεται μήνυμα λάθους προς τον χρήστη.

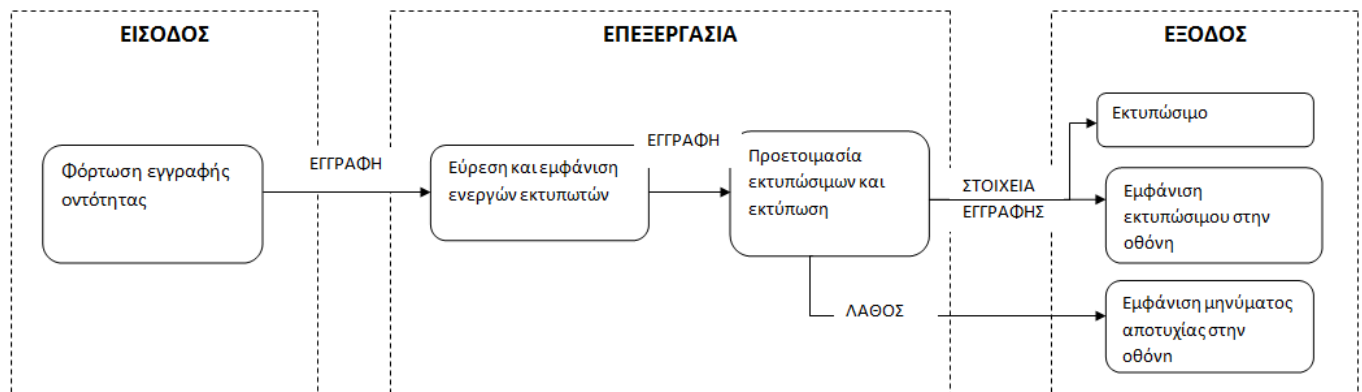
Αναζήτηση Οντότητας



Διάγραμμα 4.4

Η λειτουργία της αναζήτησης είναι υπεύθυνη να αναζητήσει εγγραφές από την βάση δεδομένων οι οποίες ικανοποιούν τα κριτήρια που έθεσε ο χρήστης. Σε περίπτωση που υπάρχει λάθος στα κριτήρια που τέθηκαν κατά την αναζήτηση ή δεν έχουν βρεθεί εγγραφές που ικανοποιούν τα κριτήρια, τότε εμφανίζεται μήνυμα λάθους στην οθόνη.

Εκτύπωση αναφοράς οντότητας

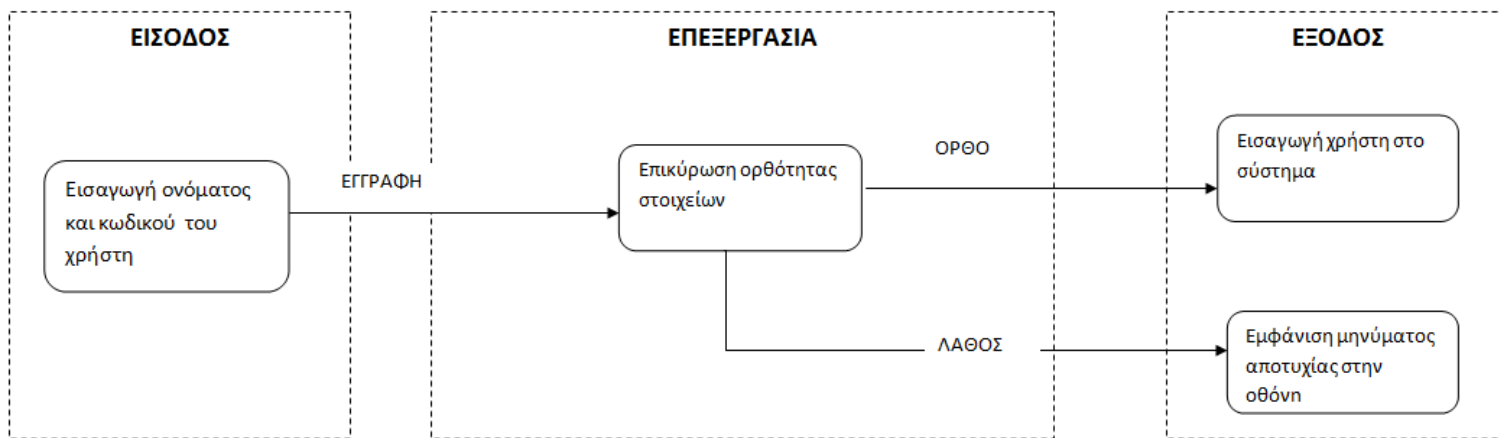


Διάγραμμα 4.5

Η λειτουργία της εκτύπωσης ανακτά μία εγγραφή από την βάση δεδομένων και αφού βρει τους ενεργούς εκτυπωτές και ομαδοποιεί κατάλληλα τα πεδία της εγγραφής με ένα προκαθορισμένο τρόπο, τα εμφανίζει στη οθόνη σε τυποποιημένη μορφή η οποία είναι δυνατό να τυπωθεί. Σε περίπτωση αποτυχίας της διαδικασίας αυτής, εμφανίζεται μήνυμα λάθους.

4.2.1.1.2 Οντότητες με επιπλέον ή διαφορετικές λειτουργίες

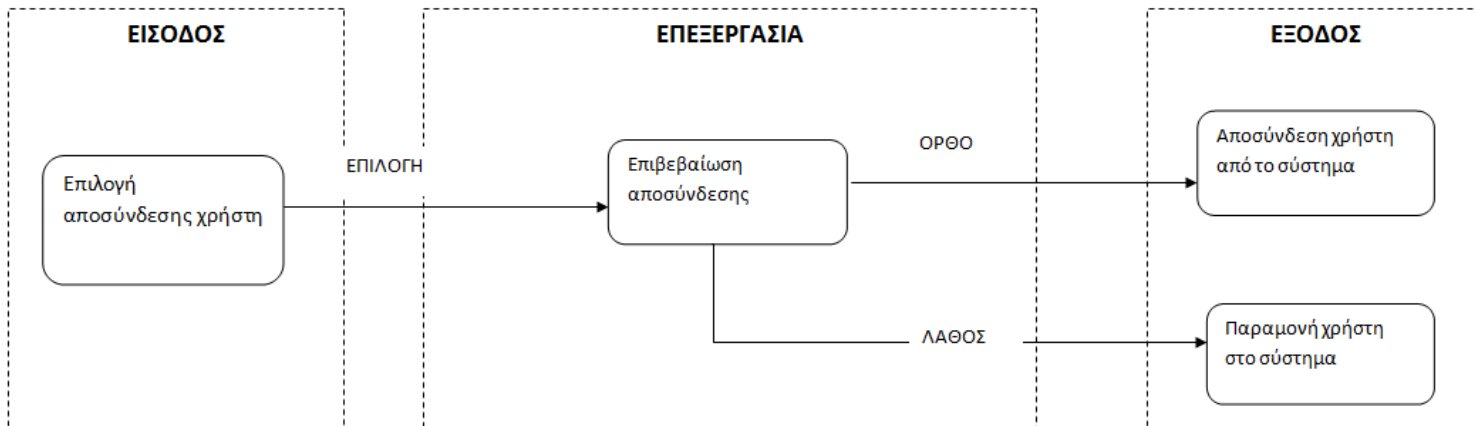
Εισαγωγή Χρήστη στο Σύστημα (LOGIN)



Διάγραμμα 4.6

Το module «εισαγωγή χρήστη στο σύστημα» είναι υπεύθυνο για να εισάγει ορθά το χρήστη στο σύστημα(διαδικασία login). Αφού ο χρήστης εισάγει το όνομα και τον κωδικό του, το module ελέγχει για την ορθότητα τους(κάνοντας και κρυπτογράφηση του κωδικού του)και εφόσον γίνει επικύρωση τότε ο χρήστης εισάγεται στο σύστημα. Στην αντίθετη περίπτωση εμφανίζεται ένα μήνυμα αποτυχημένης εισαγωγής.

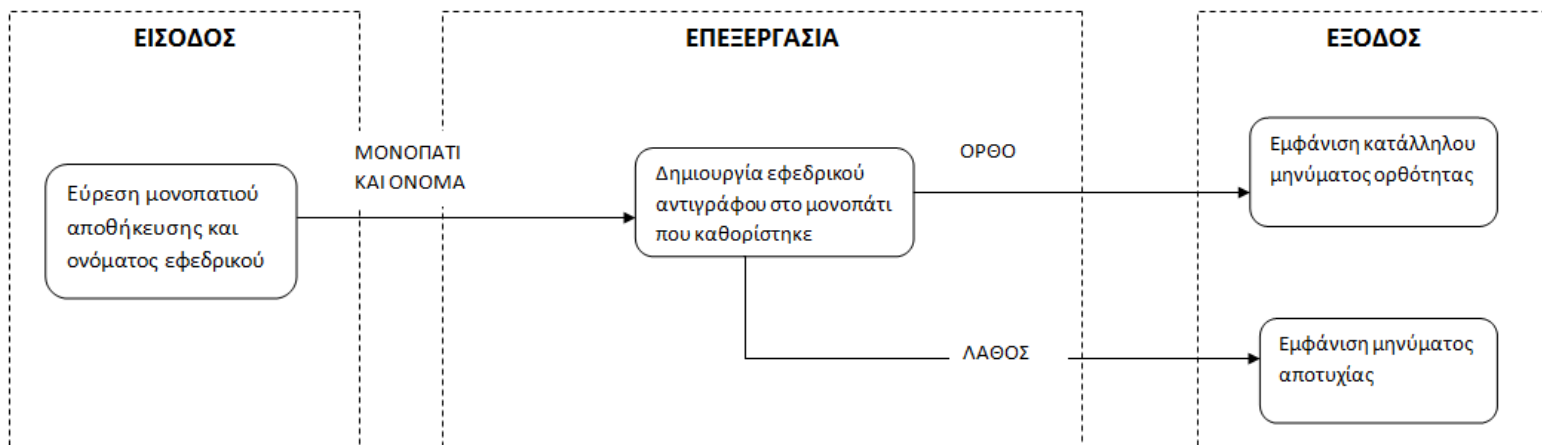
Αποσύνδεση Χρήστη από το Σύστημα (LOGOUT)



Διάγραμμα 4.7

Το module «αποσύνδεση από το σύστημα» είναι υπεύθυνο για αποσυνδέει το χρήστη από το σύστημα(διαδικασία logout). Αφού ο χρήστης έχει επιλέξει να αποσυνδεθεί από το σύστημα το module επιβεβαιώνει την αποσύνδεση κι έπειτα επιτυγχάνεται η αποσύνδεση. Στην αντίθετη περίπτωση που δεν γίνει επιβεβαίωση αποσύνδεσης ο χρήστης παραμένει συνδεδεμένος στο σύστημα και επιστρέφει στην κεντρική φόρμα του προγράμματος.

Δημιουργία Εφεδρικού Αντιγράφου

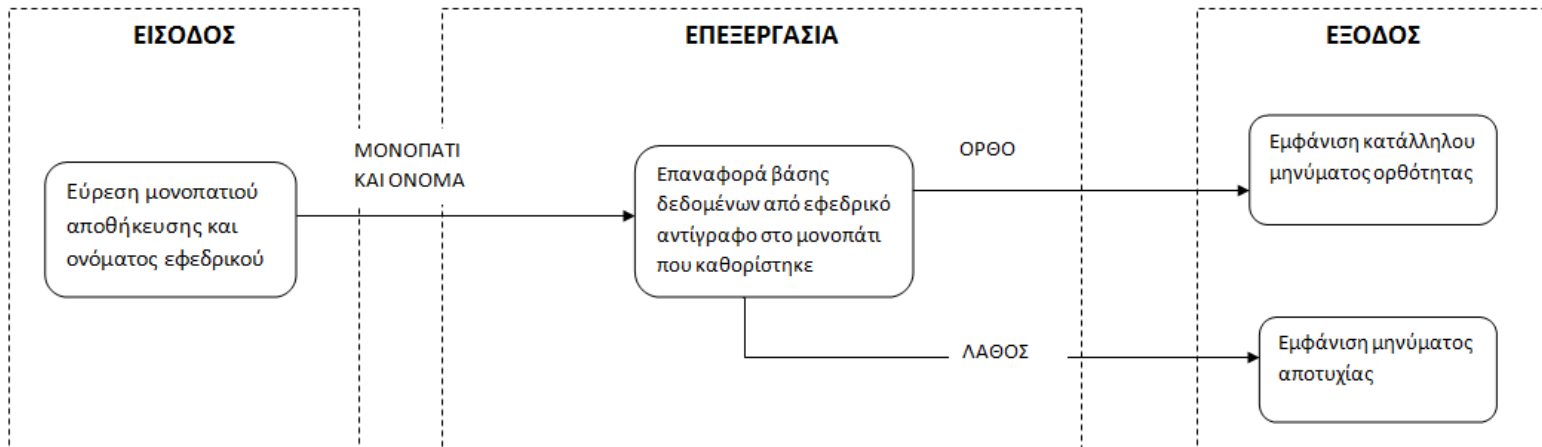


Διάγραμμα 4.8

Το module «δημιουργία εφεδρικού αντιγράφου» είναι υπεύθυνο για να δημιουργήσει ένα εφεδρικό αντίγραφο του συστήματος (log file) το οποίο θα χρησιμοποιηθεί σε περιπτώσεις απρόοπτων συμβάντων που θα προκαλέσουν απώλειες δεδομένων κτλ. Όταν ο χρήστης επιλέξει να δημιουργήσει εφεδρικό αντίγραφο

καθορίζει το μονοπάτι όπου θα αποθηκευτεί το εφεδρικό αυτό αντίγραφο του συστήματος όπως επίσης και το όνομα που θα δώσει στο συγκεκριμένο αρχείου. Έπειτα το module επιβεβαιώνει τη δημιουργία του στο συγκεκριμένου μονοπατιού στο δίσκο μέσω κάποιου μηνύματος στην οθόνη. Σε περίπτωση αποτυχίας ο χρήστης θα ειδοποιηθεί και πάλι μέσω κάποιου μηνύματος στην οθόνη.

Επαναφορά Βάσης Δεδομένων



Διάγραμμα 4.9

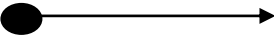
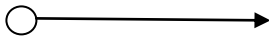
Το module «επαναφορά βάσης δεδομένων» είναι υπεύθυνο για να επαναφέρει τη βάση δεδομένων του συστήματος σε μια προηγούμενη κατάσταση. Όταν ο χρήστης επιλέξει να επαναφέρει τη βάση δεδομένων του συστήματος επιλέγει το μονοπάτι που βρίσκεται το εφεδρικό αντίγραφο όπως επίσης και το όνομα του αντιγράφου (που έχει δώσει στο προηγούμενο module). Έπειτα το module επιβεβαιώνει την επαναφορά της βάσης του συστήματος από το κατάλληλο μονοπάτι στο δίσκο και εμφανίζεται ένα μήνυμα ορθότητας στην οθόνη. Σε αντίθετη περίπτωση εμφανίζεται και πάλι ένα μήνυμα αποτυχίας στην οθόνη.

4.2.1.2. Διαγράμματα Ανάλυσης Ροών Δεδομένων

Το διάγραμμα ανάλυσης ροών δεδομένων αποτελεί ένα ιεραρχικό διάγραμμα που απεικονίζει την αρχιτεκτονική δομή ενός προγράμματος, παρουσιάζοντας τα τμήματα και τις μεταξύ τους σχέσεις. Βασικό του συστατικό είναι το τμήμα (module). Τα τμήματα συνδέονται μεταξύ τους με βέλη που συμβολίζουν την μεταφορά ελέγχου.

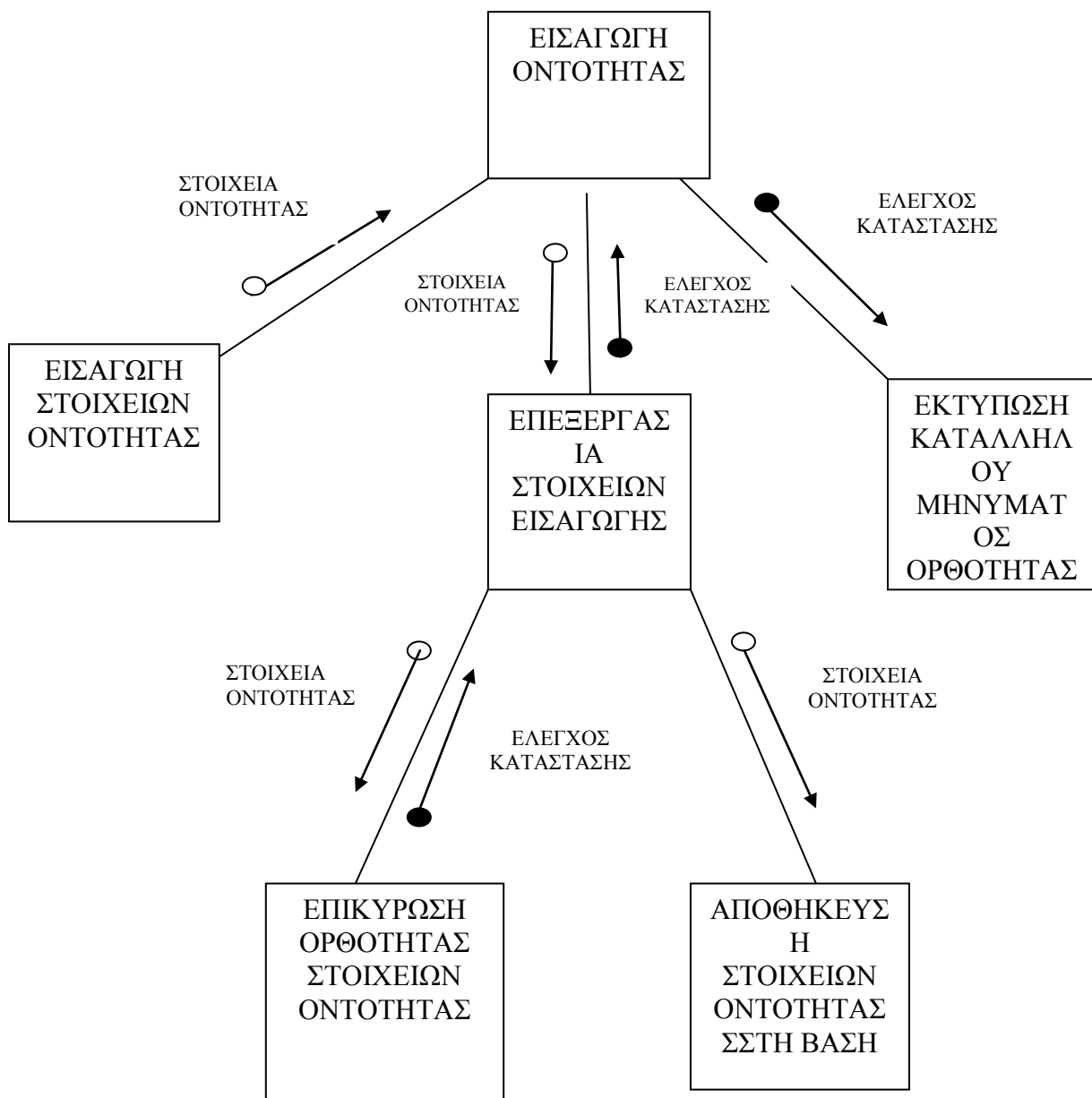
Πάνω στο βέλος μπορεί να τοποθετηθεί ένα μικρότερο βέλος – δεσμός, με όνομα που περιγράφει ένα στοιχείο δεδομένων που κινείται από την μία οντότητα προς την άλλη.

Επεξήγηση συμβόλων :
Πίνακας 4.1

Όνομα	Περιγραφή	Σχήμα
Module / Οντότητα	Τμήμα από το οποίο απαρτίζεται η οντότητα	
Control / Έλεγχος	Ροή συνθήκης	
Data / Δεδομένα	Ροή δεδομένων	

4.2.1.2.1. Οντότητες με κοινές λειτουργίες

Εισαγωγή Οντότητας



Διάγραμμα 4.10

ΠΕΡΙΓΡΑΦΗ:

Η λειτουργία εισαγωγή οντότητας αποτελείται από 6 modules. Το κεντρικό module, είναι η ΕΙΣΑΓΩΓΗ ΟΝΤΟΤΗΤΑΣ. Αυτό λειτουργεί σαν συντονιστής, καλώντας όλα

τα υπόλοιπα modules έτσι ώστε να διεκπεραιωθούν οι απαραίτητες ενέργειες και να ολοκληρωθεί η λειτουργία της εισαγωγής οντότητας.

Το module ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ, είναι αυτό το οποίο διαβάζει τα δεδομένα από την είσοδο, τα οποία δίνονται από τον χρήστη.

Το module ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΕΙΣΑΓΩΓΗΣ, έχει την υπευθυνότητα να αποφανθεί πρώτα εάν τα δεδομένα είναι ορθά, να τα αποθηκεύει στη βάση και τέλος να επιστρέψει τον έλεγχο στο κεντρικό module, με την ένδειξη εάν έγινε η εισαγωγή με επιτυχία. Για να το κάνει έλεγχο αυτό καλεί 2 άλλα modules , την Επικύρωση Ορθότητας των Στοιχείων Οντότητας και την Αποθήκευση των Στοιχείων Οντότητας στη Β.Δ.

Το module ΕΠΙΚΥΡΩΣΗ ΟΡΘΟΤΗΤΑΣ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ, δέχεται από το πιο πάνω module τα στοιχεία τα οποία εισάχθηκαν , τα ελέγχει και επιστρέφει τον έλεγχο κατάστασης στην Επεξεργασία Στοιχείων Εισαγωγής.

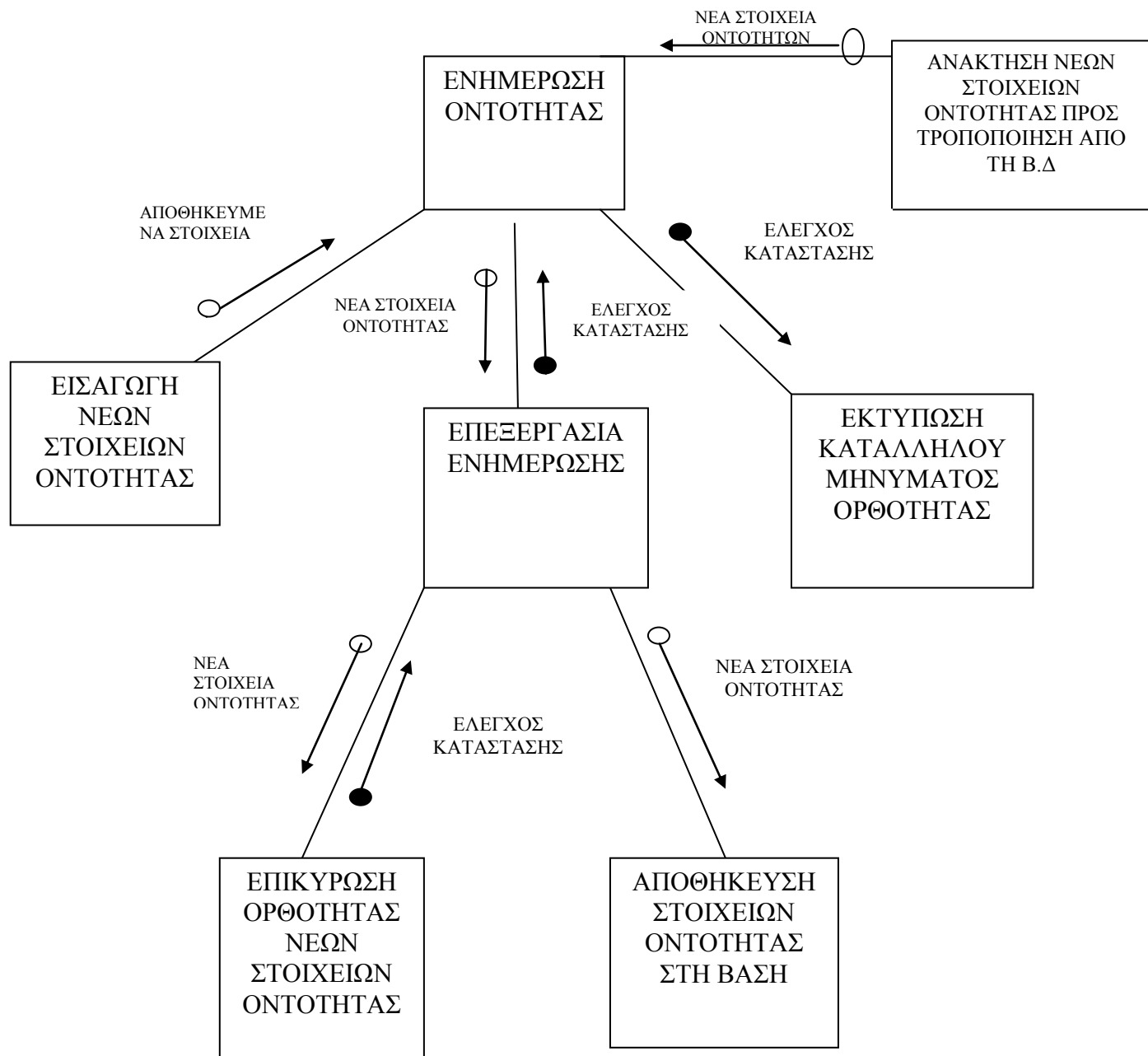
Το module ΑΠΟΘΗΚΕΥΣΗ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑ ΣΤΗ Β.Δ δέχεται τα στοιχεία που δόθηκαν κατά την εισαγωγή δεδομένων, τα οποία έχουν περάσει επιτυχώς από τον έλεγχο της ορθότητας, και τα αποθηκεύει στη βάση δεδομένων.

Το module ΕΚΤΥΠΩΣΗ ΚΑΤΑΛΛΗΛΟΥ ΜΗΝΥΜΑΤΟΣ ΟΡΘΟΤΗΤΑΣ, παίρνει σαν είσοδο την επιτυχία ή αποτυχία της εισαγωγής και αποθήκευσης και εκτυπώνει στην οθόνη το κατάλληλο μήνυμα.

ΣΧΟΛΙΑΣΜΟΣ ΣΥΝΕΚΤΙΚΟΤΗΤΑΣ ΣΥΖΕΥΞΗΣ:

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

Ενημέρωση Οντότητας



Διάγραμμα 4.11

ΠΕΡΙΓΡΑΦΗ:

Η λειτουργία ενημέρωση οντότητας οντότητας αποτελείται από 7 modules. Το κεντρικό module, είναι η ΕΝΗΜΕΡΩΣΗ ΟΝΤΟΤΗΤΑΣ. Αυτό λειτουργεί σαν συντονιστής, καλώντας όλα τα υπόλοιπα modules έτσι ώστε να γίνουν οι απαραίτητες

ενέργειες και να ενημερωθεί μία εγγραφή στην βάση δεδομένων, δηλαδή για να ολοκληρωθεί η λειτουργία της ενημέρωσης οντότητας.

Το module ΕΙΣΑΓΩΓΗ ΝΕΩΝ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ, είναι αυτό το οποίο διαβάζει τα νέα τροποποιημένα δεδομένα από την είσοδο, τα οποία δίνονται από τον χρήστη.

Το module ΑΝΑΚΤΗΣΗ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ ΠΡΟΣ ΤΡΟΠΟΠΟΙΗΣΗ ΑΠΟ ΤΗ Β.Δ είναι υπεύθυνο να φέρει τα ήδη υπάρχοντα δεδομένα της οντότητας από τη Β.Δ. και σε συνδυασμό με τα νέα στοιχεία που δίνει ο χρήστης, να αποτελέσουν την τροποποιημένη εγγραφή οντότητας.

Το module ΕΠΕΞΕΡΓΑΣΙΑ ΕΝΗΜΕΡΩΣΗΣ, έχει την υπευθυνότητα να επεξεργαστεί τα νέα στοιχεία τα οποία δίνονται από τον χρήστη και να αποφανθεί πρώτα εάν τα δεδομένα είναι ορθά, να τα αποθηκεύει στη βάση και τέλος να επιστρέψει τον έλεγχο στο κεντρικό module, με την ένδειξη εάν έγινε η εισαγωγή με επιτυχία. Για να το κάνει αυτό καλεί δύο άλλα modules , την Επικύρωση Ορθότητας των Στοιχείων Οντότητας και την Αποθήκευση των Στοιχείων Οντότητας στη Β.Δ..

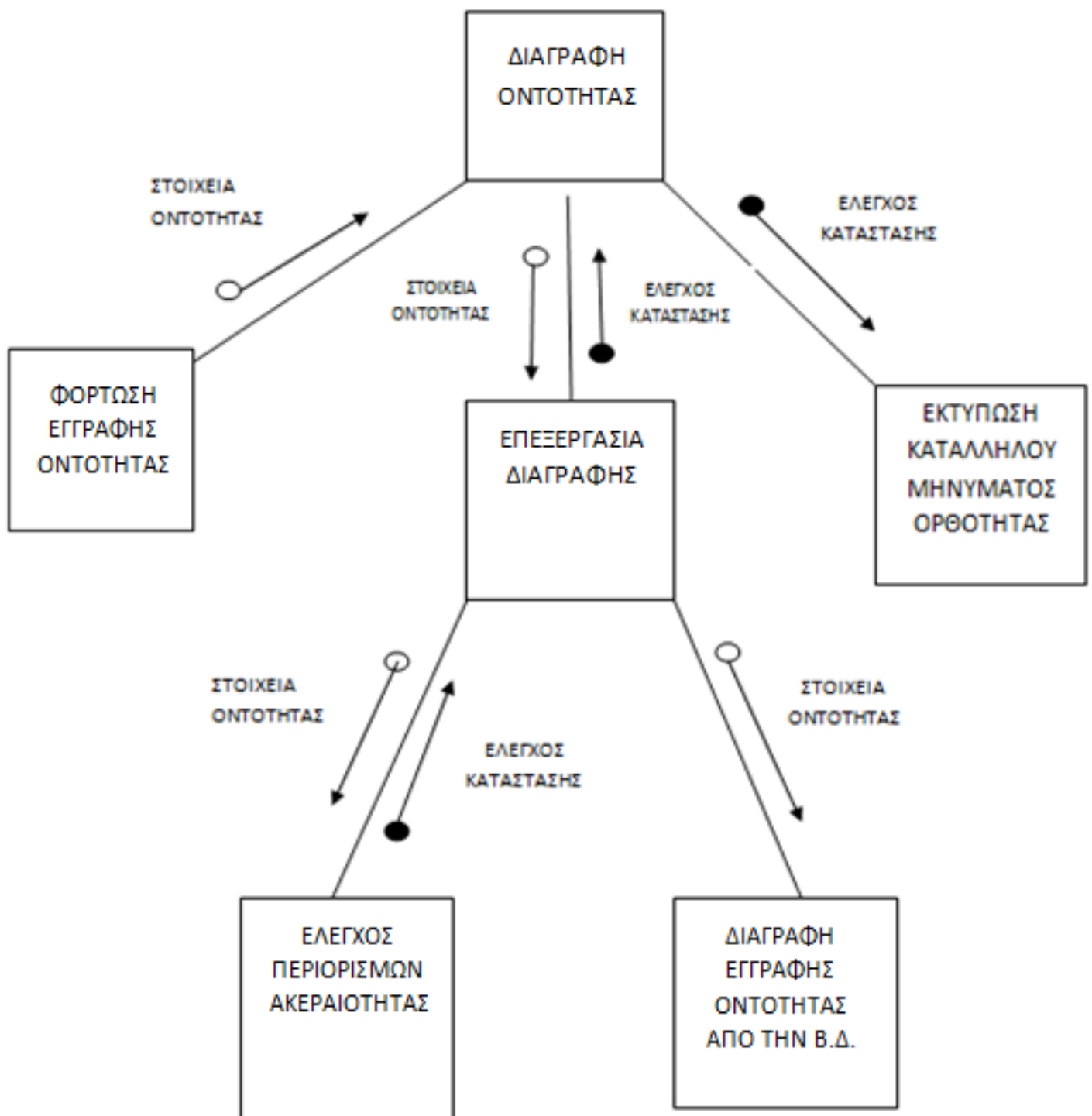
Το module ΕΠΙΚΥΡΩΣΗ ΟΡΘΟΤΗΤΑΣ ΝΕΩΝ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ, δέχεται από το πιο πάνω module τα στοιχεία τα οποία εισάχθηκαν , τα ελέγχει και επιστρέφει τον έλεγχο κατάστασης στην Επεξεργασία Ενημέρωσης.

Το module ΑΠΟΘΗΚΕΥΣΗ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑ ΣΤΗ Β.Δ. δέχεται τα στοιχεία που δόθηκαν κατά την εισαγωγή δεδομένων, τα οποία έχουν περάσει επιτυχώς από τον έλεγχο της ορθότητας, και τα αποθηκεύει στη βάση δεδομένων.

Το module ΕΚΤΥΠΩΣΗ ΚΑΤΑΛΛΗΛΟΥ ΜΗΝΥΜΑΤΟΣ ΟΡΘΟΤΗΤΑΣ, παίρνει σαν είσοδο την επιτυχία ή αποτυχία της ενημέρωσης και αποθήκευσης και εκτυπώνει στην οθόνη το κατάλληλο μήνυμα.

ΣΧΟΛΙΑΣΜΟΣ ΣΥΝΕΚΤΙΚΟΤΗΤΑΣ ΣΥΖΕΥΞΗΣ:

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.



Διάγραμμα 4.12

ΠΕΡΙΓΡΑΦΗ:

Η λειτουργία ‘διαγραφή οντότητας’ αποτελείται από 6 modules. Το κεντρικό module είναι η ΔΙΑΓΡΑΦΗ ΟΝΤΟΤΗΤΑΣ. Αυτό λειτουργεί σαν συντονιστής, καλώντας όλα τα υπόλοιπα modules έτσι ώστε να διεκπεραιωθούν οι απαραίτητες ενέργειες και να ολοκληρωθεί η διαδικασία της διαγραφής μίας εγγραφής από την βάση δεδομένων.

Το module ‘φόρτωση εγγραφής οντότητας’ είναι υπεύθυνο για την ανάκτηση των στοιχείων της εγγραφής προς διαγραφή όπως επιλέγηκε από τον χρήστη, είτε μέσω navigation στην φόρμα του πίνακα που επιθυμεί, είτε χρησιμοποιώντας την λειτουργία της αναζήτησης όπως αυτή περιγράφεται πιο κάτω.

Το module ‘επεξεργασία διαγραφής’, έχει την υπευθυνότητα να αποφανθεί πρώτα εάν η πιθανή διαγραφή της συγκεκριμένης εγγραφής θα προκαλέσει οποιαδήποτε ασυνέπεια στα δεδομένα της βάσης δεδομένων, λόγω αναφοράς της εγγραφής αυτής από εγγραφή σε κάποιο άλλο πίνακα, και έπειτα να διαγράψει από την βάση την συγκεκριμένη εγγραφή. Στο κεντρικό module θα επιστρέψει μία ενδεικτική τιμή καθορίζοντας αν έχει γίνει διαγραφή με επιτυχία ή όχι. Για να το κάνει αυτό, καλεί άλλα 2 modules, τον Έλεγχο Περιορισμών Ακεραιότητας και τη Διαγραφή Εγγραφής της οντότητας από τη ΒΔ.

Το module ‘έλεγχος περιορισμών ακεραιότητας’, δέχεται από το πιο πάνω module τα στοιχεία της εγγραφής η οποία πρόκειται να διαγραφεί, ελέγχει πρώτα εάν η πιθανή διαγραφή της συγκεκριμένης εγγραφής θα προκαλέσει οποιοδήποτε ασυνέπεια στα δεδομένα της βάσης δεδομένων και επιστρέφει τον έλεγχο κατάστασης στην επεξεργασία διαγραφής.

Το module ‘ διαγραφή εγγραφής της οντότητας από τη ΒΔ’ , δέχεται τα στοιχεία της εγγραφής η οποία πρόκειται να διαγραφεί, τα οποία έχουν περάσει επιτυχώς από τον έλεγχο περιορισμών ακεραιότητας, και τα διαγράφει από τη ΒΔ.

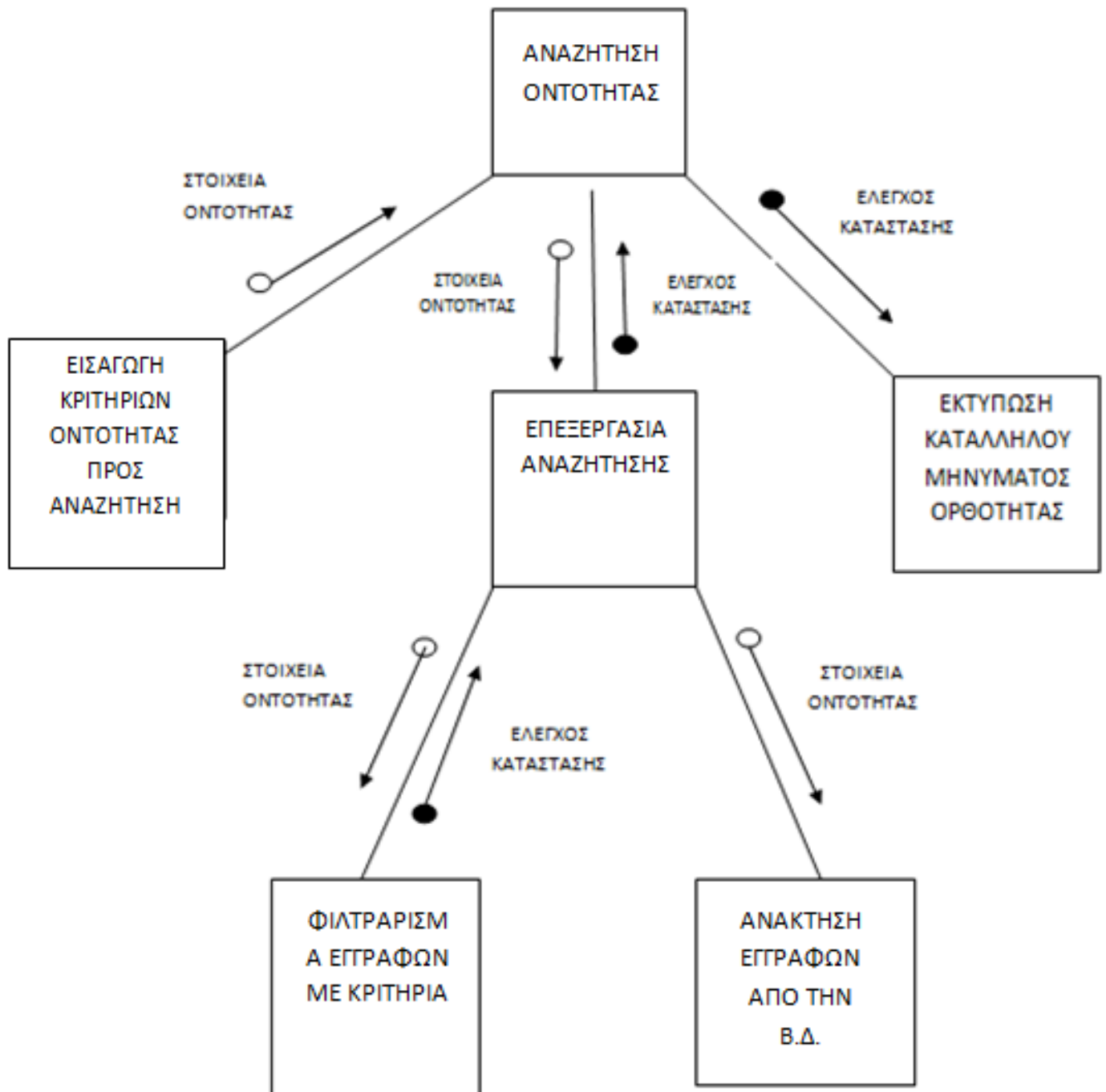
Το module ‘ εμφάνιση κατάλληλου μηνύματος ορθότητας ’, παίρνει σαν είσοδο την επιτυχία ή αποτυχία της διαγραφής και εκτυπώνει στην οθόνη το κατάλληλο μήνυμα.

ΣΧΟΛΙΑΣΜΟΣ ΣΥΝΕΚΤΙΚΟΤΗΤΑΣ / ΣΥΖΕΥΞΗΣ:

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό

αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

Αναζήτηση Οντότητας



Διάγραμμα 4.13

ΠΕΡΙΓΡΑΦΗ:

Η λειτουργία ‘αναζήτηση οντότητας’ αποτελείται από 6 modules. Το κεντρικό module είναι η ΑΝΑΖΗΤΗΣΗ ΟΝΤΟΤΗΤΑΣ. Αυτό λειτουργεί σαν συντονιστής, καλώντας όλα τα υπόλοιπα modules, έτσι ώστε να γίνονται οι απαραίτητες ενέργειες και να ολοκληρωθεί η διαδικασία της αναζήτησης εγγραφών από την βάση δεδομένων.

Το module ‘εισαγωγή κριτηρίων οντότητας προς αναζήτηση’ είναι υπεύθυνο για την ανάκτηση των κριτηρίων αναζήτησης όπως αυτά δόθηκαν από τον χρήστη.

Το module ‘επεξεργασία αναζήτησης’ είναι υπεύθυνο να αναζητήσει τις εγγραφές που πληρούν τα κριτήρια αναζήτησης καλώντας τα δύο άλλα module ‘φιλτράρισμα εγγραφών με κριτήρια’. Ακολούθως, χρησιμοποιώντας το module ‘ανάκτηση εγγραφών από την Β.Δ.’ γίνεται ανάκτηση των εγγραφών από την Β.Δ. οι οποίες πληρούν τα κριτήρια αναζήτησης τα οποία έθεσε στην αρχή ο χρήστης του συστήματος.

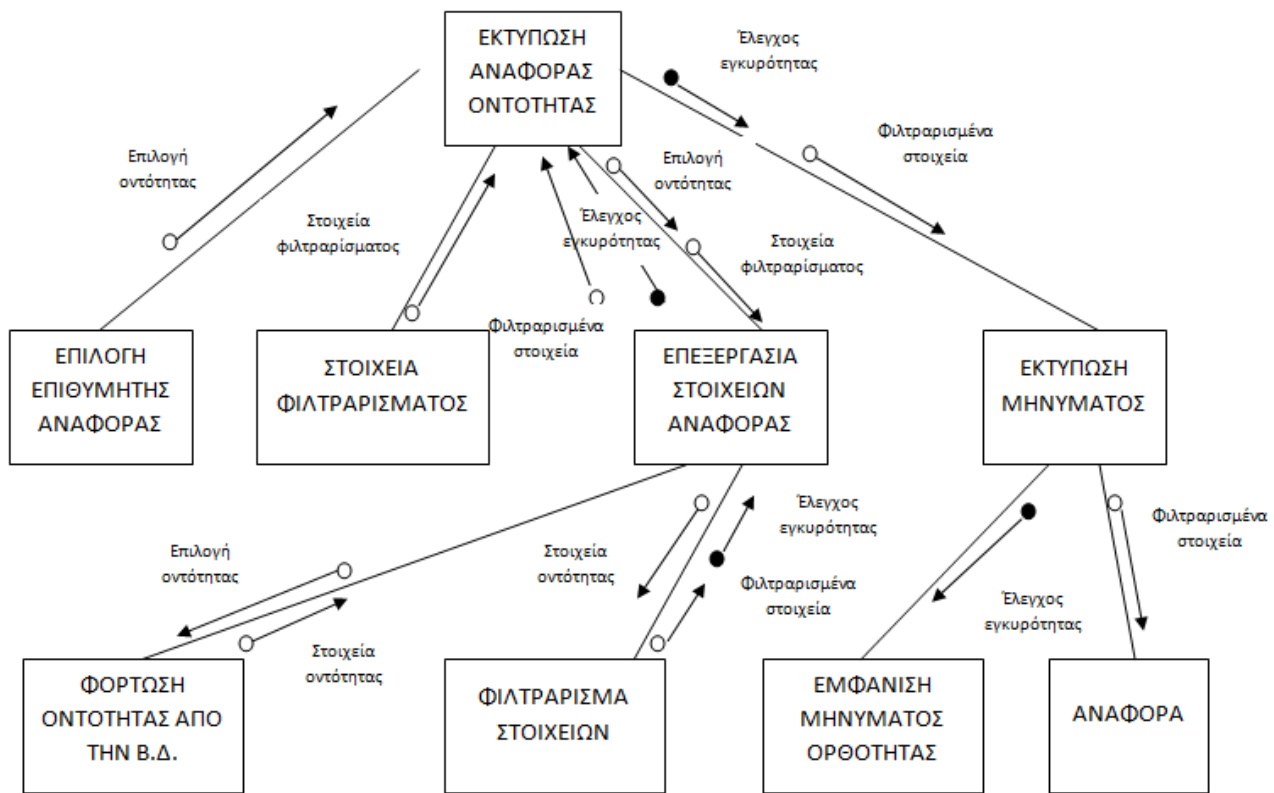
Το module ‘ανάκτηση εγγραφών από τη ΒΔ’ δέχεται τα κριτήρια με τα οποία πρόκειται να γίνει η αναζήτηση και με αναζήτηση στη ΒΔ ανακτά τις εγγραφές που ικανοποιούν τα κριτήρια. Επιστρέφει μία τιμή ελέγχου στο κεντρικό module καθορίζοντας το αν έχουν επιστραφεί καθόλου εγγραφές ή καμία εγγραφή του πίνακα δεν ικανοποιεί τα κριτήρια που έχουν δοθεί από το χρήστη. Επίσης δίνει στο κεντρικό module την λίστα εγγραφών αν υπάρχει, που ικανοποίησαν την αναζήτηση ούτως ώστε να μπορέσουν να τυπωθούν.

Το module ‘εκτύπωση κατάλληλου μηνύματος ορθότητας’, παίρνει σαν είσοδο τον έλεγχο κατάστασης της αναζήτησης καθώς επίσης και την λίστα εγγραφών που επιστράφηκαν από την ανάκτηση στη ΒΔ και αναλαμβάνει την εκτύπωση κατάλληλου μηνύματος προς τον χρήστη. Αν έχουν επιστραφεί εγγραφές από την αναζήτηση τότε η λίστα των αποτελεσμάτων εμφανίζεται στην φόρμα του πίνακα που χρησιμοποιεί ο χρήστης για την αναζήτηση. Αν δεν έχουν επιστραφεί εγγραφές, τότε τυπώνεται απλά το μήνυμα αποτυχίας αναζήτησης προς τον χρήστη για να γνωρίζει πως δεν υπάρχουν καθόλου δεδομένα με τα κριτήρια που ψάχνει.

ΣΧΟΛΙΑΣΜΟΣ ΣΥΝΕΚΤΙΚΟΤΗΤΑΣ / ΣΥΖΕΥΞΗΣ:

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

Εκτύπωση αναφοράς οντότητας



Διάγραμμα 4.14

Περιγραφή:

Η εκτύπωση αναφοράς οντότητας αποτελείται από 9 modules. Το module ΕΚΤΥΠΩΣΗ ΑΝΑΦΟΡΑΣ ΟΝΤΟΤΗΤΑΣ είναι το κεντρικό module το οποίο καλεί τα υπόλοιπα ούτως ώστε να γίνουν οι απαραίτητες ενέργειες και να εκτυπωθεί η επιλεγμένη αναφορά από την βάση δεδομένων.

Τα modules ΕΠΙΛΟΓΗ ΕΠΙΘΥΜΗΤΗΣ ΑΝΑΦΟΡΑΣ και ΣΤΟΙΧΕΙΑ ΦΙΛΤΡΑΡΙΣΜΑΤΟΣ είναι υπεύθυνα για την εισαγωγή στοιχείων στο γενικό module. Η ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΑΝΑΦΟΡΑΣ περνά την επιλογή της οντότητας στο ΦΟΡΤΩΣΗ ΟΝΤΟΤΗΤΑΣ ΑΠΟ ΤΗ Β.Δ. για να φορτωθούν τα στοιχεία της οντότητας. Αυτά μαζί με τα στοιχεία φιλτραρίσματος περνιούνται στο ΦΙΛΤΡΑΡΙΣΜΑ ΣΤΟΙΧΕΙΩΝ για να παραμείνουν μόνο τα επιλεγμένα στοιχεία. Επίσης απ' αυτό το module γίνεται και έλεγχος ορθότητας στοιχείων ως προς τον τύπο. Αφού

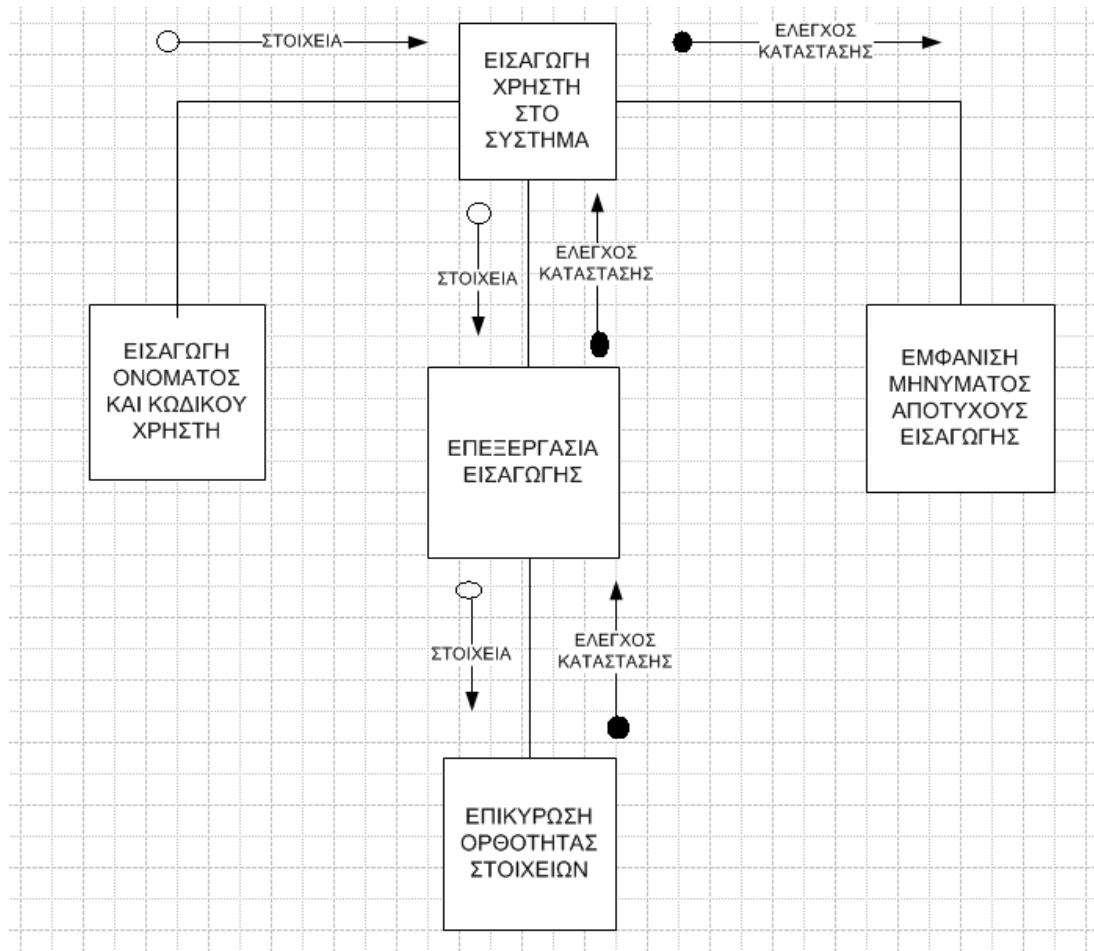
ολοκληρωθεί η εκτέλεση του module επεξεργασίας στοιχείων αναφοράς, ο έλεγχος εγκυρότητας και τα φιλτραρισμένα στοιχεία περνιούνται στο module ΕΚΤΥΠΩΣΗ ΜΗΝΥΜΑΤΟΣ το οποίο εμφανίζει στην οθόνη μήνυμα με την επιτυχή ή όχι λειτουργία των προηγούμενων modules. Αν όλες οι προηγούμενες λειτουργίες ήταν επιτυχής, τότε τα φιλτραρισμένα στοιχεία στέλνονται στον εκτυπωτή με το module ΑΝΑΦΟΡΑ.

Σχολιασμός συνεκτικότητας/σύζευξης:

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

4.2.1.2.2. Οντότητες με επιπλέον λειτουργίες ή Οντότητες με διαφορετικές λειτουργίες

Εισαγωγή Χρήστη στο Σύστημα



Διάγραμμα 4.15

ΠΕΡΙΓΡΑΦΗ

Η εισαγωγή χρήστη στο σύστημα αποτελείται από 6 modules. Το γενικό module ονομάζεται ΕΙΣΑΓΩΓΗ ΧΡΗΣΤΗ ΣΤΟ ΣΥΣΤΗΜΑ και καλεί τα υπόλοιπα με σκοπό την διεκπεραίωση της σωστής εισαγωγής του χρήστη στο σύστημα με κατάλληλες ενέργειες.

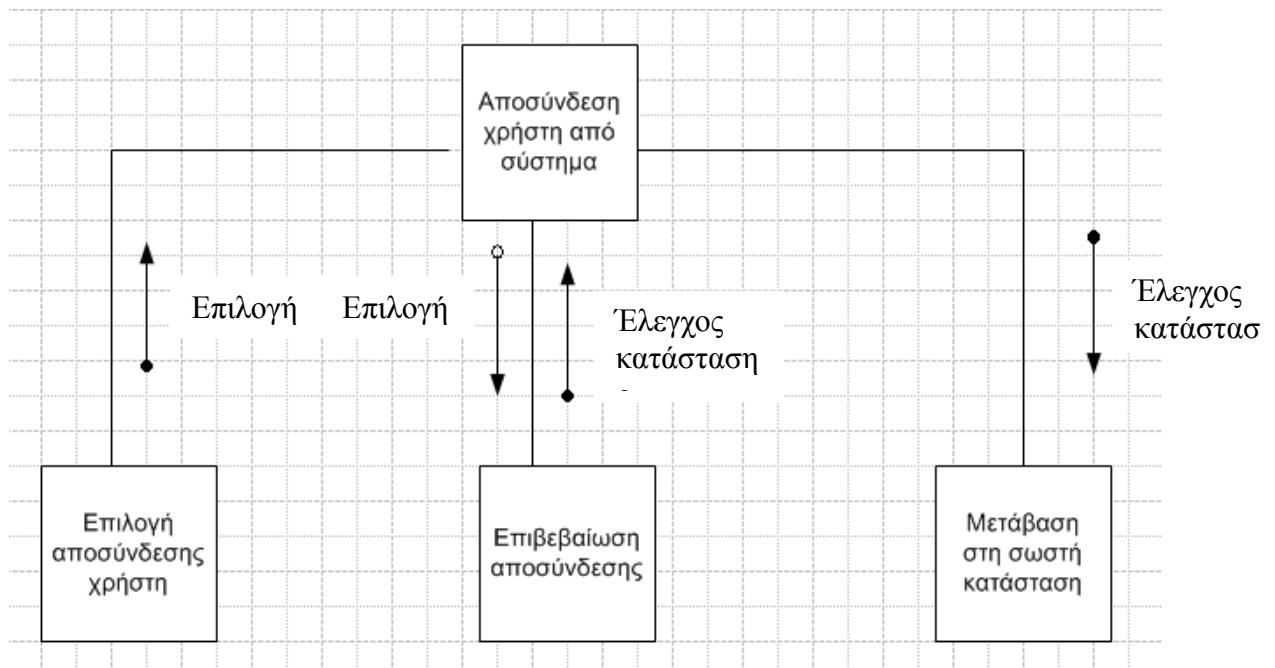
Με το module ΕΙΣΑΓΩΓΗ ΟΝΟΜΑΤΟΣ ΚΑΙ ΚΩΔΙΚΟΥ ΧΡΗΣΤΗ δίνονται από τον χρήστη το όνομα (username) και ο κωδικός (password) του για το σύστημα.

Το module επεξεργασία εισαγωγής είναι υπεύθυνο για την επεξεργασία των στοιχείων που εισαγάγει ο χρήστης. Καλεί το επικύρωση ορθότητας στοιχείων δίνοντας του σαν παράμετρο τα στοιχεία. Αυτό αναλαμβάνει να κάνει την επικύρωση των στοιχείων του χρήστη και επιστρέφει τον έλεγχο της κατάστασης πίσω στο module επεξεργασία εισαγωγής. Τέλος το module εμφάνιση μηνύματος αποτυχίας εισαγωγής είναι υπεύθυνο για την ενημέρωση του χρήστη για μη επιτυχία εισαγωγής στις περιπτώσεις που δώσει λάθος στοιχεία και επομένως δε μπορεί να εισαχθεί στο σύστημα.

Σχολιασμός συνεκτικότητας/σύζευξης:

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

Αποσύνδεση χρήστη από το σύστημα



Διάγραμμα 4.16

Περιγραφή

Η αποσύνδεση του χρήστη από το σύστημα αποτελείται από τέσσερις συναρτήσεις-modules. Υπάρχει μία γενική συνάρτηση που ονομάζεται 'αποσύνδεση χρήστη από σύστημα'. Στη συνέχεια μπορεί να καλέσει τις άλλες τρεις συναρτήσεις ανάλογα με την λειτουργία που θέλει ο χρήστης να εκτελέσει:

Επιλογή αποσύνδεσης χρήστη: όπου διαβάζεται η επιλογή του χρήστη για αποσύνδεση

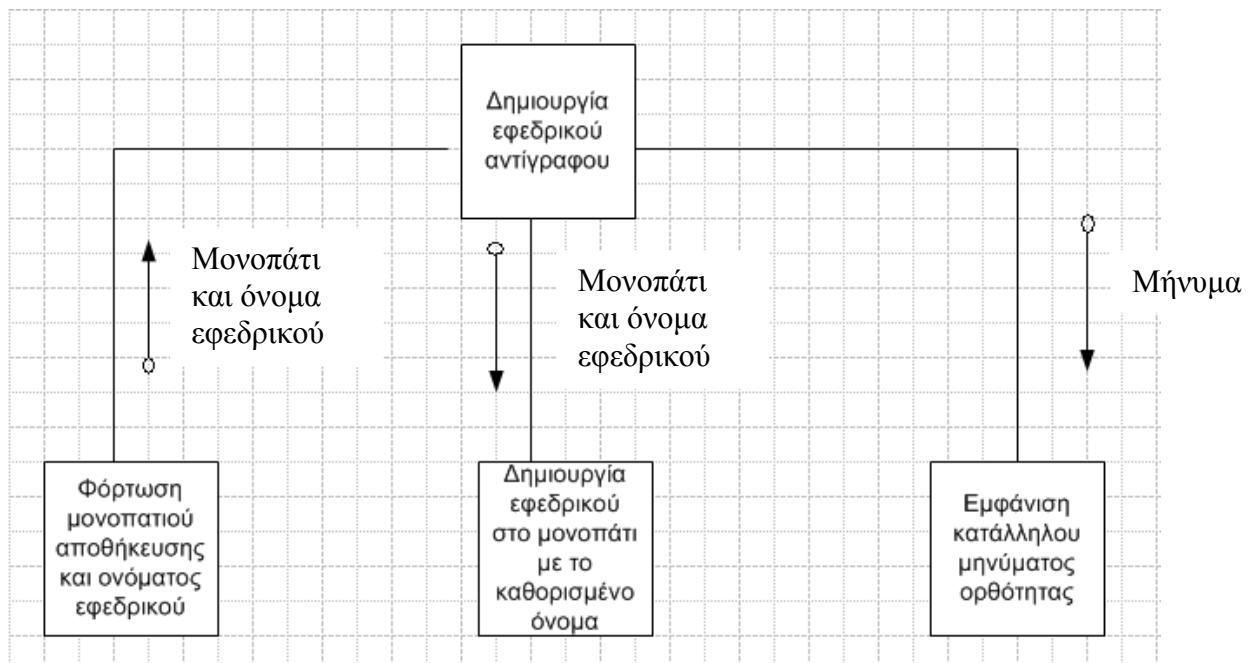
Επιβεβαίωση αποσύνδεσης: επιβεβαιώνεται η επιλογή του χρήστη για αποσύνδεση.

Μετάβαση στη σωστή κατάσταση: ο χρήστης μεταφέρεται στη φόρμα εισαγωγής χρήστη στο σύστημα, αν ο χρήστης αποσυνδέθηκε από το σύστημα επιτυχώς, αλλιώς πάει πίσω στο πρόγραμμα αν ο χρήστης δεν επιθυμεί τελικά να αποσυνδεθεί.

Συνεκτικότητα/Σύζευξη

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνουν, και που είναι όμως αναπόφευκτοι.

Δημιουργία εφεδρικού αντίγραφου



Διάγραμμα 4.17

Περιγραφή

Η δημιουργία εφεδρικού αντίγραφου αποτελείται από τέσσερις συναρτήσεις. Υπάρχει μία γενική συνάρτηση που ονομάζεται 'δημιουργία εφεδρικού αντίγραφου'. Στη συνέχεια μπορεί να καλέσει τις άλλες τρεις συναρτήσεις ανάλογα με την λειτουργία που θέλει να εκτελέσει:

Φόρτωση μονοπατιού αποθήκευσης και ονόματος: φορτώνεται το όνομα και το μονοπάτι που θέλει ο χρήστης να αποθηκεύσει το αντίγραφο

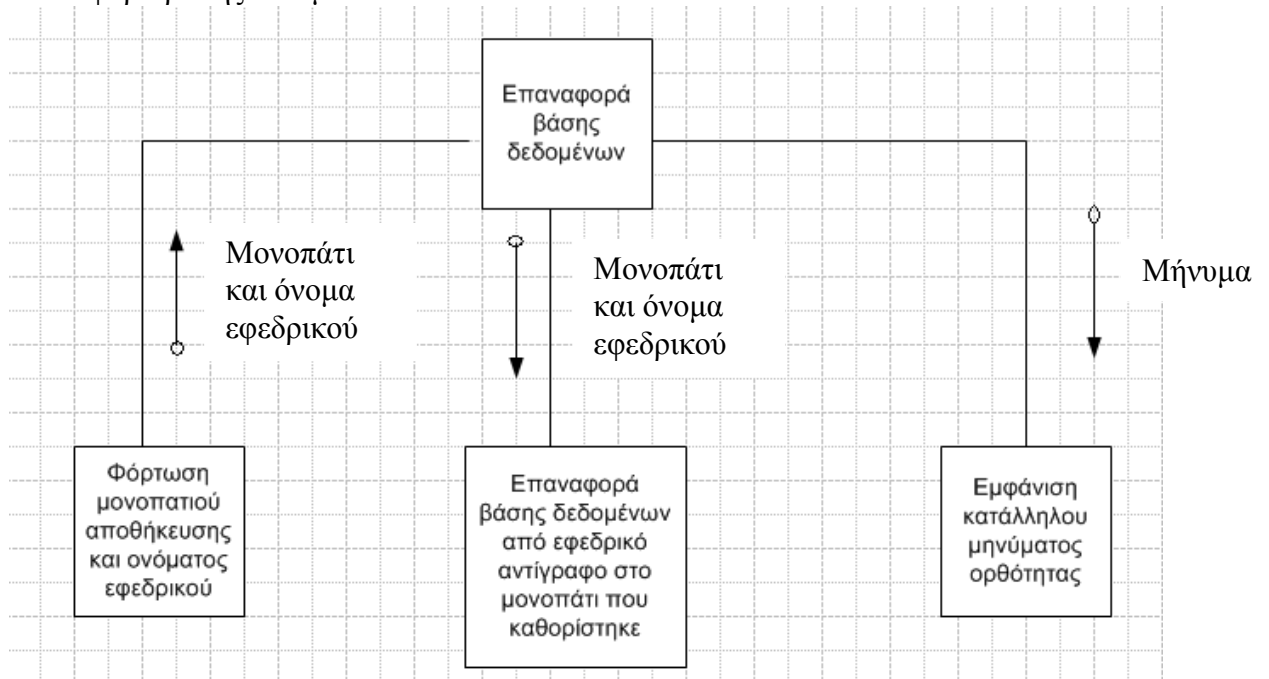
Δημιουργία εφεδρικού στο μονοπάτι με το καθορισμένο όνομα: δημιουργείται το εφεδρικό αντίγραφο

Εμφάνιση κατάλληλου μηνύματος ορθότητας: εμφανίζεται το ανάλογο μήνυμα ανάλογα αν πέτυχε ή όχι η δημιουργία του αντίγραφου.

Συνεκτικότητα/Σύζευξη

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

Επαναφορά βάσης δεδομένων



Διάγραμμα 4.18

Περιγραφή

Η επαναφορά της βάσης δεδομένων αποτελείται από τέσσερις συναρτήσεις. Υπάρχει μία γενική συνάρτηση που ονομάζεται 'επαναφορά βάσης δεδομένων'. Στη συνέχεια μπορεί να καλέσει τις άλλες τρεις συναρτήσεις ανάλογα με την λειτουργία που θέλει να εκτελέσει:

Φόρτωση μονοπατιού αποθήκευσης και ονόματος: φορτώνεται το όνομα και το μονοπάτι που θέλει ο χρήστης να αποθηκεύσει το αντίγραφο

Επαναφορά βάσης δεδομένων από εφεδρικό αντίγραφο στο μονοπάτι που καθορίστηκε: γίνεται επαναφορά της βάσης από το επιλεγμένο εφεδρικό αντίγραφο Εμφάνιση κατάλληλου μηνύματος ορθότητας: εμφανίζεται το ανάλογο μήνυμα ανάλογα αν πέτυχε ή όχι η δημιουργία του αντίγραφου.

Συνεκτικότητα/Σύζευξη

Με τον πιο πάνω σχεδιασμό επιτυγχάνεται υψηλή συνεκτικότητα αφού το κάθε module κάνει μόνο μια συγκεκριμένη διαδικασία και περνά στα modules που καλεί μόνο τις πληροφορίες τις οποίες χρειάζονται. Το επίπεδο σύζευξης δεν είναι ιδιαίτερα χαμηλό

αφού παρατηρείται σύζευξη ελέγχου-control coupling λόγω των ελέγχων καταστάσεων που χρειάζονται να γίνονται, και που είναι όμως αναπόφευκτοι.

4.3 Αναλυτική Σχεδίαση

Μετά την προκαταρκτική μελέτη ενός κατασκευαστικού μέρους ή προϊόντος, πρέπει να γίνει η εξέταση και η αναλυτική σχεδίαση κάθε στοιχείου του, έτσι ώστε να εξασφαλιστεί η εγκυρότητα των λειτουργιών που θα εκτελούνται. Παράλειψη ή αδυναμία σε οποιοδήποτε μέρος της σχεδίασης μπορεί να θέσει σε κίνδυνο την επιτυχία του προϊόντος στην αγορά, καθώς τροποποιήσεις στον σχεδιασμό κατά τα τελευταία στάδια της διαδικασίας σχεδιασμού είναι ιδιαίτερα υψηλά σε κόστος.

4.3.1 Οντότητες με κοινές λειτουργίες

4.3.1.1 Module : ΕΙΣΑΓΩΓΗ ΟΝΤΟΤΗΤΑΣ

Το module αυτό είναι υπεύθυνο για την σωστή εισαγωγή των νέων στοιχείων στην οντότητα. Κατά την εκτέλεση του, γίνεται επαλήθευση της ορθότητας των νέων στοιχείων που δίνει ο χρήστης, με βάση κάποιους περιορισμούς και επιστρέφεται μία παράμετρος κατάστασης, η οποία δείχνει κατά πόσο τα νέα στοιχεία είναι έγκυρα ή όχι. Αν ο έλεγχος είναι έγκυρος, γίνεται η προσθήκη των στοιχείων στο αρχείο, διαφορετικά εμφανίζεται μήνυμα λάθους αναφέροντας που έγινε το λάθος.

Πίνακας 4.2

Module name	Εισαγωγή οντότητας
Module type	Function
Return Type	Void
Input parameters	None
Output parameters	None
Error messages	Η ίδια η οντότητα, Log file
Files accessed	Η ίδια η οντότητα, Log file
Files changed	None
Modules called	1.Εισαγωγή στοιχείων οντότητας(arguments:στοιχεία οντότητας -string) 2.Επεξεργασία ενημέρωσης (arguments:στοιχεία οντότητας -string) 3.Έλεγχος κατάστασης-Boolean 4.Εκτύπωση μηνύματος (arguments:έλεγχος κατάστασης-Boolean)
Description	Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της εισαγωγής μιας οντότητας.

Πίνακας 4.3

Module name	ΕΙΣΑΓΩΓΗ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ
Module type	Function
Return Type	String
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, Log file
Files changed	Η ίδια η οντότητα, Log file
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την εισαγωγή των στοιχείων μιας νέας οντότητας. Τα στοιχεία αυτά δίνονται από το χρήστη μέσω μιας φόρμας και επιστρέφονται στο module που το έχει καλέσει.

Πίνακας 4.4

Module name	ΕΠΕΞΕΡΓΑΣΙΑ ΣΤΟΙΧΕΙΩΝ ΕΙΣΑΓΩΓΗΣ
Module type	Function
Return Type	Boolean
Input parameters	Στοιχεία οντότητας - string
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, Log file
Files changed	Η ίδια η οντότητα, Log file
Modules called	1.Εισαγωγή στοιχείων οντότητας(arguments:στοιχεία οντότητας -string) 2. Έλεγχος κατάστασης-boolean 3. Αποθήκευση στοιχείων οντότητας στη ΒΔ(arguments:στοιχεία οντότητας-string)
Description	Το module αυτό είναι υπεύθυνο για την επεξεργασία της εισαγωγής μιας οντότητας.

Πίνακας 4.5

Module name	ΕΠΕΞΕΡΓΑΣΙΑ ΕΜΦΑΝΙΣΗΣ ΜΗΝΥΜΑΤΟΣ
Module type	Function
Return Type	Void
Input parameters	Έλεγχος κατάστασης - boolean
Output parameters	None
Error messages	None
Files accessed	None
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να τυπώσει στην οθόνη ένα μήνυμα επιτυχίας ή αποτυχίας.

Πίνακας 4.6

Module name	ΑΠΟΘΗΚΕΥΣΗ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ ΣΤΗ ΒΔ
Module type	Function
Return Type	Void
Input parameters	Στοιχεία οντότητας –String
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, Log file
Files changed	Η ίδια η οντότητα, Log file
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την αποθήκευση των στοιχείων μιας νέας οντότητας στη ΒΔ. Εκτελεί τη λειτουργία του μέσω της σύνεσης του με τη ΒΔ και την αποθήκευση των στοιχείων της νέας οντότητας σε αυτή. Δεν επιστρέφει οτιδήποτε στο module που το έχει καλέσει.

Πίνακας 4.7

Module name	ΕΠΙΚΥΡΩΣΗ ΟΡΘΟΤΗΤΑΣ ΣΤΟΙΧΕΙΩΝ ΟΝΤΟΤΗΤΑΣ
Module type	Function
Return Type	Boolean
Input parameters	Στοιχεία οντότητας-String
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την επικύρωση της ορθότητας των στοιχείων μιας οντότητας.

4.3.1.3 Module: Τροποποίηση Οντότητας

Το module αυτό είναι υπεύθυνο για την τροποποίηση των στοιχείων στην οντότητα. Κατά την εκτέλεση του γίνεται επαλήθευση της ορθότητας των νέων στοιχείων που δίνει ο χρήστης, με βάση κάποιους περιορισμούς και επιστρέφεται μία παράμετρος κατάστασης η οποία δείχνει κατά πόσο τα νέα στοιχεία είναι έγκυρα ή όχι. Αν ο έλεγχος είναι έγκυρος τότε γίνεται ενημέρωση των στοιχείων στο αρχείο, διαφορετικά εμφανίζεται μήνυμα λάθους αναφέροντας που έγινε το λάθος.

Πίνακας 4.8

Module name	Τροποποίηση οντότητας
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, log file
Files changed	Η ίδια η οντότητα, log file
Modules called	1. Εισαγωγή νέων στοιχείων οντότητας arguments: Νέα στοιχεία οντότητας: string 2. Ενημέρωση arguments: Νέα στοιχεία οντότητας: string 3. Έλεγχος κατάστασης Boolean 4. Εμφάνιση μηνύματος arguments: έλεγχος κατάστασης Boolean
Description	Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της ενημέρωσης μιας οντότητας. Εκτελεί την λειτουργία του με την κλήση άλλων modules τα οποία είναι υπεύθυνα για την εισαγωγή, την επεξεργασία και την έξοδο της λειτουργίας.

Πίνακας 4.9

Module name	Επεξεργασία νέων στοιχείων τροποποίησης
Module type	Function
Return type	Boolean
Input parameters	Νέα στοιχεία οντότητας: string
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, log file
Files changed	Η ίδια η οντότητα, log file
Modules called	1. Επικύρωση ορθότητας νέων στοιχείων οντότητας Arguments: νέα στοιχεία οντότητας: string 2. Έλεγχος κατάστασης: boolean 3. Αποθήκευση νέων στοιχείων οντότητας στη ΒΔ Arguments: νέα στοιχεία οντότητας: string
Description	Το module αυτό είναι υπεύθυνο για την τροποποίηση μιας οντότητας. Επιστρέφει έναν έλεγχο κατάστασης στο module που το έχει καλέσει που δείχνει αν η ενημέρωση ήταν επιτυχής ή όχι.

Πίνακας 4.10

Module name	Εισαγωγή νέων στοιχείων οντότητας
Module type	Function
Return type	String
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, log file
Files changed	Η ίδια η οντότητα, log file
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την εισαγωγή νέων στοιχείων για την ενημέρωση μιας οντότητας. Εκτελεί την λειτουργία του με το διάβασμα των νέων στοιχείων από τον χρήστη μέσω μιας φόρμας και επιστρέφει τα στοιχεία αυτά στο module που το καλεί.

Πίνακας 4.11

Module name	Επικύρωση ορθότητας νέων στοιχείων οντότητας
Module type	Function
Return type	Boolean
Input parameters	Νέα στοιχεία οντότητας: string
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την επικύρωση της ορθότητας των νέων στοιχείων μιας οντότητας. Εκτελεί τη λειτουργία του με τον έλεγχο των νέων στοιχείων που έχει δώσει ο χρήστης με βάση κάποιους περιορισμούς και επιστρέφει ένα έλεγχο κατάστασης στο module που το έχει καλέσει που δείχνει αν τα νέα στοιχεία είναι έγκυρα ή όχι με βάση τους περιορισμούς.

Πίνακας 4.12

Module name	Αποθήκευση νέων στοιχείων οντότητας στη ΒΔ
Module type	Function
Return type	Void
Input parameters	Νέα στοιχεία οντότητας: string
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, log file
Files changed	Η ίδια η οντότητα, log file
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την αποθήκευση των νέων στοιχείων μιας οντότητας στη ΒΔ. Εκτελεί τη λειτουργία του με τη σύνδεση με τη ΒΔ και την αποθήκευση των νέων στοιχείων σε αυτή. Δεν επιστρέφει οτιδήποτε στο module που το έχει καλέσει.

4.3.1.4 Module: Διαγραφή Οντότητας

Το module αυτό είναι υπεύθυνο για την διαγραφή μίας εγγραφής από την οντότητα. Κατά την εκτέλεση του, ελέγχει κατά πόσο είναι δυνατή η διαγραφή της εγγραφής. Εφόσον ο έλεγχος είναι θετικός η διαδικασία ολοκληρώνεται με επιτυχία και αφαιρείται η αντίστοιχη εγγραφή από το αρχείο της οντότητας.

Πίνακας 4.13

Module name	Διαγραφή οντότητας
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, log file
Files changed	Η ίδια η οντότητα, log file
Modules called	1. Φόρτωση εγγραφής οντότητας arguments: Εγγραφή οντότητας: record 2. Επεξεργασία διαγραφής arguments: Εγγραφή οντότητας: record 3. Εκτύπωση μηνύματος arguments: Έλεγχος κατάστασης: record
Description	Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της διαγραφής μιας οντότητας. Εκτελεί την λειτουργία του με την κλήση άλλων modules τα οποία είναι υπεύθυνα για την εισαγωγή, την επεξεργασία και την έξοδο της λειτουργίας.

Πίνακας 4.14

Module name	Φόρτωση εγγραφής οντότητας
Module type	Function
Return type	Record
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για τη φόρτωση της εγγραφής που θα διαγραφεί. Εκτελεί την λειτουργία του με το διάβασμα της εγγραφής από τη λειτουργία της αναζήτησης και επιστρέφει την εγγραφή αυτή στο module που το έχει καλέσει.

Πίνακας 4.15

Module name	Επεξεργασία διαγραφής
Module type	Function
Return type	Boolean
Input parameters	Εγγραφή οντότητας : record
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα, log file
Files changed	Η ίδια η οντότητα, log file
Modules called	1. Επικύρωση ορθότητας νέων στοιχείων οντότητας Arguments: νέα στοιχεία οντότητας: string 2. Έλεγχος κατάστασης: Boolean 3. Αποθήκευση νέων στοιχείων οντότητας στη Β.Δ. arguments: νέα στοιχεία οντότητας: string
Description	Το module αυτό είναι υπεύθυνο για την επεξεργασία της διαγραφής μιας οντότητας. Εκτελεί την λειτουργία του με την κλήση άλλων modules και επιστρέφει ένα έλεγχο κατάστασης στο module που το έχει καλέσει που δείχνει αν η διαγραφή ήταν επιτυχής ή όχι.

Πίνακας 4.16

Module name	Έλεγχος περιορισμών ακεραιότητας
Module type	Function
Return type	Boolean
Input parameters	Εγγραφή οντότητας : record
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για τον έλεγχο των περιορισμών ακεραιότητας που πιθανόν παραβιάζονται με τη διαγραφή μιας οντότητας. Εκτελεί την λειτουργία του με τον έλεγχο των περιορισμών αναφορικής ακεραιότητας και επιστρέφει ένα έλεγχο κατάστασης στο module που το έχει καλέσει που δείχνει αν η διαγραφή είναι έγκυρη ή όχι με βάση τους περιορισμούς.

Πίνακας 4.17

Module Name	Διαγραφή εγγραφής οντότητας από τη ΒΔ
Module Type	Function
Return Type	void
Input Parameters	Εγγραφή οντότητας : record
Output Parameters	None
Error Messages	None
Files Accessed	Η ίδια η οντότητα : Log file
Files Changed	Η ίδια η οντότητα : Log file
Modules Called	None
Description	Το module αυτό είναι υπεύθυνο για τη διαγραφή μιας οντότητας από τη ΒΔ. Εκτελεί την λειτουργία του με την σύνδεση με τη ΒΔ και τη διαγραφή της οντότητας. Δεν επιστρέφει οτιδήποτε στο Module που το έχει καλέσει.

4.3.1.5 Module: Αναζήτηση Οντότητας

Το module αυτό είναι υπεύθυνο για την αναζήτηση των στοιχείων σε μία οντότητα. Κατά την εκτέλεση του γίνεται αναζήτηση με τα στοιχεία που έχει δώσει ο χρήστης και επιστρέφεται μία λίστα με τις φιλτραρισμένες εγγραφές.

Πίνακας 4.18

Module name	Αναζήτηση οντότητας
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Η ίδια η οντότητα
Files changed	None
Modules called	1. εισαγωγή κριτηρίων αναζήτησης οντότητας Arguments: κριτήρια αναζήτησης : string 2. επεξεργασία αναζήτησης Arguments: κριτήρια αναζήτησης : string Έλεγχος κατάστασης : Boolean Εγγραφές οντότητας :record 3. εκτύπωση μηνύματος Arguments: έλεγχος κατάστασης:Boolean Εγγραφές οντότητας :record
Description	Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της αναζήτησης μιας οντότητας . Εκτελεί την λειτουργία του με την κλήση άλλων modules τα οποία είναι υπεύθυνα για την εισαγωγή, την επεξεργασία και την έξοδο της λειτουργίας.

Πίνακας 4.19

Module name	Εισαγωγή κριτηρίων αναζήτησης οντότητας
Module type	Function
Return type	String
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	None
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την εισαγωγή των κριτηρίων για την αναζήτησης μιας οντότητας. Εκτελεί την λειτουργία του με το διάβασμα των κριτηρίων αναζήτησης από τον χρήστη μέσω μιας φόρμας και επιστρέφει τα κριτήρια αυτά στο module που το έχει καλέσει.

Πίνακας 4.20

Module name	Επεξεργασία αναζήτησης
Module type	Function
Return type	Void
Input parameters	Κριτήρια αναζήτησης : string
Output parameters	Έλεγχος κατάστασης: Boolean, εγγραφές οντότητας:record
Error messages	None
Files accessed	Η ίδια η οντότητα
Files changed	None
Modules called	- Αναζήτηση εγγράφων με κριτήρια Arguments: κριτήρια αναζήτησης : string Έλεγχος κατάστασης : Boolean - Ανάκτηση εγγράφων από Β.Δ Εγγραφές οντότητας :record
Description	Το module αυτό είναι υπεύθυνο για την επεξεργασία αναζήτησης μιας οντότητας. Εκτελεί την λειτουργία του με την κλήση άλλων module και επιστρέφει ένα έλεγχο κατάστασης και τις εγγραφές της οντότητας που πληρούν τα κριτήρια αναζήτησης στο module που το έχει καλέσει, σε περίπτωση που έχουν βρεθεί.

Πίνακας 4.21

Module name	Αναζήτηση εγγράφων με κριτήρια
Module type	Function
Return type	Boolean
Input parameters	Κριτήρια αναζήτησης : string
Output parameters	None
Error messages	None
Files accessed	Αρχείο οντότητας
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την αναζήτηση των εγγράφων που πληρούν τα κριτήρια αναζήτησης. Εκτελεί την λειτουργία του με την αναζήτηση των εγγραφών στη ΒΔ και επιστρέφει ένα έλεγχο κατάστασης στο module που το έχει καλέσει. Δείχνει αν έχουν βρεθεί εγγραφές που πληρούν τα κριτήρια αναζήτησης ή όχι.

Πίνακας 4.22

Module name	Ανάκτηση εγγραφών στην ΒΔ
Module type	Function
Return type	Record
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Αρχείο οντότητας
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την ανάκτηση των εγγράφων που πληρούν τα κριτήρια αναζήτησης. Εκτελεί τη λειτουργία του με την αναζήτηση των εγγράφων στη ΒΔ και επιστρέφει τις εγγραφές που πληρούν τα κριτήρια αναζήτησης στο module που τα έχει καλέσει.

4.3.1.6 Module: Εκτύπωση αναφοράς οντότητας

Το module αυτό είναι υπεύθυνο για την εκτύπωση των στοιχείων μίας εγγραφής οντότητας. κατά την εκτέλεση του, γίνεται η αποστολή των δεδομένων της οντότητας που έχει επιλέξει ο χρήστης στον εκτυπωτή.

Πίνακας 4.23

Module name	Εκτύπωση αναφοράς οντότητας
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Αρχείο Πελατών, Αρχείο Χρηστών, Αρχείο Πληρωμών, Αρχείο διαγωνισμών, Αρχείο Έργων
Files changed	None
Modules called	1. Επιλογή επιθυμητής αναφοράς στοιχείων οντότητας Arguments: στοιχεία οντότητας: string 2. Στοιχεία φιλτράρισμα ενημέρωσης Arguments: στοιχεία οντότητας: string 3. Επεξεργασία στοιχείων αναφοράς: string 4. Φόρτωση οντότητας από τη Β.Δ. 5. Εκτύπωση μηνύματος arguments: έλεγχος κατάστασης: Boolean
Description	Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της εκτύπωσης αναφοράς. Εκτελεί την λειτουργία του με την κλήση τριών άλλων modules τα οποία είναι υπεύθυνα για το φιλτράρισμα, την επεξεργασία και την εκτύπωση των στοιχείων της λειτουργίας.

Πίνακας 4.24

Module name	Επιλογή επιθυμητής αναφοράς
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Αρχείο Πελατών, Αρχείο Χρηστών, Αρχείο Πληρωμών
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της επιλογής αναφοράς.

Πίνακας 4.25

Module name	Στοιχεία φιλτραρίσματος
Module type	Function
Return type	Boolean
Input parameters	String
Output parameters	Valid flag
Error messages	None
Files accessed	Αρχείο Πελατών, Αρχείο Χρηστών, Αρχείο Πληρωμών
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για το φιλτράρισμα των στοιχείων. Εκτελεί τη λειτουργία του με βάση κάποιους περιορισμούς.

Πίνακας 4.26

Module name	Εισαγωγή στοιχείων φιλτραρίσματος
Module type	Function
Return type	Boolean
Input parameters	String
Output parameters	Valid flag
Error messages	None
Files accessed	Αρχείο Πελατών, Αρχείο Χρηστών, Αρχείο Πληρωμών
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο για την εισαγωγή των στοιχείων με σκοπό το φιλτράρισμά τους. Εκτελεί τη λειτουργία του με βάση κάποιους περιορισμούς.

4.3.1.7 Module: Εισαγωγή Οντότητας Εσόδου/Εξόδου

Το module αυτό είναι υπεύθυνο για την εκτέλεση της λειτουργίας της εισαγωγής εσόδου ή εξόδου ανάλογα. Εκτελεί τη λειτουργία του με την κλήση άλλων μεθόδων τα οποία είναι υπεύθυνα για την εισαγωγή, την επεξεργασία και την έξοδο της λειτουργίας. Επίσης είναι εφικτή η έκδοση τιμολογίου, απόδειξης έναντι τιμολογίου.

Πίνακας 4.27

Module name	Εισαγωγή Οντότητας Εσόδου/Εξόδου
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	Αρχείο εσόδων, αρχείο εξόδων
Files changed	None
Modules called	1. Εισαγωγή στοιχείων οντότητας (arguments: στοιχεία οντότητας : string) 2. Επεξεργασία ενημέρωσης (arguments: στοιχεία οντότητας: string) 3. Έλεγχος κατάστασης(:Boolean) 4. Εμφάνιση μηνύματος (arguments: έλεγχος κατάστασης: Boolean)
Description	Η μέθοδος αυτή είναι υπεύθυνη για την εκτέλεση της λειτουργίας της εισαγωγής εσόδου ή εξόδου ανάλογα. Εκτελεί τη λειτουργία του με την κλήση άλλων μεθόδων τα οποία είναι υπεύθυνα για την εισαγωγή, την επεξεργασία και την έξοδο της λειτουργίας. Επίσης είναι εφικτή η έκδοση τιμολογίου, απόδειξης έναντι τιμολογίου.

Πίνακας 4.28

Module name	Δημιουργία Τιμολογίου
Module type	Function
Return type	Boolean
Input parameters	String, number
Output parameters	Valid flag
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	Αρχείο εσόδων, αρχείο εξόδων
Files changed	Αρχείο τιμολογίων
Modules called	1. Εισαγωγή στοιχείων οντότητας (arguments: στοιχεία οντότητας : string) 2. Έλεγχος κατάστασης(:Boolean) 3. Εμφάνιση μηνύματος (arguments: έλεγχος κατάστασης: Boolean)
Description	Η μέθοδος αυτή είναι υπεύθυνη για την δημιουργία του τιμολογίου. Εκτελεί τη λειτουργία του με την κλήση άλλων μεθόδων. Αν ο έλεγχος είναι έγκυρος τότε το τιμολόγιο προστίθεται στο αρχείο πληρωμών.

Πίνακας 4.29

Module name	Δημιουργία Απόδειξης έναντι Τιμολογίου
Module type	Function
Return type	Boolean
Input parameters	String, number
Output parameters	Valid flag
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	Αρχείο εσόδων, αρχείο εξόδων, αρχείο τιμολογίων
Files changed	Αρχείο αποδείξεων, αρχείο τιμολογίων
Modules called	1. Εισαγωγή στοιχείων οντότητας (arguments: στοιχεία οντότητας : string) 2. Έλεγχος κατάστασης(:Boolean) 3. Εμφάνιση μηνύματος (arguments: έλεγχος κατάστασης: Boolean)
Description	Η μέθοδος αυτή είναι υπεύθυνη για την δημιουργία απόδειξης έναντι του τιμολογίου. Εκτελεί τη λειτουργία του με την κλήση άλλων μεθόδων. Αν ο έλεγχος είναι έγκυρος τότε το τιμολόγιο προστίθεται στο αρχείο πληρωμών.

4.3.1.8 Module: Διαχείριση χρήστη

Η μέθοδος αυτή είναι υπεύθυνη για την εκτέλεση της λειτουργίας της διαχείρισης χρήστη. Εκτελεί τη λειτουργία του με την κλήση άλλων μεθόδων τα οποία είναι υπεύθυνα για το login και logout.

Πίνακας 4.30

Module name	Διαχείριση χρήστη
Module type	Function
Return type	Void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	File Log
Files changed	None
Modules called	5. Εισαγωγή στοιχείων οντότητας (arguments: στοιχεία οντότητας : string) 6. Έλεγχος κατάστασης(:Boolean) 7. Εμφάνιση μηνύματος (arguments: έλεγχος κατάστασης: Boolean)
Description	Η μέθοδος αυτή είναι υπεύθυνη για την εκτέλεση της λειτουργίας της διαχείρισης χρήστη. Εκτελεί τη λειτουργία του με την κλήση άλλων μεθόδων τα οποία είναι υπεύθυνα για το login και logout.

Πίνακας 4.31

Module name	Login
Module type	Function
Return type	Boolean
Input parameters	Username String, password String
Output parameters	Success
Error messages	Λανθασμένα στοιχεία εισόδου. Δοκιμάστε ξανά
Files accessed	Αρχείο χρηστών
Files changed	None
Modules called	1. Εισαγωγή στοιχείων οντότητας (arguments: στοιχεία οντότητας : string) 2. Έλεγχος κατάστασης(:Boolean) 3. Εμφάνιση μηνύματος (arguments: έλεγχος κατάστασης: Boolean)
Description	Η μέθοδος αυτή είναι υπεύθυνη για την ελεγχόμενη πρόσβαση στο σύστημα. Διαβάζει username και password και ελέγχει αν είναι τα σωστά.

Πίνακας 4.32

Module name	Logout
Module type	Function
Return type	void
Input parameters	None
Output parameters	None
Error messages	None
Files accessed	None
Files changed	None
Modules called	None
Description	Η μέθοδος αυτή εκτελεί την έξοδο του χρήστη από το σύστημα.

4.3.1.9 Module: Δημιουργία εφεδρικού αντιγράφου

Το module αυτό είναι υπεύθυνο για τη δημιουργία εφεδρικού αντιγράφου του συστήματος (Βάσης Δεδομένων). Καλεί τα τρία modules: φόρτωση μονοπατιού και ονόματος εφεδρικού, δημιουργία εφεδρικού στο μονοπάτι με το καθορισμένο μονοπάτι και εμφάνιση μηνύματος ορθότητας, και έχει ως αποτέλεσμα έχει τη δημιουργία εφεδρικού.

Πίνακας 4.33

Module name	Δημιουργία εφεδρικού αντιγράφου
Module type	Function
Return type	Boolean
Input parameters	String, string
Output parameters	Valid flag
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	Όλα τα αρχεία του συστήματος
Files changed	None
Modules called	1. Φόρτωση μονοπατιού και ονόματος εφεδρικού Arguments: string, string 2. Δημιουργία εφεδρικού στο μονοπάτι με το καθορισμένο όνομα: Arguments: μονοπάτι, όνομα εφεδρικού: string, string 3. Εμφάνιση μηνύματος ορθότητας arguments: μήνυμα: string
Description	Το module αυτό είναι υπεύθυνο για τη δημιουργία εφεδρικού αντιγράφου του συστήματος (Βάσης Δεδομένων). Καλεί τα τρία modules, και ως αποτέλεσμα έχει τη δημιουργία εφεδρικού.

Πίνακας 4.34

Module name	Φόρτωση μονοπατιού και ονόματος εφεδρικού
Module type	Function
Return type	None
Input parameters	String, number
Output parameters	None (αλλάζει τις τιμές των inputs)
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	None
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να φορτώσει τι μονοπάτι και το όνομα που θέλει να δώσει ο χρήστης στο εφεδρικό του που θα δημιουργηθεί, ούτως ώστε να επιστραφούν στο προηγούμενο module και να γίνει η δημιουργία του αντιγράφου.

Πίνακας 4.35

Module name	Δημιουργία εφεδρικού στο μονοπάτι με το καθορισμένο όνομα
Module type	Function
Return type	None
Input parameters	String, number
Output parameters	None
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	Όλα τα αρχεία του συστήματος
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να πάρει όλα τα στοιχεία του συστήματος, από όλα τα αρχεία και να τα φυλάξει στο αρχείο και μονοπάτι που παίρνει σαν είσοδο από το module που το κάλεσε. Δημιουργεί ουσιαστικά το εφεδρικό.

Πίνακας 4.36

Module name	Εμφάνιση μηνύματος ορθότητας
Module type	Function
Return type	None
Input parameters	String
Output parameters	None
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	None
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να εμφανίσει το μήνυμα που περνιέται είτε αν έχει πετύχει είτε αν έχει αποτύχει η δημιουργία του εφεδρικού.

4.3.1.10 Module: Επαναφορά βάσης δεδομένων

Το module αυτό είναι υπεύθυνο για την επαναφορά της Β.Δ. του συστήματος από το εφεδρικό αντίγραφο του συστήματος. Καλεί τα τρία modules: Φόρτωση μονοπατιού αποθήκευσης και ονόματος εφεδρικού, Επαναφορά βάσης δεδομένων από εφεδρικό αντίγραφο και Εμφάνιση μηνύματος ορθότητας, και έχει ως αποτέλεσμα έχει τη δημιουργία του εφεδρικού.

Πίνακας 4.37

Module name	Επαναφορά βάσης δεδομένων
Module type	Function
Return type	Boolean
Input parameters	String, string
Output parameters	None
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	Όλα τα αρχεία του συστήματος
Files changed	Όλα
Modules called	1. Φόρτωση μονοπατιού αποθήκευσης και ονόματος εφεδρικού Arguments: string, string 2. Επαναφορά βάσης δεδομένων από εφεδρικό αντίγραφο: Arguments: μονοπάτι, όνομα εφεδρικού: string, string 3. Εμφάνιση μηνύματος ορθότητας arguments: μήνυμα: string
Description	Το module αυτό είναι υπεύθυνο για την επαναφορά της Β.Δ. του συστήματος από το εφεδρικό αντίγραφο του συστήματος. Καλεί τα τρία modules , και ως αποτέλεσμα έχει τη δημιουργία του εφεδρικού.

Πίνακας 4.38

Module name	Φόρτωση μονοπατιού αποθήκευσης και ονόματος εφεδρικού
Module type	Function
Return type	None
Input parameters	String, string
Output parameters	None(αλλάζει τις τιμές των input)
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	None
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να φορτώσει το μονοπάτι και το όνομα που θέλει να επαναφέρει ο χρήστης για το σύστημα, ούτως ώστε να επιστραφούν στο προηγούμενο module να γίνει η επαναφορά του από το αντίγραφο.

Πίνακας 4.39

Module name	Επαναφορά βάσης δεδομένων από εφεδρικό αντίγραφο
Module type	Function
Return type	None
Input parameters	String, string
Output parameters	None
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	Όλα τα αρχεία του συστήματος
Files changed	Όλα τα αρχεία
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να πάρει όλα τα στοιχεία του συστήματος από το εφεδρικό που καθορίζεται, και να τα φυλάξει στο αρχείο του συστήματος. Επαναφέρει ουσιαστικά τη βάση δεδομένων.

Πίνακας 4.40

Module name	Εμφάνιση μηνύματος ορθότητας
Module type	Function
Return type	None
Input parameters	String
Output parameters	None
Error messages	Λανθασμένα στοιχεία εισαγωγής
Files accessed	None
Files changed	None
Modules called	None
Description	Το module αυτό είναι υπεύθυνο να εμφανίσει το μήνυμα που περνιέται είτε αν έχει πετύχει είτε αν έχει αποτύχει η δημιουργία του εφεδρικού.

4.4 Graphical User Interface

Το φιλικό προς τον χρήστη γραφικό περιβάλλον του προγράμματος είναι ένας από τους σημαντικότερους παράγοντες στην επιτυχία ενός λογισμικού συστήματος . Το Graphical User Interface πρέπει να είναι φιλικό , εύχρηστο αλλά και έξυπνο , έτσι ώστε πάντα να δίνει στον χρήστη την εντύπωση ότι έχει το πάνω χέρι στην πλοήγηση μέσα σε ένα λογισμικό. Πρέπει να βοηθά τον χρήστη στη διεκπεραίωση του έργου του και όχι να τον καθυστερεί ή να του προκαλεί προβλήματα στην εργασία του. Με βάση τα παραπάνω ο χρήστης θα έχει την δυνατότητα:

- Να το μάθει εύκολα,
- Να αποδώσει στην εκτέλεση έργου,
- Να κάνει όσο το δυνατό λιγότερα σφάλματα κατά την λειτουργία του συστήματος,
- Να θυμάται την χρήση του, ακόμα και αν δεν έχει χρησιμοποιήσει το σύστημα αρκετό καιρό

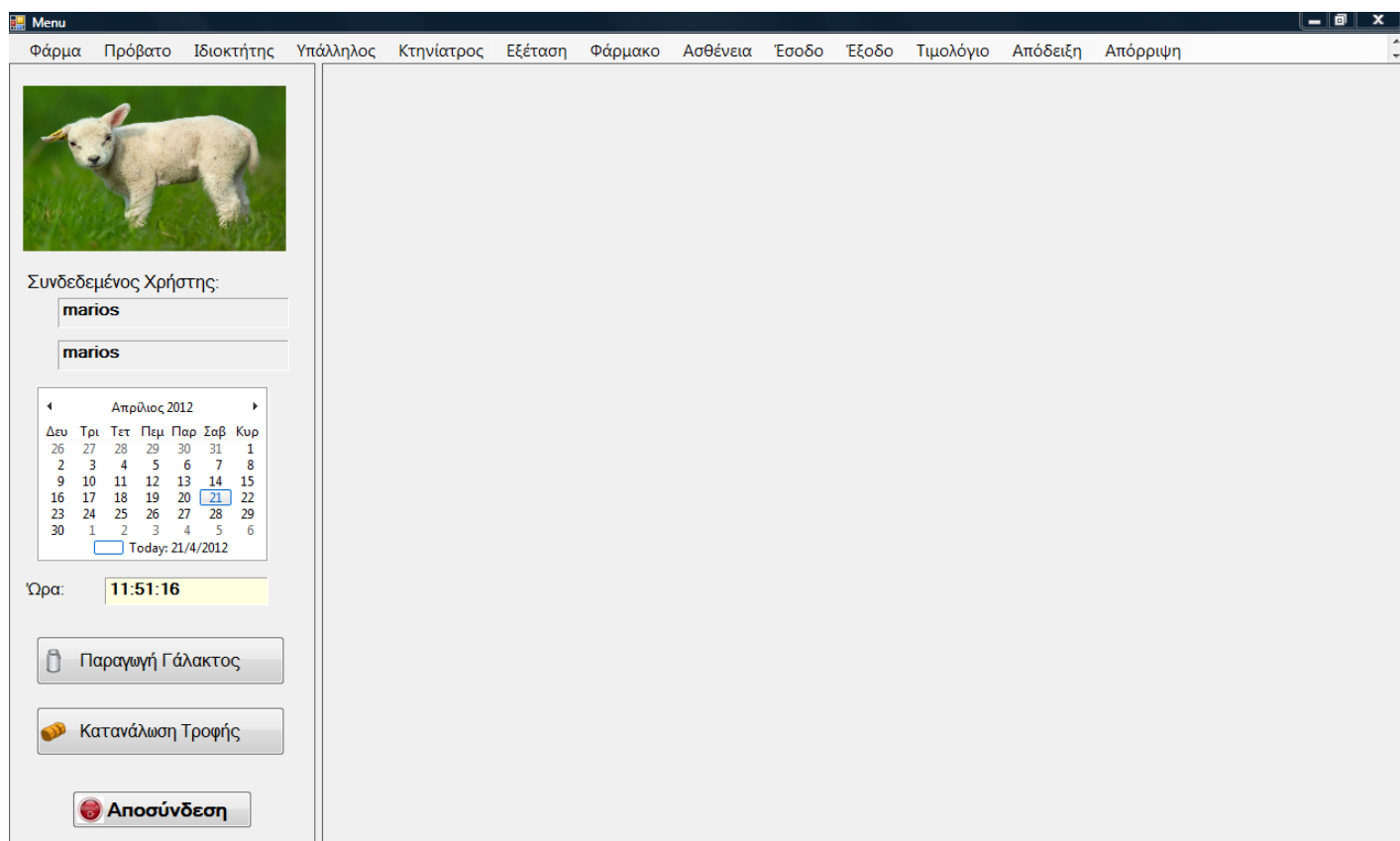
Και γενικά ο χρήστης στο τέλος της ημέρας να είναι ικανοποιημένος.

Στο γραφικό περιβάλλον εργασίας τα κουμπιά θα πρέπει να είναι τοποθετημένα στην ίδια τοποθεσία σε κάθε φόρμα , έτσι ώστε ο χρήστης να γνωρίζει που βρίσκονται τα κουμπιά σε κάθε φόρμα, λόγω συχνής χρήσης και να μην του προκαλείται σύγχυση. Έχουν χρησιμοποιηθεί ονόματα στα κουμπιά τα οποία είναι εύχρηστα και τα οποία δηλώνουν ξεκάθαρα την λειτουργία που θα έχει κάθε κουμπί. Ο χρήστης θα ξέρει από πριν τι να περιμένει σε κάθε φόρμα και δεν θα του είναι δύσκολο να χειριστεί το λογισμικό. Ακόμη, δεν πρέπει να υπάρχουν πολλά κουμπιά πάνω στη φόρμα αφού όσο πιο λίγα είναι, τόσο πιο εύχρηστο θα είναι το λογισμικό.

Τέλος, υπάρχει ένας αρμονικός συνδυασμός χρωμάτων τα οποία χρησιμοποιούνται στο γραφικό περιβάλλον εργασίας, με σκοπό να κάνουν το περιβάλλον του χρήστη πιο ευχάριστο και πιο φιλικό, να υποβοηθούν στην κατανόηση εννοιών και φυσικά, να συμβάλουν στην ικανοποίηση του χρήστη.

Με βάση τις πιο πάνω παρατηρήσεις, δημιουργήθηκαν τα ακόλουθα παραδείγματα γραφικού περιβάλλοντος εργασίας:

➤ Κυρίως Μενού Επιλογών



Εικόνα 4.1

Στην πιο πάνω εικόνα παρουσιάζεται το πώς θα είναι το κυρίως μενού επιλογών που θα μπορεί να χρησιμοποιήσει ο χρήστης του συστήματος. Στο πάνω σημείο της εικόνας παρουσιάζεται το βασικό μενού επιλογών, από το οποίο ο χρήστης θα μπορεί να διεκπεραιώνει τις βασικές λειτουργίες των οντοτήτων του συστήματος, όπως έχουν περιγραφεί σε προηγούμενες παραγράφους. Στα αριστερά της εικόνας, υπάρχει το λογότυπο της κτηνοτροφικής μονάδας, ακολούθως υπάρχει το όνομα και το επίθετο του συνδεδεμένου χρήστη, υπάρχει το ημερολόγιο και η ώρα. Ο χρήστης, στο σημείο αυτό της εικόνας, μπορεί να βρει τα κουμπιά τα οποία θα του επιτρέψουν να έχει πρόσβαση στις λειτουργίες: Παραγωγή Γάλακτος: όπου θα μπορεί να καταχωρεί την ημερήσια παραγωγή γάλακτος για κάθε πρόβατο, να ομαδοποιεί κάθε πρόβατο και να βλέπει τις αναφορές της παραγωγής γάλακτος των προβάτων, και Κατανάλωση τροφής: όπου θα μπορεί να καταχωρεί την ημερήσια κατανάλωση τροφής για ομάδες προβάτων, να ομαδοποιεί κάθε πρόβατο και να βλέπει τις αναφορές της κατανάλωσης τροφής των προβάτων. Επιπλέον, κάτω αριστερά στην εικόνα υπάρχει το κουμπί Αποσύνδεσης από

το οποίο ο χρήστης βγαίνει από το σύστημα και μεταφέρεται στην αρχική φόρμα εισαγωγής στο σύστημα.

➤ Εισαγωγή Ιδιοκτήτη

Εισαγωγή Ιδιοκτήτη

Όνομα

Επίθετο

Αρ. Ταυτότητας

Φύλο ☐ Άρρεν ☐ Θήλυ

Ημερ. Γέννησης ,

Διεύθυνση

Αριθμός

Οδός

Ταχυδρ. Τομέας

Πόλη

Χώρα

Log In

Όνομα

Συνθηματικό

Τηλέφωνο

Κινητό

Fax

Email

Εικόνα 4.2

Στην πιο πάνω εικόνα παρουσιάζεται ένα παράδειγμα εισαγωγής δεδομένων στο σύστημα, και συγκεκριμένα παρουσιάζεται η φόρμα εισαγωγής νέου ιδιοκτήτη στο σύστημα. Όπως φαίνεται πιο πάνω, ο χρήστης μπορεί να εισάγει δεδομένα πληκτρολογώντας τα στα πεδία εισαγωγής, είτε επιλέγοντας τα από μία λίστα επιλογών, είτε με επιλογή μίας συγκεκριμένης τιμής. Στο κάτω μέρος της φόρμας παρουσιάζονται τα κουμπιά: Αποθήκευση: όπου τα στοιχεία του νέου ιδιοκτήτη, αφού περάσουν από έλεγχο ορθότητας, αποθηκεύονται τελικά στον κατάλληλο πίνακα της Β.Δ., Αναίρεση: ο χρήστης με την επιλογή αυτή μπορεί να αφαιρέσει όλες τις τιμές που έχουν τα πεδία της φόρμας, και Ακύρωση: ο χρήστης μπορεί να φύγει από την φόρμα χωρίς να

αποθηκεύσει τις αλλαγές στα στοιχεία του ιδιοκτήτη, και άρα δεν δημιουργείται καινούργια εγγραφή στον πίνακα της Β.Δ..

➤ Ενημέρωση Ιδιοκτήτη

Ενημέρωση Ιδιοκτήτη

Όνομα

Επίθετο

Αρ. Ταυτότητας

Φύλο ☐ Άρρεν ☐ Θήλυ

Ημερ. Γέννησης

Διεύθυνση

Αριθμός

Οδός

Ταχυδρ. Τομέας

Πόλη

Χώρα

Τηλέφωνο

Κινητό

Fax

Email

Log In

Όνομα

Συνθηματικό

Εικόνα 4.3

Στην πιο πάνω εικόνα παρουσιάζεται ένα παράδειγμα ενημέρωσης δεδομένων του συστήματος, και συγκεκριμένα παρουσιάζεται η φόρμα ενημέρωσης κάποιου υπάρχον ιδιοκτήτη του συστήματος. Όπως φαίνεται πιο πάνω, ο χρήστης μπορεί να ενημερώσει δεδομένα είτε πληκτρολογώντας τα στα πεδία εισαγωγής, είτε επιλέγοντας τα από μία λίστα επιλογών, είτε με επιλογή μίας συγκεκριμένης τιμής. Στο κάτω μέρος της φόρμας παρουσιάζονται τα κουμπιά: Ενημέρωση: όπου τα στοιχεία του ιδιοκτήτη, αφού περάσουν από έλεγχο ορθότητας, αποθηκεύονται τελικά στον κατάλληλο πίνακα της Β.Δ., Αναίρεση: ο χρήστης με την επιλογή αυτή μπορεί να αφαιρέσει όλες τις τιμές που

έχουν τα πεδία της φόρμας, και Ακύρωση: ο χρήστης μπορεί να φύγει από την φόρμα χωρίς να αποθηκεύσει τις αλλαγές στα στοιχεία του ιδιοκτήτη, και άρα δεν ενημερώνεται η υπάρχων εγγραφή στον πίνακα της Β.Δ..

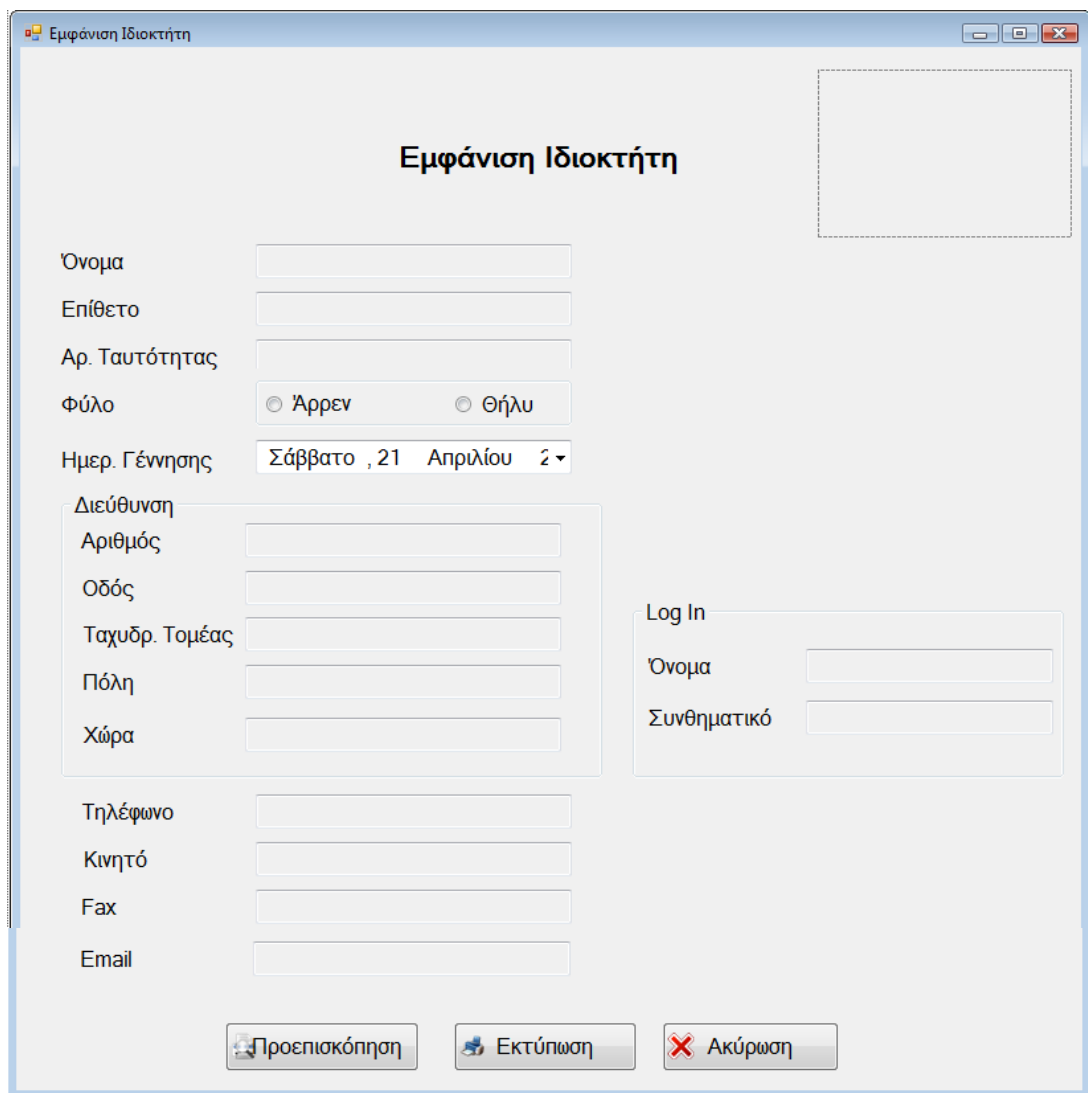
➤ Διαγραφή Ιδιοκτήτη

Εικόνα 4.4

Στην πιο πάνω εικόνα παρουσιάζεται ένα παράδειγμα διαγραφής δεδομένων του συστήματος, και συγκεκριμένα παρουσιάζεται η φόρμα διαγραφής κάποιου υπάρχων ιδιοκτήτη του συστήματος. Όπως φαίνεται πιο πάνω, ο χρήστης πρέπει πρώτα να αναζητήσει τον ιδιοκτήτη. Αυτό επιτυγχάνεται πληκτρολογώντας στοιχεία σε μερικά, όλα ή καθόλου από τα πεδία που βρίσκονται στα αριστερά της φόρμας, τα οποία είναι τα κριτήρια αναζήτησης του ιδιοκτήτη. Αφού γίνει η αναζήτηση, τα στοιχεία των ιδιοκτητών που πληρούν τα κριτήρια αναζήτησης θα παρουσιαστούν στον πίνακα που

βρίσκεται στο δεξιό μισό της φόρμας, και τότε ο χρήστης μπορεί να επιλέξει τον χρήστη που επιθυμεί να διαγραφεί. Στο κάτω μέρος της φόρμας παρουσιάζονται τα κουμπιά: Αναζήτηση: όπου ο χρήστης μπορεί να αναζητήσει τα στοιχεία των ιδιοκτητών που πληρούν τα κριτήρια αναζήτησης, Διαγραφή: ο ιδιοκτήτης ο οποίος επιλέχθηκε από τον χρήστη μέσω του πίνακα της φόρμας, διαγράφεται από τον αντίστοιχο πίνακα της Β.Δ., Αναίρεση: ο χρήστης με την επιλογή αυτή μπορεί να αφαιρέσει όλες τις τιμές που έχουν τα πεδία της φόρμας, και Ακύρωση: ο χρήστης μπορεί να φύγει από την φόρμα χωρίς να αποθηκεύσει τις αλλαγές στα στοιχεία του ιδιοκτήτη, και άρα δεν διαγράφεται η υπάρχων εγγραφή στον πίνακα της Β.Δ..

➤ Παρουσίαση - Εκτύπωση Ιδιοκτήτη



Εμφάνιση Ιδιοκτήτη

Όνομα

Επίθετο

Αρ. Ταυτότητας

Φύλο ☐ Άρρεν ☐ Θήλυ

Ημερ. Γέννησης

Διεύθυνση

Αριθμός

Οδός

Ταχυδρ. Τομέας

Πόλη

Χώρα

Τηλέφωνο

Κινητό

Fax

Email

Log In

Όνομα

Συνθηματικό

Εικόνα 4.5

Στην πιο πάνω εικόνα παρουσιάζεται ένα παράδειγμα παρουσίασης και εκτύπωσης δεδομένων του συστήματος, και συγκεκριμένα παρουσιάζεται η φόρμα εμφάνισης κάποιου υπάρχον ιδιοκτήτη του συστήματος. Όπως φαίνεται πιο πάνω, ο χρήστης μπορεί να δει όλα τα στοιχεία κάποιου συγκεκριμένου ιδιοκτήτη. Στο κάτω μέρος της φόρμας παρουσιάζονται τα κουμπιά: Προεπισκόπηση: όπου ο χρήστης μπορεί να δει τα στοιχεία του ιδιοκτήτη όπως αυτά θα εκτυπωθούν, Εκτύπωση: ο χρήστης επιλέγει τον εκτυπωτή από τον οποίο θα τυπωθούν τα στοιχεία του ιδιοκτήτη, ο οποίος παρουσιάζεται στην φόρμα ,και Ακύρωση: ο χρήστης μπορεί να φύγει από την φόρμα χωρίς να εκτυπώσει τα στοιχεία του ιδιοκτήτη.

Κεφάλαιο 5. Φάση Υλοποίησης

Κεφάλαιο 5. Φάση Υλοποίησης	191
5.1 Εισαγωγή	191
5.2 Σύντομη Περιγραφή Λειτουργικότητας του Προγράμματος	191
5.3 Εργαλεία και Τεχνολογίες	194
5.3.1 Microsoft C - Sharp .NET.....	195
5.3.2 Microsoft SQL Server®2008.....	196
5.3.3 SAP Crystal Reports	197
5.3.4 Περιγραφή βάσης δεδομένων	197
5.3.5 Δύσκολα σημεία στον κώδικα	199
5.3.6 Πηγαίος κώδικας με σχόλια.....	202

Κεφάλαιο 5. Φάση Υλοποίησης

5.1 Εισαγωγή

Σκοπός της φάσης αυτής είναι η καταγραφή της γενικής σύνοψης της υφιστάμενης κατάστασης του συστήματος που αναπτύχθηκε, και των γενικών στοιχείων που πιστοποιούν το σύστημα. Δηλαδή, ο σκοπός του εγγράφου της υλοποίησης είναι η τεκμηρίωση της φάσης της υλοποίησης.

Επιπλέον, σκοπός της φάσης αυτής είναι η επεξήγηση των εργαλείων και των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν και της βάσης δεδομένων. Γίνεται μία περιφραστική αναφορά σε συγκεκριμένα σημεία του κώδικα με ιδιαίτερη επισήμανση στα επίμαχα σημεία δυσκολίας και δίνεται μία μικρή περιγραφή του τρόπου επίλυσης των προβλημάτων.

Το πιο σημαντικό και ουσιαστικό κομμάτι του εγγράφου αυτού είναι ο ενδεδειγμένος έλεγχος των περιπτώσεων και υποπεριπτώσεων του προγράμματος που διεξάγεται μέσω test cases για να διαβεβαιωθεί η ορθή λειτουργία του συστήματος που υλοποιήθηκε και ότι η φάση της υλοποίησης έχει ακολουθήσει τα βήματα που προηγήθηκαν και η όλη λειτουργικότητα του προγράμματος έχει γίνει με τον τρόπο που έχει από προηγουμένως ορισθεί.

Αφού ολοκληρωθεί η φάση της υλοποίησης και του εγγράφου, θα πρέπει τα προηγούμενα παραδοτέα έγγραφα που έχουν δημιουργηθεί, να είναι όσο το δυνατόν πιο ολοκληρωμένα και κατανοητά από τον πελάτη και όχι μόνο από την ομάδα υλοποίησης του λογισμικού.

5.2 Σύντομη Περιγραφή Λειτουργικότητας του Προγράμματος

Το υλοποιημένο σύστημα είναι το σύστημα αποθήκευσης και διαχείρισης των δεδομένων μίας κτηνοτροφικής μονάδας, συμπεριλαμβανομένων των ιδιοκτητών, υπαλλήλων προβάτων, φαρμών, εσόδων, εξόδων, κτηνιατρικών εξετάσεων, τιμολογίων και αποδείξεων. Επίσης, καταγράφει την ημερήσια παραγωγή γάλακτος για κάθε πρόβατο, και την ημερήσια κατανάλωση τροφής τους, τα οποία σε συνδυασμό με τις κτηνιατρικές τους εξετάσεις και την κρεοπαραγωγή, μας βοηθούν να αποφασίσουμε αν ένα πρόβατο θα παραμείνει στην φάρμα, θα πωληθεί ή θα σφαχτεί. Οι πιο πάνω λειτουργίες υποστηρίζονται από μία βάση δεδομένων και μία διεπιφάνεια χρήστη.

Η βάση του συστήματος περιλαμβάνει τα δεδομένα των ιδιοκτητών, υπαλλήλων, προβάτων, φαρμών, κτηνιάτρων, ασθενειών, φαρμάκων, εξετάσεων, εσόδων, εξόδων, τιμολογίων, αποδείξεων, καθώς επίσης και τα δεδομένα για την ημερήσια παραγωγή γάλακτος και την ημερήσια κατανάλωση τροφής για κάθε πρόβατο. Γενικά, η βάση δεδομένων είναι υπεύθυνη για την διεκπεραίωση των διεργασιών που αφορούν το πιο πάνω σύστημα διαχείρισης.

Το σύστημα αποθήκευσης και διαχείρισης δεδομένων των ιδιοκτητών, υπαλλήλων, κτηνιάτρων, φαρμών, εσόδων και εξόδων είναι υπεύθυνο για την εισαγωγή στοιχείων των πιο πάνω οντοτήτων, την ενημέρωση τους στην ανάγκη αλλαγών, την διαγραφή τους και την παρουσίαση και εκτύπωση τους από τον χρήστη του συστήματος.

Το σύστημα διαχείρισης των τιμολογίων της κτηνοτροφικής μονάδας χωρίζεται σε δύο κατηγορίες: στα τιμολόγια που αφορούν τα έσοδα, και στα τιμολόγια που αφορούν τα έξοδα της κτηνοτροφικής μονάδας. Κατά την διαχείριση των τιμολογίων, ο χρήστης πρέπει να εισάγει τον ιδιοκτήτη ο οποίος εκδίδει το τιμολόγιο, την φάρμα στην οποία αναφέρεται, το έσοδο ή έξοδο για το οποίο εκδίδεται το συγκεκριμένο τιμολόγιο, την μονάδα μέτρησης και το πλήθος των τεμαχίων του συγκεκριμένου εσόδου, και τέλος, την ημερομηνία έκδοσης του. Με βάση τα πιο πάνω στοιχεία, ο χρήστης έχει την δυνατότητα να εισάγει, να ενημερώσει να διαγράψει και να τυπώσει ένα συγκεκριμένο τιμολόγιο.

Το σύστημα διαχείρισης των αποδείξεων της κτηνοτροφικής μονάδας χωρίζεται σε δύο κατηγορίες: στις αποδείξεις οι οποίες σχετίζονται με τιμολόγια που αφορούν τα έσοδα, και στις αποδείξεις οι οποίες σχετίζονται με τιμολόγια που αφορούν τα έξοδα της κτηνοτροφικής μονάδας. Κατά την διαχείριση των αποδείξεων, ο χρήστης πρέπει να εισάγει το τιμολόγιο με το οποίο σχετίζεται μία συγκεκριμένη απόδειξη, την ημερομηνία και το ποσό πληρωμής. Όταν ο χρήστης εισάγει ένα ποσό, τότε αυτόματα ενημερώνεται το υπολειπόμενο και το πληρωμένο ποσό του αντίστοιχου τιμολογίου. Με βάση τα πιο πάνω στοιχεία, ο χρήστης έχει την δυνατότητα να εισάγει, να ενημερώσει να διαγράψει και να τυπώσει μία συγκεκριμένη απόδειξη.

Το σύστημα διαχείρισης των κτηνιατρικών εξετάσεων της κτηνοτροφικής μονάδας είναι υπεύθυνο για την διαχείριση των στοιχείων που αφορούν μία κτηνιατρική εξέταση ενός προβάτου από ένα συγκεκριμένο κτηνίατρο. Δηλαδή, κατά την διαχείριση της κτηνιατρικής εξέτασης, ο χρήστης πρέπει να επιλέξει ένα κτηνίατρο από μία λίστα επιλογών, το πρόβατο στο οποίο έγινε η εξέταση, τις ασθένειες που διαγνώστηκαν στο

πρόβατο κατά την συγκεκριμένη εξέταση, καθώς και τα φάρμακα που θα του χορηγηθούν, και εισάγονται και κάποια σχόλια σχετικά με την εξέταση. Με την καταχώρηση των στοιχείων αυτών, ενημερώνονται και οι αντίστοιχοι πίνακες της Β.Δ.. Με βάση τα πιο πάνω στοιχεία, ο χρήστης έχει την δυνατότητα να εισάγει, να ενημερώσει να διαγράψει και να τυπώσει μία συγκεκριμένη κτηνιατρική εξέταση.

Το σύστημα διαχείρισης της ημερήσιας παραγωγής γάλακτος των προβάτων της κτηνοτροφικής μονάδας είναι υπεύθυνο για την διαχείριση των στοιχείων που αφορούν την ημερήσια παραγωγή γάλακτος για το κάθε πρόβατο της κτηνοτροφικής μονάδας. Δηλαδή, κατά την διαχείριση της παραγωγής γάλακτος, ο χρήστης πρέπει να επιλέξει ένα πρόβατο, ανάλογα με την θέση στην οποία βρίσκεται στο γαλευτόρι, την ποσότητα γάλακτος που είχε το συγκεκριμένο πρόβατο και την ημερομηνία καταγραφής του. Στο τέλος κάθε μήνα, τα στοιχεία του πίνακα της Β.Δ. που αφορούν την ημερήσια γαλακτοπαραγωγή των προβάτων, εξάγονται σε ένα αρχείο Excel, και ακολούθως τα στοιχεία αυτά διαγράφονται από τον πίνακα της βάσης δεδομένων. Με βάση τα πιο πάνω στοιχεία, ο χρήστης έχει την δυνατότητα να εισάγει παραγωγές γάλακτος, να ομαδοποιήσει τα πρόβατα ανάλογα με την θέση που έχουν στο γαλευτόρι και να τυπώσει μία αναφορά για την συνολική γαλακτοπαραγωγή, ή για το κάθε πρόβατο ξεχωριστά.

Το σύστημα διαχείρισης της ημερήσιας κατανάλωσης τροφής των προβάτων της κτηνοτροφικής μονάδας είναι υπεύθυνο για την διαχείριση των στοιχείων που αφορούν την ημερήσια κατανάλωσης τροφής για μία ομάδα προβάτων της κτηνοτροφικής μονάδας. Δηλαδή, κατά την διαχείριση της κατανάλωσης τροφής, ο χρήστης επιλέγει μία ομάδα προβάτων, ανάλογα με την θέση στην οποία βρίσκονται στην φάρμα, την ποσότητα τροφής που κατανάλωσαν τα συγκεκριμένα πρόβατα και την ημερομηνία καταγραφής του. Στο τέλος κάθε μήνα, τα στοιχεία του πίνακα της Β.Δ. που αφορούν την ημερήσια κατανάλωση τροφής των προβάτων, εξάγονται σε ένα αρχείο Excel, και ακολούθως τα στοιχεία αυτά διαγράφονται από τον πίνακα της βάσης δεδομένων. Με βάση τα πιο πάνω στοιχεία, ο χρήστης έχει την δυνατότητα να εισάγει κατανάλωση τροφής, να ομαδοποιήσει τα πρόβατα ανάλογα με την θέση που έχουν στην φάρμα, και να τυπώσει μία αναφορά για την συνολική κατανάλωση τροφής, ή για το κάθε πρόβατο ξεχωριστά.

Τέλος, οποιαδήποτε στοιχεία τα οποία είναι βοηθητικά για λειτουργίες και υπολογισμούς της βάσης δεδομένων φυλάσσονται στην βάση δεδομένων, έτσι ώστε να

αποφεύγεται η επανάληψη και η συνεχής πληκτρολόγηση τους από τον χρήστη του συστήματος.

5.3 Εργαλεία και Τεχνολογίες

Κατά την ανάπτυξη ενός λογισμικού συστήματος, σημαντικό ρόλο παίζουν τα εργαλεία με τα οποία οι απαιτήσεις και οι προδιαγραφές που θέτει ο ιδιοκτήτης της κτηνοτροφικής μονάδας, μπορούν να υλοποιηθούν. Όσο πιο τεχνολογικά ανεπτυγμένα είναι τα εργαλεία αυτά, τόσο μεγαλύτερη ευχρηστία και λειτουργικότητα παρέχουν στους κατασκευαστές του συστήματος. Για αυτό, η επιλογή των κατάλληλων εργαλείων για ανάπτυξη ενός λογισμικού συστήματος, συμβάλλει με καθοριστικό τρόπο στον βαθμό που θα επιτευχθούν οι στόχοι που θέτονται.

Πριν την υλοποίηση του συστήματος, μελετήθηκαν προσεκτικά τα διάφορα εργαλεία ανάπτυξης λογισμικών, τα οποία ήταν δυνατό να ληφθούν δωρεάν (open-source) είτε μέσω της δυνατότητας που έχουμε ως φοιτητές στα προγράμματα της Microsoft, για την επιλογή των πιο λειτουργικών, εύχρηστων, αλλά και κατάλληλων εργαλείων για την υλοποίηση του συστήματος.

Η έρευνα κατέληξε στην χρησιμοποίηση της γλώσσας προγραμματισμού Microsoft C-Sharp (C#) .NET, ως εργαλείου δημιουργίας διεπιφανειών χρήστη, αφού παρέχει το κατάλληλο γραφικό περιβάλλον για τον χρήστη του συστήματος, σύμφωνα με τις απαιτήσεις που καθορίστηκαν σε προηγούμενη φάση της ανάπτυξης του συστήματος. Το συγκεκριμένο περιβάλλον διευκολύνει σε μεγάλο βαθμό τον χρήστη, παρέχοντας του φόρμες οθόνης, χρήση ποντικιού, εμφάνιση βοήθειας και εικόνων. Για τον προγραμματισμό της βάσης δεδομένων του συστήματος επιλέχθηκε η χρήση του Microsoft SQL Server® 2008 [2]. Επιπλέον, για την δημιουργία αναφορών των διάφορων οντοτήτων του συστήματος χρησιμοποιήθηκε το εργαλείο SAP Crystal Reports.

Με την χρησιμοποίηση των πιο πάνω εργαλείων, οι στόχοι του ιδιοκτήτη της κτηνοτροφικής μονάδας έχουν επιτευχθεί, και το παρόν λογισμικό θα μπορεί να συντηρηθεί και να μετατραπεί εύκολα, αν στο μέλλον προκύψουν επιπλέον ή διαφορετικές απαιτήσεις, με σκοπό να ικανοποιούνται πάντοτε οι ανάγκες του ιδιοκτήτη.

Στις παραγράφους που ακολουθούν, δίνεται η περιγραφή των εργαλείων ανάπτυξης του συστήματος, γίνονται αναφορές στις δυσκολίες ανάπτυξης του πηγαίου

κώδικα για συγκεκριμένες διεργασίες και οι τρόποι με τους οποίους αντιμετωπίστηκαν οι δυσκολίες αυτές, όπως και κομμάτια του πηγαίου κώδικα τα οποία θεωρούνται άξια αναφοράς συνοδευόμενα με εκτενή περιγραφή. Επίσης παρουσιάζονται και δεδομένα ελέγχου για κάθε λειτουργία του συστήματος τα οποία χρησιμοποιήθηκαν για να ελεγχθεί η ορθότητα των λειτουργιών του συστήματος.

5.3.1 Microsoft C - Sharp .NET

Η όλη ανάπτυξη του συστήματος βασίστηκε στην συγκεκριμένη γλώσσα προγραμματισμού. Η C-sharp .NET, γνωστή και ως C-sharp, είναι μία από τις γλώσσες προγραμματισμού τρίτης γενεάς οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης από την Microsoft για το μοντέλο προγραμματισμού COM (Component Object Model). Η c-sharp θεωρείται επίσης μία σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και χρησιμοποίηση της λόγω των χαρακτηριστικών της, καθώς συνοδεύεται από ένα γραφικό περιβάλλον για τον χρήστη. Ανάμεσα σε άλλα η C- Sharp επιτρέπει:

- a) Την ταχεία ανάπτυξη εφαρμογών (RAD – Rapid Application Development) με γραφικό περιβάλλον χρήστη (GUI – Graphical User Interface).
- b) Πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects ή ActiveX Data Objects)
- c) Και την δημιουργία στοιχείων ελέγχου ActiveX

Προγράμματα γραμμένα σε C-Sharp μπορούν επίσης, να χρησιμοποιήσουν το Windows API, αλλά κάτι τέτοιο απαιτεί δηλώσεις εξωτερικών συναρτήσεων.

Επιπλέον, με την C-Sharp είναι δυνατή η λειτουργία εκτελέσιμων (.exe) αρχείων, στοιχείων ελέγχου ActiveX ή αρχείων DLL, αλλά χρησιμοποιείται κυρίως για την ανάπτυξη εφαρμογών για τα Windows και τη διασύνδεση συστημάτων βάσεων δεδομένων. Πλαίσια διαλόγου με λιγότερες λειτουργίες μπορούν να χρησιμοποιηθούν για pop-up δυνατότητες. Τα στοιχεία ελέγχου παρέχουν τις βασικές λειτουργίες της εφαρμογής, ενώ οι προγραμματιστές μπορούν να εισάγουν επιπλέον λογική μέσα στο κατάλληλο χειριστή γεγονότων. Για παράδειγμα, ένα πτυσσόμενο πλαίσιο θα εμφανίζει αυτόματα μία λίστα που θα επιτρέπει στο χρήστη να επιλέξει οποιοδήποτε στοιχείο. Ένας χειριστής γεγονότων καλείται όταν ένα αντικείμενο είναι επιλεγμένο και στην συνέχεια μπορεί να εκτελεστεί πρόσθετος κώδικας που δημιουργείται από τον

προγραμματιστή για να εκτελεστεί κάποια ενέργεια που βασίζεται στο στοιχείο που έχει επιλεγεί.

5.3.2 Microsoft SQL Server®2008

Ο Microsoft SQL Server είναι μία σχεσιακή βάση δεδομένων, η οποία αναπτύχθηκε από την Microsoft. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μία βάση δεδομένων, η οποία αποτελείται από μία συλλογή πινάκων και κώδικα. Η κεντρική βάση δεδομένων του SQL υποστηρίζει διάφορους τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, varchar(σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών(για τα μη-δομημένα δεδομένα), κειμένων (για κείμενα). Η επερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση των στοιχείων από μία βάση δεδομένων. Η επερώτηση εκφράζεται χρησιμοποιώντας μία παραλλαγή της T-SQL, η οποία είναι μία διάλεκτος SQL που αναπτύχθηκε από την Microsoft και την Sybase. Η επερώτηση διευκρινίζει επεξηγηματικά αυτό που είναι να ανακτηθεί από την βάση δεδομένων [1].

Ο SQL Server επιτρέπει επίσης αποθηκευμένες επερωτήσεις σε μορφή διαδικασιών (stored procedures). Οι αποθηκευμένες διαδικασίες είναι επερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server. Συνήθως αποτελούνται από 100 μέχρι 1000 γραμμές κώδικα SQL. Μπορούν να δεχτούν τιμές που στέλλονται από τον πελάτη (client) ως παράμετροι εισαγωγής (παράμετροι εισόδου), και να στείλουν τα αποτελέσματα ως παραμέτρους παραγωγής (παράμετροι εξόδου). Μπορούν να καλέσουν συναρτήσεις (functions) και άλλες αποθηκευμένες διαδικασίες (stored procedures).

Ένα από τα σημαντικά στοιχεία του SQL Server είναι η ύπαρξη ενός ξεχωριστού είδους αποθηκευμένων διαδικασιών που είναι όμως σε μορφή διαδικασιών, και λέγονται σκανδάλες (triggers). Η σκανδάλη είναι αντικείμενο της βάσης το οποίο είναι προσαρτημένο σε ένα πίνακα. Αυτό που τις κάνει να διαφέρουν από τις αποθηκευμένες διαδικασίες είναι το γεγονός ότι η σκανδάλη αναφέρεται μόνο σε ένα πίνακα της βάσης δεδομένων και ενεργοποιείται μόνο αν γίνει εισαγωγή, ενημέρωση ή διαγραφή στον συγκεκριμένο πίνακα της βάσης δεδομένων.

5.3.3 SAP Crystal Reports

Το λογισμικό της SAP Crystal Reports επιτρέπει στους χρήστες της να σχεδιάζουν εύκολα διαδραστικές αναφορές και να τις συνδέουν σε οποιαδήποτε νοητή πηγή δεδομένων. Οι χρήστες του μπορούν να επωφεληθούν από την ταξινόμηση και το φιλτράρισμα που παρέχουν οι αναφορές, δίνοντας του έτσι την δυνατότητα να εκτελέσουν άμεσα τις αποφάσεις τους.

Επιπλέον, με το SAP Crystal Reports Dashboard Design package, μπορεί κανείς να κερδίσει ακόμη περισσότερη λειτουργικότητα για επιτακτικές αναφορές. Το προϊόν αυτό δίνει την δυνατότητα να δημιουργήσει κανείς εξαιρετικά διαμορφωμένες αναφορές με what-if μοντέλα σεναρίων και διαδραστικά γραφήματα, και τα παραδίνει μέσω του διαδικτύου, e-mail, Microsoft Office, Adobe PDF, ή με ενσωματωμένες επιχειρησιακές εφαρμογές. Επακόλουθα, οι αναφορές μπορούν να αξιοποιηθούν για να φτιάξουν καλύτερες λειτουργικές και στρατηγικές αποφάσεις.

Με τα SAP Crystal Reports μπορεί κανείς:

- a) Να αξιοποιήσει επαγγελματικές εκθέσεις- σε τιμές για όλους
- b) Να ενισχύσει τους τελικούς χρήστες στο να εξερευνήσουν τις αναφορές με ταξινόμηση των αναφορών, αλλά και με χρήση παραμέτρων
- c) Να ελαχιστοποιήσει την προσπάθεια των IT αλλά και των προγραμματιστών με την χρήση διαδραστικών αναφορών
- d) Να αναπτύξει ισχυρούς συνδυασμούς δεδομένων
- e) Να εξοικονομήσει πολύτιμο χρόνο που απαιτείται για την σχεδίαση της αναφοράς
- f) Να ενσωματώσει αναφορές με επαγγελματική εμφάνιση σε Java και .NET εφαρμογές
- g) Να προσαρμόσει την λύση του προσθέτοντας εργαλεία διαχείρισης και προβολής των αναφορών

5.3.4 Περιγραφή βάσης δεδομένων

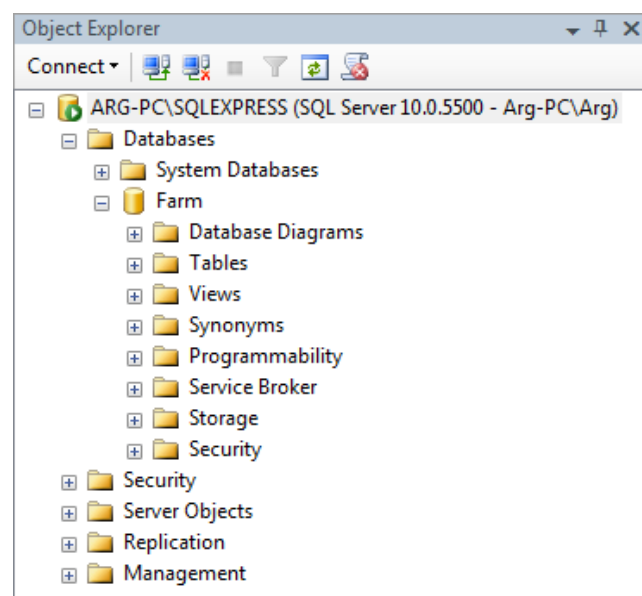
Η βάση δεδομένων του συστήματος δημιουργείται, όπως έχει αναφερθεί και πιο πάνω από τον Microsoft SQL Server. Πρόσβαση στην βάση μπορεί να έχει μόνο ο διαχειριστής της βάσης. Ανοίγοντας τον Server, εμφανίζεται μία οθόνη, στην οποία ζητείται το όνομα του χρήστη, και ο κωδικός πρόσβασης, τα οποία θα του επιτρέψουν

την εισαγωγή στην βάση. Στην εικόνα που ακολουθεί παρουσιάζεται η φόρμα εισαγωγής του χρήστη στο SQL Management Studio:



Εικόνα 5.1

Αν τα πιο πάνω στοιχεία εισαχθούν επιτυχώς, τότε το SQL Management Studio παρουσιάζει στα αριστερά της οθόνης κάτω από τον Object Explorer, την βάση του συστήματος σε δένδροειδές διάγραμμα το οποίο περιέχει τους πίνακες της βάσης δεδομένων του συστήματος. Το δένδροειδές διάγραμμα της βάσης δεδομένων παρουσιάζεται στην αμέσως επόμενη εικόνα:



Εικόνα 5.2

Στην βάση δεδομένων του συστήματος έχουν δημιουργηθεί διάφορες αποθηκευμένες επερωτήσεις σε μορφή διαδικασιών και όψεων τα οποία συμβάλλουν για την καλύτερη και ευκολότερη λειτουργία του συστήματος.

Για κάθε πίνακα της βάσης ο οποίος αφορά μία συγκεκριμένη οντότητα του συστήματος, έχουν δημιουργηθεί αποθηκευμένες διαδικασίες (stored procedures), οι οποίες είναι υπεύθυνες για την εισαγωγή, την ενημέρωση, την διαγραφή, και την αναζήτηση μίας ή περισσότερων εγγραφών του αντίστοιχου πίνακα.

Ανάμεσα στις αποθηκευμένες διαδικασίες που έχουν δημιουργηθεί είναι και αυτές που υπολογίζουν τους παράγοντες της εξίσωσης που χρησιμοποιήθηκε για την διαδικασία απόρριψης των προβάτων. Για παράδειγμα, υπολογίστηκαν τα κτηνιατρικά έξοδα της φάρμας για μία συγκεκριμένη χρονική περίοδο, η παραγωγικότητα ενός προβάτου όσο αφορά τα πρόβατα, καθώς επίσης και το κόστος που έχουν στην κατανάλωση τροφής, και τέλος, η κρεοπαραγωγή του προβάτου, δηλαδή η αναλογία του αριθμού των ζώων που επέζησαν προς τον αριθμό των ζώων που γεννήθηκαν συνολικά από ένα συγκεκριμένο πρόβατο.

Επιπλέον, έχουν δημιουργηθεί οι όψεις (views) του κάθε πίνακα. Η κάθε όψη χρησιμοποιείται όταν δημιουργείται η αναφορά για κάθε οντότητα, έτσι ώστε ο χρήστης να βλέπει και να τυπώνει τα πεδία του πίνακα που πραγματικά έχουν σημασία και νόημα.

Η περιγραφή των πινάκων της βάσης δεδομένων έγινε στην παράγραφο 2.3 της φάσης καθορισμού των προδιαγραφών του συστήματος.

5.3.5 Δύσκολα σημεία στον κώδικα

5.3.5.1 Εισαγωγή και επιλογή δεδομένων από το εργαλείο Combo box της εργαλειοθήκης του Microsoft Visual Studio 2010.

Κατά την ανάπτυξη του λογισμικού του συστήματος, εισηγήθηκε η εισαγωγή δεδομένων με χρήση του εργαλείου Combo box της εργαλειοθήκης του Visual Studio, καθώς είναι ένας αποτελεσματικός και εύχρηστος τρόπος για τον χρήστη στην επιλογή δεδομένων. Στην πορεία όμως, παρατηρήθηκε ότι όταν η εισαγωγή δεδομένων στο Combo box γινόταν με την χρήση κάποιας επερώτησης, δηλαδή με στοιχεία από την βάση δεδομένων, τότε η επιλογή δεδομένων από το εργαλείο αυτό δεν ήταν δυνατή, καθώς η επιλογή του χρήστη ακυρωνόταν, κα επιλεγόταν πάντα η πρώτη

τιμή του Combo box. Μετά από έρευνα, παρατηρήθηκε ότι δεν υπάρχει κάποιος τρόπος επιδιόρθωσης του προβλήματος αυτού μέσω του Visual Studio. Για τον λόγο αυτό, έβαλα τα στοιχεία στο Combo box προγραμματιστικά. Δηλαδή, εκτέλεσα την επερώτηση (query), και κάθε τιμή που επέστρεφε την πρόσθεσα στην συλλογή των αντικειμένων του Combo box.

5.3.5.2 Μετατροπή του τύπου μίας εικόνας από image σε varbinary(max)

Μία ανάγκη του συστήματος ήταν η μετατροπή του τύπου της εικόνας από image σε varbinary [8]. Αυτό έπρεπε να γίνει, γιατί σύμφωνα με ανακοίνωση της Microsoft, ο τύπος δεδομένων image θα αφαιρεθεί από μελλοντικές εκδόσεις του Microsoft SQL Server. Έτσι προέκυψε η ανάγκη μετατροπής από image σε varbinary, ο οποίος συστήθηκε από τους διαχειριστές της Microsoft, προς αντικατάσταση του τύπου image.

5.3.5.3 Εξαγωγή δεδομένων του πίνακα της βάσης σε αρχείο Excel

Όπως έχει αναφερθεί σε προηγούμενες παραγράφους, δύο από τους πίνακες που αποτελούν την βάση δεδομένων, είναι ο πίνακας για την καταχώρηση των στοιχείων της ημερήσιας παραγωγής γάλακτος των προβάτων, καθώς επίσης και ο πίνακας καταχώρησης των στοιχείων της ημερήσιας κατανάλωσης τροφής των προβάτων. Ως εκ τούτου, το μέγεθος των πινάκων αυτών θα ήταν πάρα πολύ μεγάλο. Για τον λόγο αυτό, στο τέλος κάθε μήνα δημιουργείται ένα excel αρχείο [5], στο οποίο εξάγονται τα στοιχεία του αντίστοιχου πίνακα τα οποία αφορούσαν τον περασμένο μήνα. Αφού ολοκληρωθεί η διαδικασία αυτή, τότε τα στοιχεία στους 2 πίνακες διαγράφονται, έτσι ώστε κάθε φορά ο πίνακας να έχει τα στοιχεία του τρέχοντος μήνα.

5.3.5.4 Εισαγωγή δεδομένων με γραφικό τρόπο

Όταν γίνεται η καταχώρηση των στοιχείων της ημερήσιας παραγωγής γάλακτος, και κατανάλωσης τροφής από την φόρμα, τότε τα ένα ζώο πρέπει να έχει ήδη ομαδοποιηθεί για να μπορέσει να γίνει η καταχώρησης της παραγωγής γάλακτος και κατανάλωσης τροφής του. Όταν η πιο πάνω διαδικασία ολοκληρωθεί τότε, παρουσιάζονται στον χρήστη οι ‘κυψέλες’ μέσα στις οποίες μπαίνουν τα ζώα για την κατανάλωση τροφής τους, που είναι δηλαδή η θέση τους στην φάρμα, καθώς

επίσης και ‘κυψέλες’ που συμβολίζουν την θέση που παίρνει το πρόβατο μέσα στο γαλευτήρι. Η καταχώρηση τροφής, γίνεται για όλα τα πρόβατα που ανήκουν σε κάθε ‘κυψέλη’, ενώ η καταχώρηση παραγωγής γάλακτος γίνεται για κάθε ζώο ξεχωριστά.

5.3.5.5 Δημιουργία Crystal Report – αναφορά που να δέχεται παράμετρο δεδομένων

Κατά την δημιουργία αναφορών σχετικά με την παραγωγή γάλακτος και την κατανάλωση τροφής των προβάτων, παρουσιάστηκε η ανάγκη δημιουργίας αναφορών για κάθε πρόβατο ξεχωριστά, με σκοπό ο χρήστης του συστήματος να παρακολουθεί την πορεία της γαλακτοπαραγωγής και κατανάλωσης τροφής του κάθε προβάτου και έτσι να καταλάβει αν υπάρχει κάποιο πρόβλημα με το κάθε πρόβατο της φάρμας. Επομένως, έπρεπε να δημιουργηθεί η κατάλληλη αποθηκευμένη διαδικασία(stored procedure) η οποία θα χρησιμοποιείτο για την προσθήκη στοιχείων στην αναφορά, και κάθε φορά που θα εκτελείτο η διαδικασία αυτή, θα έπαιρνε τιμές στις παραμέτρους εισόδου της που θα προέρχονταν από το Crystal Report.

5.3.5.6 Εκτύπωση αποδείξεων και τιμολογίων

Μία από τις απαιτήσεις του συστήματος, ήταν η εκτύπωση τιμολογίων και αποδείξεων με τυποποιημένη μορφή. Ζητήθηκε από τους χρήστες του συστήματος να υπάρχει η δυνατότητα επέμβασης στο εκτυπώσιμο έγγραφο για τυχόν τροποποιήσεις και αποθήκευσης του. Τα εργαλεία που προσφέρει η C-Sharp, δεν ικανοποιούν τις ανάγκες αυτές, αφού εμφανίζεται μόνο μία προεπισκόπηση του εκτυπώσιμου, στην οποία δεν επιτρέπεται η τροποποίηση ή η αποθήκευση. Μία από τις λύσεις που ήταν προσιτές ήταν η χρήση της Microsoft Office, και συγκεκριμένα του Microsoft Word τα οποία βρίσκονται κάτω από το namespace Microsoft.Office.Interop της C-Sharp. Τα έγγραφα τυποποιημένης μορφής (templates) και οι σελιδοδείκτες (bookmarks) που προσφέρει η Word, βοήθησαν στην δημιουργία ενός ικανοποιητικού και εμφανίσιμου αποτελέσματος. Αφού δημιουργήθηκε η όλη εμφάνιση του εγγράφου, έπρεπε να βρεθεί τρόπος να τοποθετούνται όλες οι πληροφορίες στο έγγραφο [6]. Έτσι προστέθηκαν σελιδοδείκτες στο έγγραφο και μέσω της C-Sharp τοποθετούνται οι πληροφορίες στο σωστό σημείο του εγγράφου.

5.3.6 Πηγαίος κώδικας με σχόλια

5.3.6.1 Διαχείριση Ιδιοκτητών

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου ιδιοκτήτη της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand ();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter ();

Exception exc = new Exception();

    try
    {

        adapter.SelectCommand = new System.Data.SqlClient.SqlCommand ();

        cmd.CommandText = "sp_insert_owner";
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Connection = Module.connection;

        cmd.Parameters.Clear();
        cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
        cmd.Parameters.Add("@ID", SqlDbType.NVarChar).Value = id;
        cmd.Parameters.Add("@Fname", SqlDbType.NVarChar).Value = name;
        cmd.Parameters.Add("@Lname", SqlDbType.NVarChar).Value = surname;
        cmd.Parameters.Add("@Sex", SqlDbType.Char).Value = sex;
        cmd.Parameters.Add("@BirthDate", SqlDbType.SmallDateTime).Value =
birthdate;
        cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
        cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value =
street;
        cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value =
postal_code;
        cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
        cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value =
country;
        cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
        cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
        cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
        cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
        cmd.Parameters.Add("@username", SqlDbType.NVarChar).Value =
username;
        cmd.Parameters.Add("@password", SqlDbType.NVarChar).Value =
password;

        if (cmd.Connection.State != ConnectionState.Open)
        {
            cmd.Connection.Open();
        }
        cmd.ExecuteNonQuery();
```

```

        cmd.Connection.Close();
        ManageAppointmentResources.insert_resource(name, surname, '0');
    }
    catch
    {
        MessageBox.Show("not connected " + exc.Message);
    }
}

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand ();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter ();

Exception exc = new Exception();

    try
    {

        adapter.SelectCommand = new System.Data.SqlClient.SqlCommand ();

        cmd.CommandText = "sp_update_owner";
        cmd.CommandType = CommandType.StoredProcedure;
        cmd.Connection = Module.connection;

        cmd.Parameters.Clear();
        cmd.Parameters.Add("@owner_id", SqlDbType.Int).Value =
Module.owner_id;
        cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
        cmd.Parameters.Add("@ID", SqlDbType.NVarChar).Value = id;
        cmd.Parameters.Add("@Fname", SqlDbType.NVarChar).Value = name;
        cmd.Parameters.Add("@Lname", SqlDbType.NVarChar).Value = surname;
        cmd.Parameters.Add("@Sex", SqlDbType.Char).Value = sex;
        cmd.Parameters.Add("@BirthDate", SqlDbType.SmallDateTime).Value =
birthdate;
        cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
        cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value =
street;
        cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value =
postal_code;
        cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
        cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value =
country;
        cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
        cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
        cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
        cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
        cmd.Parameters.Add("@username", SqlDbType.NVarChar).Value =
username;
        cmd.Parameters.Add("@password", SqlDbType.NVarChar).Value =
password;

        if (cmd.Connection.State != ConnectionState.Open)
        {
            cmd.Connection.Open();
        }
    }
}

```

```

        cmd.ExecuteNonQuery();
        cmd.Connection.Close();
        ManageAppointmentResources.insert_resource(name, surname, '0');
    }
    catch
    {
        MessageBox.Show("not connected " + exc.Message);
    }

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_search_owner";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@surname", SqlDbType.NVarChar).Value = surname;
    cmd.Parameters.Add("@id", SqlDbType.NVarChar).Value = id;
    cmd.Parameters.Add("@sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@town", SqlDbType.NVarChar).Value = town;
    cmd.Parameters.Add("@phone", SqlDbType.NVarChar).Value = phone;
    cmd.Parameters.Add("@mobile", SqlDbType.NVarChar).Value = mobile;
    cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value = email;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_owner");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }

    cmd.Connection.Close();

    dgvOwner.DataSource = dataset;
    dgvOwner.DataMember = "search_owner";

    int counter = 0;
    int rows = dgvOwner.RowCount;

    this.dgvOwner.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvOwner.DefaultCellStyle.ForeColor = Color.Black;
}

```

```

        this.dgvOwner.DefaultCellStyle.SelectionForeColor = Color.White;
        this.dgvOwner.DefaultCellStyle.SelectionBackColor = Color.Black;

        this.dgvOwner.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
        this.dgvOwner.BackgroundColor = Color.LightSteelBlue;
    }
    catch
    {
        MessageBox.Show("Exception :" + exc.Message);
    }

```

Κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
Try
{
    cmd.CommandText = "sp_delete_owner";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@owner_id", SqlDbType.Int).Value = Module.owner_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
    ManageAppointmentResources.delete_resource(Module.owner_id, '0');
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.2 Διαχείριση Υπαλλήλων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου υπαλλήλου της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    cmd.CommandText = "sp_insert_employee";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

```

```

cmd.Parameters.Clear();
cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
cmd.Parameters.Add("@ID", SqlDbType.NVarChar).Value = id;
cmd.Parameters.Add("@Fname", SqlDbType.NVarChar).Value = name;
cmd.Parameters.Add("@Lname", SqlDbType.NVarChar).Value = surname;
cmd.Parameters.Add("@Sex", SqlDbType.Char).Value = sex;
cmd.Parameters.Add("@BirthDate", SqlDbType.SmallDateTime).Value = birthdate;
cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value = street;
cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value = postal_code;
cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value = country;
cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
cmd.Parameters.Add("@username", SqlDbType.NVarChar).Value = username;
cmd.Parameters.Add("@password", SqlDbType.NVarChar).Value = password;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}

cmd.ExecuteNonQuery();
cmd.Connection.Close();
ManageAppointmentResources.insert_resource(name, surname, '1');
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    cmd.CommandText = "sp_update_employee";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@employee_id", SqlDbType.Int).Value = Module.employee_id;
    cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@ID", SqlDbType.NVarChar).Value = id;
    cmd.Parameters.Add("@Fname", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@Lname", SqlDbType.NVarChar).Value = surname;
    cmd.Parameters.Add("@Sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@BirthDate", SqlDbType.SmallDateTime).Value = birthdate;
    cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
    cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value = street;
    cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value = postal_code;
}

```

```

cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value = country;
cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
cmd.Parameters.Add("@username", SqlDbType.NVarChar).Value = username;
cmd.Parameters.Add("@password", SqlDbType.NVarChar).Value = password;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
ManageAppointmentResources.update_resource(Module.employee_id, name, surname,
'1');
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_search_employee";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@surname", SqlDbType.NVarChar).Value = surname;
    cmd.Parameters.Add("@id", SqlDbType.NVarChar).Value = id;
    cmd.Parameters.Add("@sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@town", SqlDbType.NVarChar).Value = town;
    cmd.Parameters.Add("@phone", SqlDbType.NVarChar).Value = phone;
    cmd.Parameters.Add("@mobile", SqlDbType.NVarChar).Value = mobile;
    cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value = email;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_employee");
    }
    catch (Exception excep)
    {

```

```

        MessageBox.Show(excep.Message);
    }

    cmd.Connection.Close();
    dgvEmployee.DataSource = dataset;
    dgvEmployee.DataMember = "search_employee";

    int counter = 0;
    int rows = dgvEmployee.RowCount;

    this.dgvEmployee.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvEmployee.DefaultCellStyle.ForeColor = Color.Black;

    this.dgvEmployee.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvEmployee.DefaultCellStyle.SelectionBackColor = Color.Black;

    this.dgvEmployee.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvEmployee.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}

```

Κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_delete_employee";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@employee_id", SqlDbType.Int).Value = Module.employee_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();

    ManageAppointmentResources.delete_resource(Module.employee_id, '1');
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.3 Διαχείριση Φαρμών

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιας από τις φάρμες της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    cmd.CommandText = "sp_insert_farm";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
    cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value = street;
    cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value = postal_code;
    cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
    cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value = country;
    cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
    cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
    cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
    cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
    cmd.Parameters.Add("@gov_licence", SqlDbType.NVarChar).Value = gov_license;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

Κώδικας ενημέρωσης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    cmd.CommandText = "sp_update_farm";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = Module.farm_id;
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
    cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value = street;
```

```

cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value = postal_code;
cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value = country;
cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
cmd.Parameters.Add("@gov_licence", SqlDbType.NVarChar).Value = gov_license;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_search_farm";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@town", SqlDbType.NVarChar).Value = town;
    cmd.Parameters.Add("@country", SqlDbType.NVarChar).Value = country;
    cmd.Parameters.Add("@phone", SqlDbType.NVarChar).Value = phone;
    cmd.Parameters.Add("@mobile", SqlDbType.NVarChar).Value = mobile;
    cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value = email;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_farm");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }
    cmd.Connection.Close();

    dgvFarm.DataSource = dataset;
}

```

```

dgvFarm.DataMember = "search_farm";

int counter = 0;
int rows = dgvFarm.RowCount;

this.dgvFarm.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
this.dgvFarm.DefaultCellStyle.ForeColor = Color.Black;

this.dgvFarm.DefaultCellStyle.SelectionForeColor = Color.White;
this.dgvFarm.DefaultCellStyle.SelectionBackColor = Color.Black;

this.dgvFarm.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
this.dgvFarm.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" +exc.Message);
}

```

Κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_delete_farm";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = Module.farm_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.4 Διαχείριση Προβάτων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου προβάτου της κτηνοτροφικής μονάδας. Επιπλέον, κατά την διαδικασία αυτή είναι δυνατή η εισαγωγή θηλυκών προβάτων, καθώς και η εισαγωγή, ενημέρωση, αναζήτηση και διαγραφή γεννών των προβάτων.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    byte[] imageData;
    if (txtImage.Text != "")
    {
        imageData = ReadFile(txtImage.Text);
    }
    else
    {
        imageData = ReadFile(Module.picture_source);
        image = Module.picture_source;
    }

    cmd.CommandText = "sp_insert_sheep";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@animal_num", SqlDbType.Int).Value=num;
    cmd.Parameters.Add("@Name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@Birthdate", SqlDbType.SmallDateTime).Value = birthdate;
    cmd.Parameters.Add("@Sex", SqlDbType.NChar).Value = sex;
    cmd.Parameters.Add("@Genotype", SqlDbType.NChar).Value = genotype;
    cmd.Parameters.Add("@WeightInKilos", SqlDbType.Decimal).Value = weight;
    cmd.Parameters.Add("@WeightInKilos2", SqlDbType.Decimal).Value = weight2;
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
    cmd.Parameters.Add("@food_consumption", SqlDbType.Decimal).Value = food_cons;
    cmd.Parameters.Add("@food_cost", SqlDbType.Decimal).Value = food_cost;
    cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@sheep_img", SqlDbType.NVarChar).Value = image;
    cmd.Parameters.Add("@mother_id", SqlDbType.Int).Value = mother_id;
    cmd.Parameters.Add("@father_id", SqlDbType.Int).Value = father_id;
    cmd.Parameters.Add("@milk_prod", SqlDbType.Decimal).Value = milk_prod;
    cmd.Parameters.Add("@milk_cost", SqlDbType.Decimal).Value = milk_cost;
    cmd.Parameters.Add("@numBirths", SqlDbType.Int).Value = numBirts;
    cmd.Parameters.Add("@team", SqlDbType.Char).Value = '-';
    cmd.Parameters.Add("@src_img", SqlDbType.VarBinary).Value = imageData;
    cmd.Parameters.Add("@kind", SqlDbType.NVarChar).Value = kind;
    cmd.Parameters.Add("@food_count", SqlDbType.Int).Value = food_count;
    cmd.Parameters.Add("@milk_count", SqlDbType.Int).Value = milk_count;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

```
}
```

Κώδικας ενημέρωσης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    byte[] imageData;

    if (txtImage.Text != "")
    {
        imageData = ReadFile(txtImage.Text);
    }
    else
    {
        imageData = ReadFile(Module.picture_source);
        image = Module.picture_source;
    }

    cmd.CommandText = "sp_update_sheep";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = Module.sheep_id;
    cmd.Parameters.Add("@animal_num", SqlDbType.Int).Value = num;
    cmd.Parameters.Add("@Name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@Birthdate", SqlDbType.SmallDateTime).Value = birthdate;
    cmd.Parameters.Add("@Sex", SqlDbType.NChar).Value = sex;
    cmd.Parameters.Add("@Genotype", SqlDbType.NChar).Value = genotype;
    cmd.Parameters.Add("@WeightInKilos", SqlDbType.Decimal).Value = weight;
    cmd.Parameters.Add("@WeightInKilos2", SqlDbType.Decimal).Value = weight2;
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
    cmd.Parameters.Add("@food_consumption", SqlDbType.Decimal).Value = food_cons;
    cmd.Parameters.Add("@food_cost", SqlDbType.Decimal).Value = food_cost;
    cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@sheep_img", SqlDbType.NVarChar).Value = image;
    cmd.Parameters.Add("@mother_id", SqlDbType.Int).Value = mother_id;
    cmd.Parameters.Add("@father_id", SqlDbType.Int).Value = father_id;
    cmd.Parameters.Add("@milk_prod", SqlDbType.Decimal).Value = milk_prod;
    cmd.Parameters.Add("@milk_cost", SqlDbType.Decimal).Value = milk_cost;
    cmd.Parameters.Add("@numBirths", SqlDbType.Int).Value = numBirts;
    cmd.Parameters.Add("@src_img", SqlDbType.VarBinary).Value = imageData;
    cmd.Parameters.Add("@kind", SqlDbType.NVarChar).Value = kind;
    cmd.Parameters.Add("@food_count", SqlDbType.Int).Value = food_count;
    cmd.Parameters.Add("@milk_count", SqlDbType.Int).Value = milk_count;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

```
}
```

Κώδικας αναζήτησης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_search_sheep";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@animal_num", SqlDbType.Int).Value = num;
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@birthdate", SqlDbType.SmallDateTime).Value = birthdate;
    cmd.Parameters.Add("@sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@genotype", SqlDbType.NVarChar).Value = genotype;
    cmd.Parameters.Add("@weight", SqlDbType.Decimal).Value = weight;
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
    cmd.Parameters.Add("@kind", SqlDbType.NVarChar).Value = kind;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_farm");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }
    cmd.Connection.Close();

    dgvSheep.DataSource = dataset;
    dgvSheep.DataMember = "search_farm";

    int counter = 0;
    int rows = dgvSheep.RowCount;

    this.dgvSheep.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvSheep.DefaultCellStyle.ForeColor = Color.Black;

    this.dgvSheep.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvSheep.DefaultCellStyle.SelectionBackColor = Color.Black;

    this.dgvSheep.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvSheep.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}
```

Κώδικας διαγραφής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_delete_sheep";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = Module.sheep_id;
    cmd.Parameters.Add("@type", SqlDbType.Char).Value = delete_cond;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}
```

5.3.6.5 Διαχείριση Κτηνιάτρων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου κτηνιάτρου της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    cmd.CommandText = "sp_insert_veterinarian";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@ID", SqlDbType.NVarChar).Value = id;
    cmd.Parameters.Add("@Fname", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@Lname", SqlDbType.NVarChar).Value = surname;
    cmd.Parameters.Add("@Sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@BirthDate", SqlDbType.SmallDateTime).Value = birthdate;
    cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
    cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value = street;
    cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value = postal_code;
```

```

        cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
        cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value = country;
        cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
        cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
        cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
        cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
        cmd.Parameters.Add("@OfficeAddressNum", SqlDbType.NVarChar).Value = off_num;
        cmd.Parameters.Add("@OfficeAddressStreet", SqlDbType.NVarChar).Value =
off_street;
        cmd.Parameters.Add("@OfficePostalCode", SqlDbType.NVarChar).Value =
off_postalcode;
        cmd.Parameters.Add("@OfficeAddressTown", SqlDbType.NVarChar).Value =
off_town;
        cmd.Parameters.Add("@OfficeAddressCountry", SqlDbType.NVarChar).Value =
off_country;

        if (cmd.Connection.State != ConnectionState.Open)
        {
            cmd.Connection.Open();
        }
        cmd.ExecuteNonQuery();
        cmd.Connection.Close();

        ManageAppointmentResources.insert_resource(name, surname, '2');
    }
    catch
    {
        MessageBox.Show("not connected " + exc.Message);
    }
}

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    cmd.CommandText = "sp_update_veterinarian";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@vet_id", SqlDbType.Int).Value = Module.vet_id;
    cmd.Parameters.Add("@Condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@ID", SqlDbType.NVarChar).Value = id;
    cmd.Parameters.Add("@Fname", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@Lname", SqlDbType.NVarChar).Value = surname;
    cmd.Parameters.Add("@Sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@BirthDate", SqlDbType.SmallDateTime).Value = birthdate;
    cmd.Parameters.Add("@AddressNum", SqlDbType.NVarChar).Value = num;
    cmd.Parameters.Add("@AddressStreet", SqlDbType.NVarChar).Value = street;
    cmd.Parameters.Add("@PostalCode", SqlDbType.NVarChar).Value = postal_code;
    cmd.Parameters.Add("@AddressTown", SqlDbType.NVarChar).Value = town;
    cmd.Parameters.Add("@AddressCountry", SqlDbType.NVarChar).Value = country;
    cmd.Parameters.Add("@Phone", SqlDbType.NVarChar).Value = phone;
    cmd.Parameters.Add("@Mobile", SqlDbType.NVarChar).Value = mobile;
    cmd.Parameters.Add("@Fax", SqlDbType.NVarChar).Value = fax;
}

```

```

        cmd.Parameters.Add("@Email", SqlDbType.NVarChar).Value = email;
        cmd.Parameters.Add("@OfficeAddressNum", SqlDbType.NVarChar).Value = off_num;
        cmd.Parameters.Add("@OfficeAddressStreet", SqlDbType.NVarChar).Value =
off_street;
        cmd.Parameters.Add("@OfficePostalCode", SqlDbType.NVarChar).Value =
off_postalcode;
        cmd.Parameters.Add("@OfficeAddressTown", SqlDbType.NVarChar).Value =
off_town;
        cmd.Parameters.Add("@OfficeAddressCountry", SqlDbType.NVarChar).Value =
off_country;

        if (cmd.Connection.State != ConnectionState.Open)
        {
            cmd.Connection.Open();
        }
        cmd.ExecuteNonQuery();
        cmd.Connection.Close();

        ManageAppointmentResources.insert_resource(name, surname, '2');
    }
    catch
    {
        MessageBox.Show("not connected " + exc.Message);
    }
}

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_search_veterinarian";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@surname", SqlDbType.NVarChar).Value = surname;
    cmd.Parameters.Add("@id", SqlDbType.NVarChar).Value = id;
    cmd.Parameters.Add("@sex", SqlDbType.Char).Value = sex;
    cmd.Parameters.Add("@town", SqlDbType.NVarChar).Value = town;
    cmd.Parameters.Add("@office_town", SqlDbType.NVarChar).Value=off_town;
    cmd.Parameters.Add("@phone", SqlDbType.NVarChar).Value = phone;
    cmd.Parameters.Add("@mobile", SqlDbType.NVarChar).Value = mobile;
    cmd.Parameters.Add("@email", SqlDbType.NVarChar).Value = email;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_vet");
    }
    catch (Exception excep)

```

```

    {
        MessageBox.Show(excep.Message);
    }

    cmd.Connection.Close();
    dgvVet.DataSource = dataset;
    dgvVet.DataMember = "search_vet";

    int counter = 0;
    int rows = dgvVet.RowCount;

    this.dgvVet.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvVet.DefaultCellStyle.ForeColor = Color.Black;

    this.dgvVet.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvVet.DefaultCellStyle.SelectionBackColor = Color.Black;

    this.dgvVet.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvVet.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}

```

Κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_delete_veterinarian";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@vet_id", SqlDbType.Int).Value = Module.vet_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
    ManageAppointmentResources.delete_resource(Module.vet_id, '2');
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.6 Διαχείριση Ασθενειών

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, και παρουσίαση των στοιχείων κάποιας ασθένειας οι οποία μπορεί να διαγνωσθεί σε κάποια από τα πρόβατα της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    cmd.CommandText = "sp_insert_illness";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@weight", SqlDbType.Int).Value = weight;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

Κώδικας ενημέρωσης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    cmd.CommandText = "sp_update_illness";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@illness_id", SqlDbType.Int).Value = Module.illness_id;
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@weight", SqlDbType.Int).Value = weight;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

Κώδικας αναζήτησης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_search_illness";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@weight", SqlDbType.Int).Value = weight;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_illness");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }

    cmd.Connection.Close();
    dgvIllness.DataSource = dataset;
    dgvIllness.DataMember = "search_illness";

    int counter = 0;
    int rows = dgvIllness.RowCount;

    this.dgvIllness.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvIllness.DefaultCellStyle.ForeColor = Color.Black;
    this.dgvIllness.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvIllness.DefaultCellStyle.SelectionBackColor = Color.Black;
    this.dgvIllness.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvIllness.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}
```

5.3.6.7 Διαχείριση Φαρμάκων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, και παρουσίαση των στοιχείων κάποιου φαρμάκου το οποίο μπορεί να χορηγηθεί σε κάποια από τα πρόβατα της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    cmd.CommandText = "sp_insert_medicine";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

Κώδικας ενημέρωσης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    cmd.CommandText = "sp_update_medicine";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@medicine_id", SqlDbType.Int).Value = Module.medicine_id;
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}
```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_search_medicine";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@name", SqlDbType.NVarChar).Value = name;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@price", SqlDbType.Real).Value = price;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_medicine");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }

    cmd.Connection.Close();
    dgvMedicine.DataSource = dataset;
    dgvMedicine.DataMember = "search_medicine";

    int counter = 0;
    int rows = dgvMedicine.RowCount;

    this.dgvMedicine.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvMedicine.DefaultCellStyle.ForeColor = Color.Black;
    this.dgvMedicine.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvMedicine.DefaultCellStyle.SelectionBackColor = Color.Black;
    this.dgvMedicine.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvMedicine.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}

```

5.3.6.8 Διαχείριση Κτηνιατρικών Εξετάσεων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιας κτηνιατρικής εξέτασης η οποία έγινε σε κάποιο από τα πρόβατα της κτηνοτροφικής μονάδας από κάποιο κτηνίατρο.

Κώδικας εισαγωγής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataReader reader;
Exception exc = new Exception();
try
{
    int x = 0;

    for (x = 0; x < lstIllness.Items.Count; x++)
    {
        if (lstIllness.GetSelected(x)==true)
        {
            illness_id.Add(illness[x].ValueMember);
        }
    }

    for (x = 0; x < lstMedicine.Items.Count; x++)
    {
        if (lstMedicine.GetSelected(x) == true)
        {
            medicine_id.Add(medicine[x].ValueMember);
        }
    }

    cmd.CommandText = "sp_insert_examination";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
    cmd.Parameters.Add("@vet_id", SqlDbType.Int).Value = vet_id;
    cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = sheep_id;
    cmd.Parameters.Add("@comments", SqlDbType.NVarChar).Value = comment;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    reader = cmd.ExecuteReader();

    reader.Read();
    examination_id = reader.GetInt32(0);
    int i=0;
    reader.Close();
    cmd.CommandText = "sp_insert_examination_illness";

    for (i = 0; i < illness_id.Count; i++)
    {
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value=examination_id;
        cmd.Parameters.Add("@illness_id", SqlDbType.Int).Value=illness_id[i];
        cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
        cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
        cmd.ExecuteNonQuery();
    }

    cmd.CommandText = "sp_insert_examination_medicine";
```

```

        for (i = 0; i < medicine_id.Count; i++)
        {
            cmd.Parameters.Clear();
            cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
examination_id;
            cmd.Parameters.Add("@medicine_id", SqlDbType.Int).Value = medicine_id[i];
            cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
            cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
            cmd.ExecuteNonQuery();
        }
        cmd.CommandText = "sp_insert_sheep_illness";

        for (i = 0; i < illness_id.Count; i++)
        {
            cmd.Parameters.Clear();
            cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value=sheep_id;
            cmd.Parameters.Add("@illness_id", SqlDbType.Int).Value=illness_id[i];
            cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value=examination_id;
            cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
            cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
            cmd.ExecuteNonQuery();
        }
        cmd.CommandText = "sp_insert_sheep_medicine";

        for (i = 0; i < medicine_id.Count; i++)
        {
            cmd.Parameters.Clear();
            cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value=sheep_id;
            cmd.Parameters.Add("@medicine_id", SqlDbType.Int).Value = medicine_id[i];
            cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value = examination_id;
            cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
            cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
            cmd.ExecuteNonQuery();
        }
        cmd.Connection.Close();
    }
    catch
    {
        MessageBox.Show("not connected " + exc.Message);
    }
}

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataReader reader;
Exception exc = new Exception();
try
{
    int x = 0;

    for (x = 0; x < 1stIllness.Items.Count; x++)
    {
        if (1stIllness.GetSelected(x)==true)
        {
            illness_id.Add(illness[x].ValueMember);
        }
    }

    for (x = 0; x < 1stMedicine.Items.Count; x++)
    {

```

```

        if (lstMedicine.GetSelected(x) == true)
        {
            medicine_id.Add(medicine[x].ValueMember);
        }
    }

    cmd.CommandText = "sp_update_examination";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
Module.examination_id;
    cmd.Parameters.Add("@condition", SqlDbType.Char).Value = '1';
    cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
    cmd.Parameters.Add("@vet_id", SqlDbType.Int).Value = vet_id;
    cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = sheep_id;
    cmd.Parameters.Add("@comments", SqlDbType.NVarChar).Value = comment;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    cmd.ExecuteNonQuery();

    cmd.CommandText = "sp_update_examination_illness";
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
Module.examination_id;
    cmd.ExecuteNonQuery();

    cmd.CommandText = "sp_insert_examination_illness";

    for (i = 0; i < illness_id.Count; i++)
    {
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value = examination_id;
        cmd.Parameters.Add("@illness_id", SqlDbType.Int).Value = illness_id[i];
        cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
        cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
        cmd.ExecuteNonQuery();
    }

    cmd.CommandText = "sp_update_examination_medicine";
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
Module.examination_id;
    cmd.ExecuteNonQuery();

    cmd.CommandText = "sp_insert_examination_medicine";

    for (i = 0; i < medicine_id.Count; i++)
    {
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
examination_id;
        cmd.Parameters.Add("@medicine_id", SqlDbType.Int).Value = medicine_id[i];
        cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;

```

```

        cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
        cmd.ExecuteNonQuery();
    }

    cmd.CommandText = "sp_update_sheep_illness";
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
Module.examination_id;
    cmd.ExecuteNonQuery();

    cmd.CommandText = "sp_insert_sheep_illness";

    for (i = 0; i < illness_id.Count; i++)
    {
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value=sheep_id;
        cmd.Parameters.Add("@illness_id", SqlDbType.Int).Value=illness_id[i];
        cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value=examination_id;
        cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
        cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
        cmd.ExecuteNonQuery();
    }

    cmd.CommandText = "sp_update_sheep_medicine";
    cmd.Parameters.Clear();
    cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
Module.examination_id;
    cmd.ExecuteNonQuery();

    cmd.CommandText = "sp_insert_sheep_medicine";

    for (i = 0; i < medicine_id.Count; i++)
    {
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value=sheep_id;
        cmd.Parameters.Add("@medicine_id", SqlDbType.Int).Value = medicine_id[i];
        cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value = examination_id;
        cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value = date;
        cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comment;
        cmd.ExecuteNonQuery();
    }
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();
try
{

```

```

cmd.CommandText = "sp_search_examination";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;
cmd.Parameters.Clear();
cmd.Parameters.Add("@examin_date", SqlDbType.SmallDateTime).Value=date;
cmd.Parameters.Add("@vet_id", SqlDbType.Int).Value = vet_id;
cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = sheep_id;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
adapter.SelectCommand = cmd;

try
{
    adapter.Fill(dataset, "search_examination");
}
catch (Exception excep)
{
    MessageBox.Show(excep.Message);
}
cmd.Connection.Close();

dgvExamination.DataSource = dataset;
dgvExamination.DataMember = "search_examination";

int counter = 0;
int rows = dgvExamination.RowCount;
this.dgvExamination.DefaultCellStyle.Font = new Font("Microsoft Sans Serif",
12);
this.dgvExamination.DefaultCellStyle.ForeColor = Color.Black;

this.dgvExamination.DefaultCellStyle.SelectionForeColor = Color.White;
this.dgvExamination.DefaultCellStyle.SelectionBackColor = Color.Black;
this.dgvExamination.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
this.dgvExamination.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}

```

Κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_delete_examination";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@examination_id", SqlDbType.Int).Value =
Module.examination_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
}

```

```

    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.9 Διαχείριση Εσόδων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου εσόδου της κτηνοτροφικής μονάδας.

Κώδικας εισαγωγής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

cmd.CommandText = "sp_insert_incoming";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@incom_date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = description;
cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value = unit;
cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
cmd.CommandText = "sp_update_incoming";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@incoming_id", SqlDbType.Int).Value = Module.incoming_id;
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@incom_date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = description;

```

```

cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value = unit;
cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
DataSet dataset = new DataSet();

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

cmd.CommandText = "sp_search_incoming";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@incom_date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = description;
cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value = unit;
cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
adapter.SelectCommand = cmd;

try
{
    adapter.Fill(dataset, "search_incoming");
}
catch (Exception excep)
{
    MessageBox.Show(excep.Message);
}
cmd.Connection.Close();
dgvIncoming.DataSource = dataset;
dgvIncoming.DataMember = "search_incoming";

int counter = 0;
int rows = dgvIncoming.RowCount;
this.dgvIncoming.DefaultCellStyle.Font = new Font("Microsoft Sans Serif",
12);
this.dgvIncoming.DefaultCellStyle.ForeColor = Color.Black;

this.dgvIncoming.DefaultCellStyle.SelectionForeColor = Color.White;
this.dgvIncoming.DefaultCellStyle.SelectionBackColor = Color.Black;
this.dgvIncoming.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
this.dgvIncoming.BackgroundColor = Color.LightSteelBlue;

```

Κώδικας Διαγραφής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_delete_incoming";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@incoming_id", SqlDbType.Int).Value = Module.incoming_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}
```

5.3.6.11 Διαχείριση Εξόδων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου εσόδου της κτηνοτροφικής μονάδας.

Κώδικας Εισαγωγής

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

cmd.CommandText = "sp_insert_outgoing";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@outg_date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@outg_description", SqlDbType.NVarChar).Value =
description;
cmd.Parameters.Add("@outg_unit", SqlDbType.NVarChar).Value = unit;
cmd.Parameters.Add("@outg_price_per_unit", SqlDbType.Decimal).Value = price;
cmd.Parameters.Add("@condition", SqlDbType.Char).Value='1';
cmd.Parameters.Add("@outg_state", SqlDbType.SmallInt).Value=state;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
```

Κώδικας ενημέρωσης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
cmd.CommandText = "sp_update_outgoing";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@outgoing_id", SqlDbType.Int).Value = Module.outgoing_id;
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@outg_date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@outg_description", SqlDbType.NVarChar).Value =
description;
cmd.Parameters.Add("@outg_unit", SqlDbType.NVarChar).Value = unit;
cmd.Parameters.Add("@outg_price_per_unit", SqlDbType.Decimal).Value = price;
cmd.Parameters.Add("@condition", SqlDbType.Char).Value = '1';
cmd.Parameters.Add("@outg_state", SqlDbType.SmallInt).Value = state;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
```

Κώδικας αναζήτησης:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
DataSet dataset = new DataSet();

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

cmd.CommandText = "sp_search_outgoing";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = description;
cmd.Parameters.Add("@unit", SqlDbType.NVarChar).Value = unit;
cmd.Parameters.Add("@price", SqlDbType.Decimal).Value = price;
cmd.Parameters.Add("@state", SqlDbType.SmallInt).Value = state;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
adapter.SelectCommand = cmd;

try
{
    adapter.Fill(dataset, "search_incoming");
```

```

    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }
    cmd.Connection.Close();
    dgvOutgoing.DataSource = dataset;
    dgvOutgoing.DataMember = "search_incoming";

    int counter = 0;
    int rows = dgvOutgoing.RowCount;
    this.dgvOutgoing.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvOutgoing.DefaultCellStyle.ForeColor = Color.Black;
    this.dgvOutgoing.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvOutgoing.DefaultCellStyle.SelectionBackColor = Color.Black;
    this.dgvOutgoing.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvOutgoing.BackgroundColor = Color.LightSteelBlue;

```

κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_delete_outgoing";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@outgoing_id", SqlDbType.Int).Value = Module.outgoing;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.11 Διαχείριση Τιμολογίων Εσόδων/Εξόδων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιου τιμολογίου της κτηνοτροφικής μονάδας το οποία αναφέρεται σε κάποιο έσοδο ή έξοδο.

Κώδικας εισαγωγής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{

```

```

adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

cmd.CommandText = "sp_insert_invoice_in";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

date = dtpDate.Value.ToString();
num_of_units = Convert.ToDecimal(txtNumUnits.Text);
amount = Convert.ToDecimal(txtTotalAmount.Text);

cmd.Parameters.Clear();
cmd.Parameters.Add("@owner_id", SqlDbType.Int).Value = owner_id;
cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
cmd.Parameters.Add("@incoming_id", SqlDbType.Int).Value = incoming_id;
cmd.Parameters.Add("@inv_date", SqlDbType.SmallDateTime).Value = date;
cmd.Parameters.Add("@num_of_units", SqlDbType.Decimal).Value = num_of_units;
cmd.Parameters.Add("@total_amount", SqlDbType.Decimal).Value = amount;
cmd.Parameters.Add("@paid", SqlDbType.Decimal).Value = 0;
cmd.Parameters.Add("@remaining", SqlDbType.Decimal).Value = amount;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    cmd.CommandText = "sp_update_invoice_in";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    date = dtpDate.Value.ToString();
    num_of_units = Convert.ToDecimal(txtNumUnits.Text);
    amount = Convert.ToDecimal(txtTotalAmount.Text);
    paid = Convert.ToDecimal(txtPaid.Text);
    remaining = amount - paid;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@invoice_id", SqlDbType.Int).Value = Module.invoice_id;
    cmd.Parameters.Add("@owner_id", SqlDbType.Int).Value = owner_id;
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
    cmd.Parameters.Add("@incoming_id", SqlDbType.Int).Value = incoming_id;
    cmd.Parameters.Add("@inv_date", SqlDbType.SmallDateTime).Value = date;
    cmd.Parameters.Add("@num_of_units", SqlDbType.Decimal).Value = num_of_units;
    cmd.Parameters.Add("@total_amount", SqlDbType.Decimal).Value = amount;
    cmd.Parameters.Add("@paid", SqlDbType.Decimal).Value = paid;
}

```

```

cmd.Parameters.Add("@remaining", SqlDbType.Decimal).Value = remaining;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας αναζήτησης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();

try
{
    date = dtpDate.Value.ToString();
    cmd.CommandText = "sp_search_invoice_in";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@owner_id", SqlDbType.Int).Value = owner_id;
    cmd.Parameters.Add("@farm_id", SqlDbType.Int).Value = farm_id;
    cmd.Parameters.Add("@incoming_id", SqlDbType.Int).Value = incoming_id;
    cmd.Parameters.Add("@inv_date", SqlDbType.SmallDateTime).Value = date;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_invoice");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }
    cmd.Connection.Close();

    dgvInvoice.DataSource = dataset;
    dgvInvoice.DataMember = "search_invoice";

    int counter = 0;
    int rows = dgvInvoice.RowCount;

    this.dgvInvoice.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvInvoice.DefaultCellStyle.ForeColor = Color.Black;
    this.dgvInvoice.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvInvoice.DefaultCellStyle.SelectionBackColor = Color.Black;
}

```

```

        this.dgvInvoice.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
        this.dgvInvoice.BackgroundColor = Color.LightSteelBlue;
    }
    catch
    {
        MessageBox.Show("Exception :" + exc.Message);
    }
}

```

Κώδικας διαγραφής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();

try
{
    cmd.CommandText = "sp_delete_invoice_in";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@invoice_id", SqlDbType.Int).Value = Module.invoice_id;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}
}

```

5.3.6.12 Διαχείριση Αποδείξεων

Στο τμήμα αυτό έχουν υλοποιηθεί η εισαγωγή, ενημέρωση, αναζήτηση, διαγραφή και παρουσίαση των στοιχείων κάποιας απόδειξης της κτηνοτροφικής μονάδας το οποία αναφέρεται στο τιμολόγιο κάποιου εσόδου ή εξόδου.

Κώδικας εισαγωγής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();
try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();
    cmd.CommandText = "sp_insert_receipt_in";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@invoice_id", SqlDbType.Int).Value = Module.invoice_id;
    cmd.Parameters.Add("@amount_given", SqlDbType.Decimal).Value = given;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = comments;
    cmd.Parameters.Add("@rec_date", SqlDbType.SmallDateTime).Value = date;
    cmd.Parameters.Add("@paid", SqlDbType.Decimal).Value = paid;
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}
}

```

```

cmd.Parameters.Add("@remaining", SqlDbType.Decimal).Value = remaining;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας ενημέρωσης:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
Exception exc = new Exception();

try
{
    adapter.SelectCommand = new System.Data.SqlClient.SqlCommand();

    if (lblPaid.Text != "")
        paid = Convert.ToDecimal(lblPaid.Text);
    else
        paid = 0;

    if (lblRemaining.Text != "")
        remaining = Convert.ToDecimal(lblRemaining.Text);
    else
        remaining = 0;

    cmd.CommandText = "sp_update_receipt_in";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@receipt_id", SqlDbType.Int).Value = Module.receipt_id;
    cmd.Parameters.Add("@invoice_id", SqlDbType.Int).Value = invoice_id;
    cmd.Parameters.Add("@amount_given", SqlDbType.Decimal).Value = price;
    cmd.Parameters.Add("@description", SqlDbType.NVarChar).Value = descr;
    cmd.Parameters.Add("@rec_date", SqlDbType.SmallDateTime).Value = date;
    cmd.Parameters.Add("@paid", SqlDbType.Decimal).Value = paid;
    cmd.Parameters.Add("@remaining", SqlDbType.Decimal).Value = remaining;
    cmd.Parameters.Add("@flag", SqlDbType.Int).Value = flag2;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    cmd.ExecuteNonQuery();
    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας αναζήτησης:

```
System.Data.SqlClient.SqlDataAdapter adapter = new
System.Data.SqlClient.SqlDataAdapter();
DataSet dataset = new DataSet();
Exception exc = new Exception();
try
{
    cmd.CommandText = "sp_search_receipt_in";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;
    if (txtSelectedInvoice.Text == "")
        Module.invoice_id = 0;

    cmd.Parameters.Clear();
    cmd.Parameters.Add("@invoice_id", SqlDbType.Int).Value = Module.invoice_id;
    cmd.Parameters.Add("@amount_given", SqlDbType.Decimal).Value = given;
    cmd.Parameters.Add("@rec_date", SqlDbType.SmallDateTime).Value = date;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    adapter.SelectCommand = cmd;

    try
    {
        adapter.Fill(dataset, "search_receipt");
    }
    catch (Exception excep)
    {
        MessageBox.Show(excep.Message);
    }

    cmd.Connection.Close();

    dgvReceipt.DataSource = dataset;
    dgvReceipt.DataMember = "search_receipt";

    int counter = 0;
    int rows = dgvReceipt.RowCount;

    this.dgvReceipt.DefaultCellStyle.Font = new Font("Microsoft Sans Serif", 12);
    this.dgvReceipt.DefaultCellStyle.ForeColor = Color.Black;
    this.dgvReceipt.DefaultCellStyle.SelectionForeColor = Color.White;
    this.dgvReceipt.DefaultCellStyle.SelectionBackColor = Color.Black;
    this.dgvReceipt.AlternatingRowsDefaultCellStyle.BackColor = Color.Beige;
    this.dgvReceipt.BackgroundColor = Color.LightSteelBlue;
}
catch
{
    MessageBox.Show("Exception :" + exc.Message);
}
```

Κώδικας διαγραφής:

```
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();

try
{
```

```

cmd.CommandText = "sp_delete_receipt_in";
cmd.CommandType = CommandType.StoredProcedure;
cmd.Connection = Module.connection;

cmd.Parameters.Clear();
cmd.Parameters.Add("@receipt_id", SqlDbType.Int).Value = Module.receipt_id;

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}
cmd.ExecuteNonQuery();
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("Exception: " + exc.Message);
}

```

5.3.6.13 Διαχείριση παραγωγής γάλακτος

Στο τμήμα αυτό έχει υλοποιηθεί η εισαγωγή παραγωγής γάλακτος για κάθε πρόβατο, η ομαδοποίηση των προβάτων ανάλογα με την θέση της ‘κυψέλης’ που βρίσκονται στο γαλεντήριο, και η δημιουργία αναφορών σχετικά με την παραγωγή γάλακτος για όλα τα πρόβατα ή για κάθε πρόβατο ξεχωριστά.

Κώδικας εισαγωγής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
int i = 0, j = 0;
double milk = 0.0;
string date = dtpDate.Value.ToString();
int sheep_id;
char ltr = 'A';

try
{
    cmd.CommandText = "sp_insert_daily_milk";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }

    for (i = 0; i < 12; i++)
    {
        milk = 0.0;
        try
        {
            if (txtBoxList[i].Text != "")
            {
                milk = Convert.ToDouble(txtBoxList[i].Text);
            }
            else

```

```

        milk = 0.0;
    }
    catch
    {
        MessageBox.Show("Error Saving: Invalid Input for parameter: Milk  
Production at Position" + ltr);
    }

    j = 0;
    j = cmbBoxList[i].SelectedIndex;
    if (j != -1)
    {
        sheep_id = myCombos[i][j].ValueMember;
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@daily_date", SqlDbType.SmallDateTime).Value =
date;
        cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = sheep_id;
        cmd.Parameters.Add("@milk", SqlDbType.Decimal).Value = milk;
        cmd.ExecuteNonQuery();
    }
    ltr++;
}
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας ομαδοποίησης:

```

string query = "UPDATE SHEEP SET Team=@team WHERE sheep_id=@sheep_id";
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
char team;
Exception exc = new Exception();
try
{
    cmd.CommandText = query;
    cmd.CommandType = CommandType.Text;
    cmd.Connection = Module.connection;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    int i = 0;
    for (i = 0; i < dgvSheep.Rows.Count; i++)
    {
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value =
Convert.ToInt32(dgvSheep.Rows[i].Cells[0].Value);
        if (Convert.ToChar(dgvSheep.Rows[i].Cells[4].Value) == ' ')
            team = '-';
        else
        {
            team = Convert.ToChar(dgvSheep.Rows[i].Cells[4].Value);
            team = Char.ToUpper(team);
        }
        cmd.Parameters.Add("@team", SqlDbType.Char).Value = team;
        cmd.ExecuteNonQuery();
    }
}

```

```

        cmd.Connection.Close();
    }
    catch
    {
        MessageBox.Show("not connected " + exc.Message);
    }
}

```

Κώδικας δημιουργίας αναφορών:

```

mother_id = myCombo[cmbMother.SelectedIndex].ValueMember;

string source;
source = Module.reports_source + "rptMilk2.rpt";

ReportDocument rd = new ReportDocument();
rd.Load(source);
ParameterFields parameterFields = new ParameterFields();
ParameterField parameterField = null;
ParameterDiscreteValue parameterValue = null;

parameterField = new ParameterField();
parameterValue = new ParameterDiscreteValue();
parameterField.Name = "@sheep_id";
parameterValue.Value = Convert.ToString(mother_id);
parameterField.CurrentValues.Add(parameterValue);
parameterFields.Add(parameterField);
try
{
    rd.SetParameterValue("@sheep_id", parameterValue);
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message);
}
cryMilk2.ReportSource = rd;
cryMilk2.Refresh();

```

5.3.6.14 Διαχείριση κατανάλωσης τροφής

Στο τμήμα αυτό έχει υλοποιηθεί η εισαγωγή κατανάλωσης τροφής για κάθε ομάδα προβάτων τα οποία βρίσκονται στην ίδια θέση στην φάρμα, η ομαδοποίηση των προβάτων ανάλογα με την θέση της ‘κυψέλης’ που βρίσκεται το κάθε ζώο μέσα στην φάρμα, και η δημιουργία αναφορών σχετικά με την κατανάλωση τροφής για όλα τα πρόβατα ή για κάθε πρόβατο ξεχωριστά.

Κώδικας εισαγωγής:

```

System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
Exception exc = new Exception();
char ltr='A';

try
{
    cmd.CommandText = "sp_insert_daily_food";
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Connection = Module.connection;
}

```

```

if (cmd.Connection.State != ConnectionState.Open)
{
    cmd.Connection.Open();
}

for (i = 0; i < 12; i++)
{
    food = 0.0;
    try
    {
        if (txtBoxList[i].Text != "")
        {
            food = Convert.ToDouble(txtBoxList[i].Text);
        }
        else
        {
            food = 0.0;
        }
    }
    catch
    {
        MessageBox.Show("Error Saving: Invalid Input for parameter: Food Consumption at Group"+ ltr);
    }

    for (j = 0; j < cmbBoxList[i].Items.Count; j++)
    {
        sheep_id = myCombos[i][j].ValueMember;
        cmd.Parameters.Clear();
        cmd.Parameters.Add("@daily_date", SqlDbType.SmallDateTime).Value =
date;
        cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value = sheep_id;
        cmd.Parameters.Add("@food", SqlDbType.Decimal).Value=food;
        cmd.ExecuteNonQuery();
    }
    ltr++;
}
cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας ομαδοποίησης:

```

string query = "UPDATE SHEEP SET Team=@team WHERE sheep_id=@sheep_id";
System.Data.SqlClient.SqlCommand cmd = new System.Data.SqlClient.SqlCommand();
char team;
Exception exc = new Exception();
try
{
    cmd.CommandText = query;
    cmd.CommandType = CommandType.Text;
    cmd.Connection = Module.connection;

    if (cmd.Connection.State != ConnectionState.Open)
    {
        cmd.Connection.Open();
    }
    int i = 0;
    for (i = 0; i < dgvSheep.Rows.Count; i++)
    {

```

```

        cmd.Parameters.Clear();
        cmd.Parameters.Add("@sheep_id", SqlDbType.Int).Value =
Convert.ToInt32(dgvSheep.Rows[i].Cells[0].Value);
        if (Convert.ToChar(dgvSheep.Rows[i].Cells[4].Value) == ' ')
            team = '-';
        else
        {
            team = Convert.ToChar(dgvSheep.Rows[i].Cells[4].Value);
            team = Char.ToUpper(team);
        }
        cmd.Parameters.Add("@team", SqlDbType.Char).Value =team;
        cmd.ExecuteNonQuery();
    }

    cmd.Connection.Close();
}
catch
{
    MessageBox.Show("not connected " + exc.Message);
}

```

Κώδικας δημιουργίας αναφορών:

```

sheep_id = myCombo[cmbSheep.SelectedIndex].ValueMember;

string source;
source = Module.reports_source + "rptFood2.rpt";

ReportDocument rd = new ReportDocument();
rd.Load(source);
ParameterFields parameterFields = new ParameterFields();
ParameterField parameterField = null;
ParameterDiscreteValue parameterValue = null;

parameterField = new ParameterField();
parameterValue = new ParameterDiscreteValue();
parameterField.Name = "@sheep_id";
parameterValue.Value = Convert.ToString(sheep_id);
parameterField.CurrentValues.Add(parameterValue);
parameterFields.Add(parameterField);
try
{
    rd.SetParameterValue("@sheep_id", parameterValue);
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message);
}

cryFood2.ReportSource = rd;
cryFood2.Refresh();

```

Κεφάλαιο 6. Συμπεράσματα

Κεφάλαιο 6. Συμπεράσματα	244
6.1 Εισαγωγή	244
6.2 Ευχρηστία.....	244
6.3 Μελλοντική Εργασία	245

Κεφάλαιο 6. Συμπεράσματα

6.1 Εισαγωγή

Αφού το υπό-ανάπτυξη σύστημα παραδοθεί στους τελικούς χρήστες του, θα πρέπει τότε να γίνει και η αξιολόγηση του. Με τον όρο αξιολόγηση ενός συστήματος, εννοείται η καταμέτρησης της επίδοσης αλλά και της αποδοτικότητας του. Γενικά, όταν μία εταιρεία αναλαμβάνει να διεκπεραιώσει ένα συγκεκριμένο λογισμικό συστήματος, πριν το λογισμικό αυτό να βγει στην αγορά, πρέπει να επιβεβαιώσει ότι το λογισμικό πληροί στο έπακρο τους στόχους οι οποίοι αρχικά καθορίστηκαν. Επιπλέον, πρέπει να γνωρίζει τα μειονεκτήματα του και τα πλεονεκτήματα του. Η αξιολόγηση είναι η διαδικασία κατά την οποία η εταιρεία που αναπτύσσει ένα λογισμικό μπορεί να επιβεβαιώσει ότι το λογισμικό καλύπτει τις ανάγκες των πελατών της.

Σημαντική παράμετρος στην αξιολόγηση ενός λογισμικού είναι η ευχρηστία. Ένα υλοποιημένο σύστημα που ανταποκρίνεται στις προσδοκίες του χρήστη και ικανοποιεί τις απαιτήσεις του, κάνει την δουλειά των τελικών χρηστών περισσότερο αποδοτική και εύκολη, εξαιτίας της αλληλεπίδρασης τους με το λογισμικό. Τέλος, η ευχρηστία αποτελεί ένα από τους σημαντικότερους παράγοντες για την επιτυχία ενός λογισμικού συστήματος.

6.2 Ευχρηστία

Ευχρηστία ενός συστήματος αποτελεί την δυνατότητα ενός προϊόντος συστήματος ή υπηρεσίας που χρησιμοποιείται από καθορισμένους χρήστες με καθορισμένους στόχους, υπό καθορισμένες συνθήκες χρήσης, να είναι αποτελεσματικό, αποδοτικό, και να παρέχει υποκειμενική ικανοποίηση στους χρήστες του.

Κατά την αξιολόγηση ενός συστήματος, εξετάζεται ανάμεσα σε άλλα η ευχρηστία του, συνήθως μέσω ερωτηματολογίων ή συνεντεύξεων, που σχετίζονται με το λογισμικό. Η αξιολόγηση του συστήματος πρέπει να καλύπτει όλες τις πτυχές του, με σκοπό να δημιουργηθεί μία ολοκληρωμένη εικόνα για τις λειτουργίες που είναι ικανό να προσφέρει το συγκεκριμένο λογισμικό. Με την βοήθεια των ερωτηματολογίων και των προσωπικών συνεντεύξεων, διαπιστώνεται κατά πόσο οι χρήστες μπορούν να χρησιμοποιήσουν το σύστημα, η αποδοτικότητα του συστήματος, η σχεδίαση και η λειτουργικότητα του.

6.3 Μελλοντική Εργασία

Μερικές από τις επεκτάσεις που θα μπορούσαν να εφαρμοστούν στο συγκεκριμένο λογισμικό συστήματος, το οποίο αναπτύχθηκε στα πλαίσια της διπλωματικής αυτής εργασίας, είναι οι εξής:

- Ανάπτυξη του συστήματος στον παγκόσμιο ιστό: το λογισμικό αναπτύχθηκε σαν stand-alone εφαρμογή. Για περισσότερη ευχρηστία, θα μπορούσε να αναπτυχθεί σαν διαδικτυακή εφαρμογή. Ο λόγος που δεν αναπτύχθηκε στον ιστό, στα πλαίσια αυτής της διπλωματικής είναι το γεγονός ότι στα περισσότερα από τα υποστατικά των κτηνοτροφικών μονάδων δεν υπάρχει σύνδεση διαδικτύου.
- Ανάπτυξη μέρους της εφαρμογής σε κινητό τηλέφωνο: μία από τις σημαντικότερες λειτουργίες του συγκεκριμένου λογισμικού είναι η καταχώρηση της γαλακτοπαραγωγής για κάθε πρόβατο, αλλά και της κατανάλωσης του σε τροφή. Επειδή οι λειτουργίες αυτές είναι που χρησιμοποιούνται περισσότερο σε σχέση με τις υπόλοιπες, θα μπορούσαν να αναπτυχθούν σε εφαρμογή για κινητά τηλέφωνα, με σκοπό να διευκολύνει τον χρήστη στην καταχώρηση των στοιχείων αυτών.
- Προσθήκη στο υπάρχον λογισμικό ενός portal το οποίο να καταχωρεί αυτόματα τα στοιχεία από την ημερήσια γαλακτοπαραγωγή των προβάτων. Δηλαδή, το γαλευτήρι θα συνδέεται μέσω portal με την εφαρμογή, ο χρήστης θα επιλέγει το πρόβατο που βρίσκεται σε κάθε κυψελίδα, και τότε αυτόματα θα καταχωρείται η ποσότητα παραγωγής γάλακτος του κάθε προβάτου. Ο λόγος που δεν υλοποιήθηκε στο υπάρχον λογισμικό είναι ότι το portal είναι πολύ ακριβό για να αγοραστεί και οι ιδιοκτήτες των κτηνοτροφικών μονάδων δεν μπορούν να επωμισθούν τέτοια έξοδα.

Βιβλιογραφία

1. Fundamentals of Database Systems, 5/E Ramez Elmasri, Shamkant B. Navathe
Publisher: Addison-Wesley 1168 pages, 2007.
2. Θεμελιώδης Αρχές Συστημάτων Βάσεων Δεδομένων- 5^η έκδοση (1^{ος} τόμος),
γραμμένο από τους Ramez Elmasri και Sham B. Navathe (μετάφραση Μιχάλης
Χατζόπουλος), Εκδόσεις Δίαυλος, 2007.
3. Sommerville, Software Engineering, 9th Edition, Addison-Wesley, 2011.
[ελληνική μετάφραση για την 8^η έκδοση] I. Sommerville, Βασικές Αρχές
Τεχνολογίας Λογισμικού, 8^η έκδοση, Εκδ. Κλειδάριθμος, Μάιος 2009.
4. S. R. Schach, Classical and Object-Oriented Software Engineering, Eighth
Edition, McGrawHill, 2011.
5. «Export to Excel», <http://www.codeproject.com/Articles/248531/Export-Excel-File-for-Csharp>
6. “Export to Word”, http://www.c-sharpcorner.com/UploadFile/amrish_deep/WordAutomation05102007223934PM/WordAutomation.aspx
7. “Connect Visual Studio with Matlab”,
<http://www.mathworks.com/matlabcentral/fileexchange/12987-integrating-matlab-with-c>
8. “Store varbinary images in sql”, <http://msdn.microsoft.com/en-us/library/ms188362.aspx>
9. Διδακτορική διατριβή; «Αξιοποίηση του ελέγχου γαλακτοπαραγωγής
προβατίνων της φυλής Χίου», <http://invenio.lib.auth.gr/record/60923/files/gri-2006-901.pdf?version=1>

Παράρτημα Α

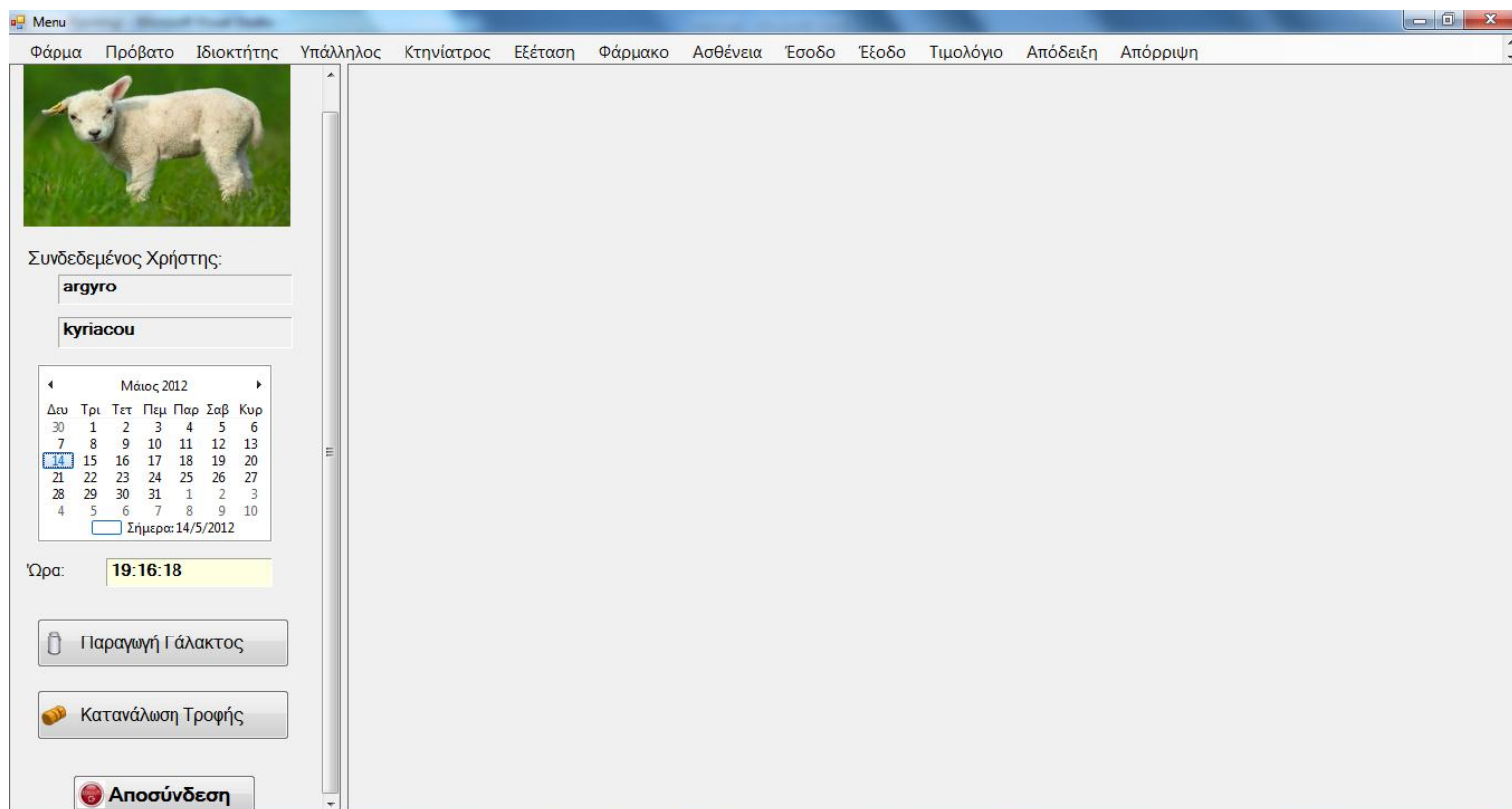
1. Log-In



Εικόνα A-2

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα εισαγωγής του χρήστη στο σύστημα. Ο χρήστης μπορεί να είναι είτε ο ιδιοκτήτης κάποιας από τις φάρμες, είτε υπάλληλος σε κάποια από αυτές. Τα στοιχεία που πρέπει να εισάγει είναι το συνθηματικό του όνομα και κωδικό στα κατάλληλα πεδία, και ακολούθως γίνεται έλεγχος για την ορθότητα τους. Αν ο έλεγχος είναι ορθός, τότε γίνεται εισαγωγή του χρήστη στο σύστημα, αν όχι εμφανίζεται κατάλληλο μήνυμα στον χρήστη. Επιπλέον, παρατηρούμε ότι το σύστημα είναι διαθέσιμο σε δύο γλώσσες: Αγγλικά και Ελληνικά. Κάποιος ο οποίος θέλει να έχει πρόσβαση στο σύστημα, μπορεί να δημιουργήσει λογαριασμό μέσω του κουμπιού: REGISTER, όπου εμφανίζει την φόρμα εισαγωγής του ιδιοκτήτη ή του υπαλλήλου, ανάλογα με το τι έχει επιλεγεί προηγουμένως.

2. Κεντρικό μενού επιλογών

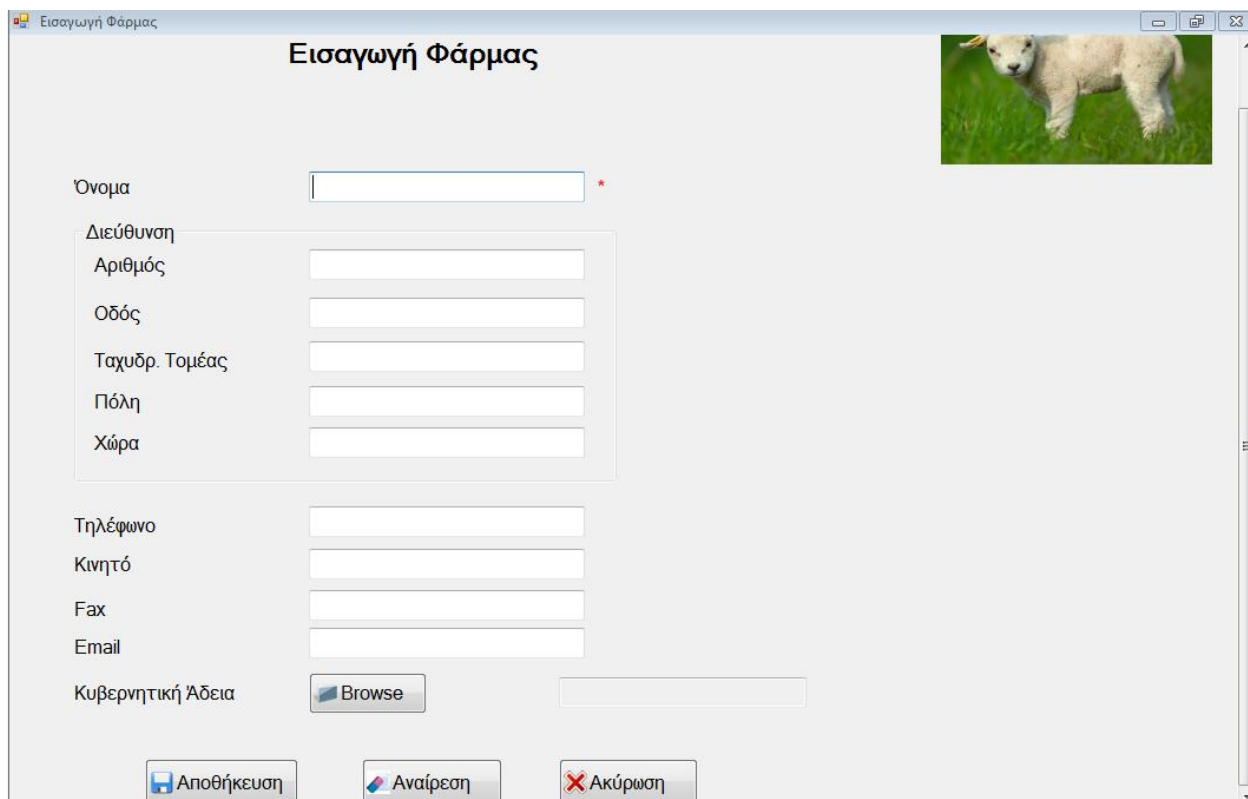


Εικόνα A-3

Στην πιο πάνω εικόνα παρουσιάζεται το κεντρικό μενού επιλογών, το οποίο είναι η βασική επιφάνεια εργασίας του συστήματος. Στο πάνω μέρος, παρατηρούνται οι βασικές κατηγορίες του συστήματος στις οποίες μπορεί να εφαρμοστούν οι διαδικασίες της εισαγωγής, ενημέρωσης, αναζήτησης, διαγραφής, καθώς επίσης και της δημιουργίας αναφορών μέσω του εργαλείου Crystal Report Viewer. Επιπλέον, στα αριστερά της εικόνας παρατηρείται το όνομα και του επίθετο του συνδεδεμένου χρήστη, ένα ημερολόγιο και η τοπική ώρα, και επιπλέον οι κατηγορίες των διαδικασιών Παραγωγής Γάλακτος και Κατανάλωσης Τροφής. Οι κατηγορίες αυτές έχουν επιλεγεί να βρίσκονται σε αυτή την θέση εξαιτίας της συχνής χρήσης τους σε σχέση με τις υπόλοιπες κατηγορίες. Τέλος, η τελευταία επιλογή στο κεντρικό μενού επιλογών είναι η Αποσύνδεση, όπου με επιλογή της ο χρήστης μεταφέρεται στην φόρμα Log-In (Εικόνα A-1).

3. Κατηγορία Φάρμα

3.1 Εισαγωγή

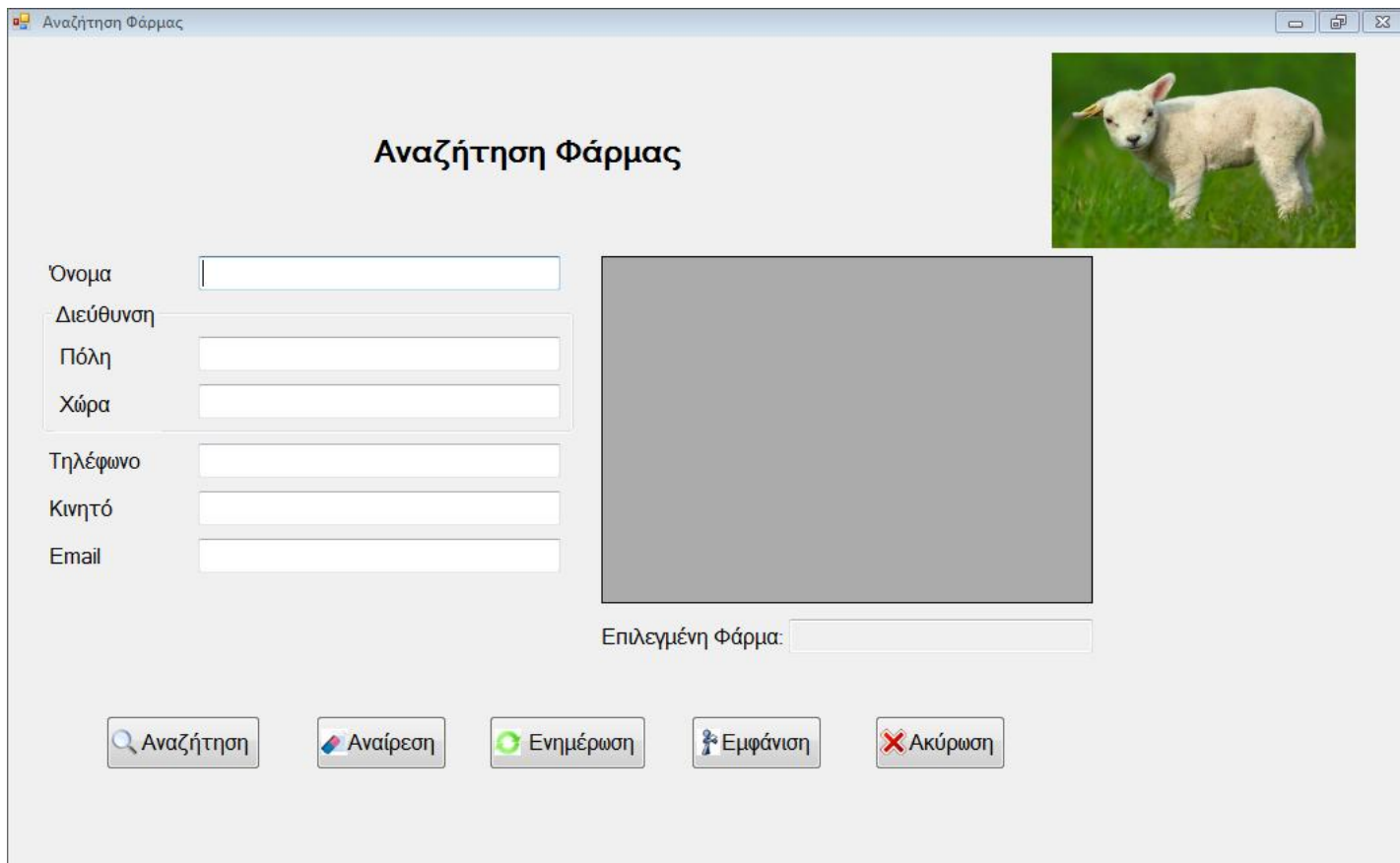


Εικόνα Α-4

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Φάρμα, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα Farm, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να εισάγει ένα όνομα για την φάρμα. Προαιρετικά, ο χρήστης μπορεί να εισάγει την πλήρη διεύθυνση στην οποία βρίσκεται η φάρμα (αριθμός, οδός, ταχυδρομικός τομέας, πόλη και χώρα), στοιχεία επικοινωνίας όπως τηλέφωνο, κινητό, Fax και e-mail, και τέλος, μπορεί μέσω της επιλογής του κουμπιού: Browse, κατά την οποία εμφανίζεται ένα καινούργιο παράθυρο πλοήγησης στα αρχεία του υπολογιστή, ο χρήστης μπορεί να επιλέξει το αρχείο στο οποίο βρίσκεται η κυβερνητική άδεια/ άδεια λειτουργίας της κτηνοτροφικής μονάδας. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην

ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

3.2 Αναζήτηση

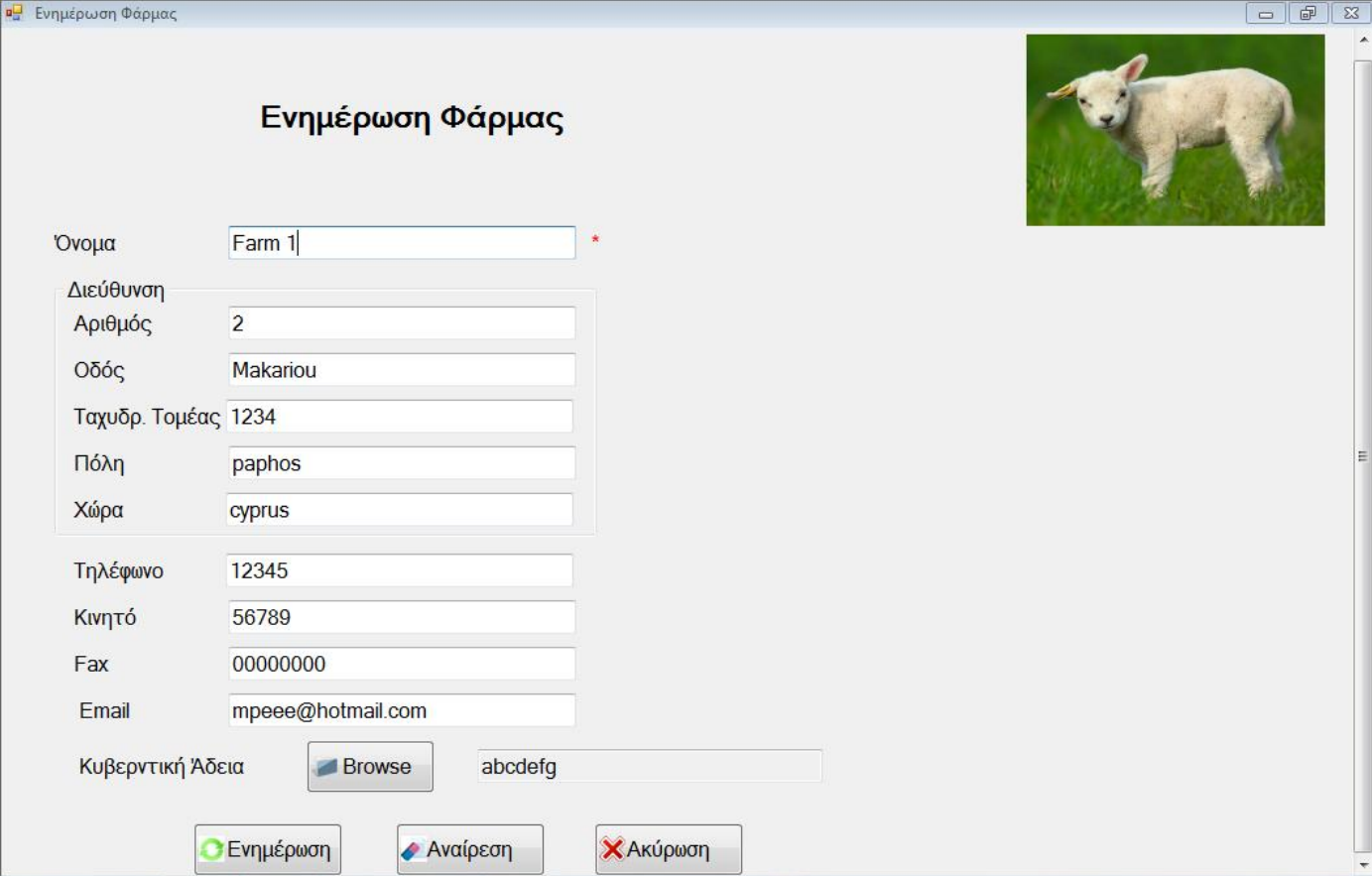


Εικόνα Α-5

Στην πιο πάνω εικόνα, παρουσιάζεται η διαδικασία αναζήτησης κάποιας προϋπάρχουσας φάρμας από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις κτηνοτροφικές μονάδες. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που

εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων της επιλεγμένης φάρμας, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία της συγκεκριμένης φάρμας και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

3.3 Ενημέρωση



Ενημέρωση Φάρμας

Όνομα Farm 1 *

Διεύθυνση

Αριθμός 2

Οδός Makariou

Ταχυδρ. Τομέας 1234

Πόλη paphos

Χώρα cypus

Τηλέφωνο 12345

Κινητό 56789

Fax 00000000

Email mpeee@hotmail.com

Κυβερντική Άδεια Browse abcdefg

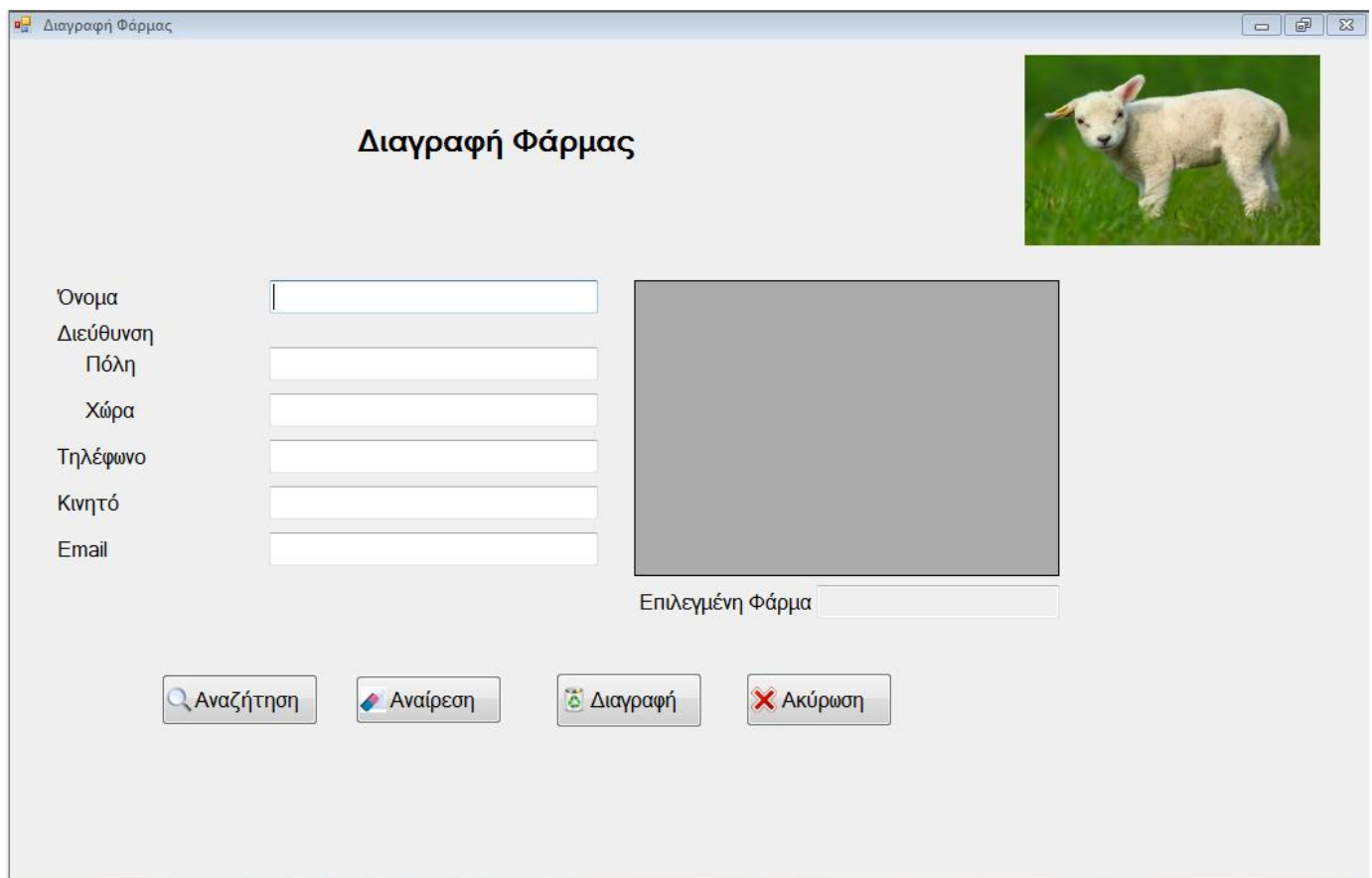
Ενημέρωση Αναίρεση Ακύρωση

Εικόνα A-6

Στην πιο πάνω εικόνα, παρουσιάζεται η φόρμα ενημέρωσης των στοιχείων μίας προϋπάρχουσας φάρμας του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία της φάρμας τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των στοιχείων

της συγκεκριμένης φάρμας στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

3.4 Διαγραφή

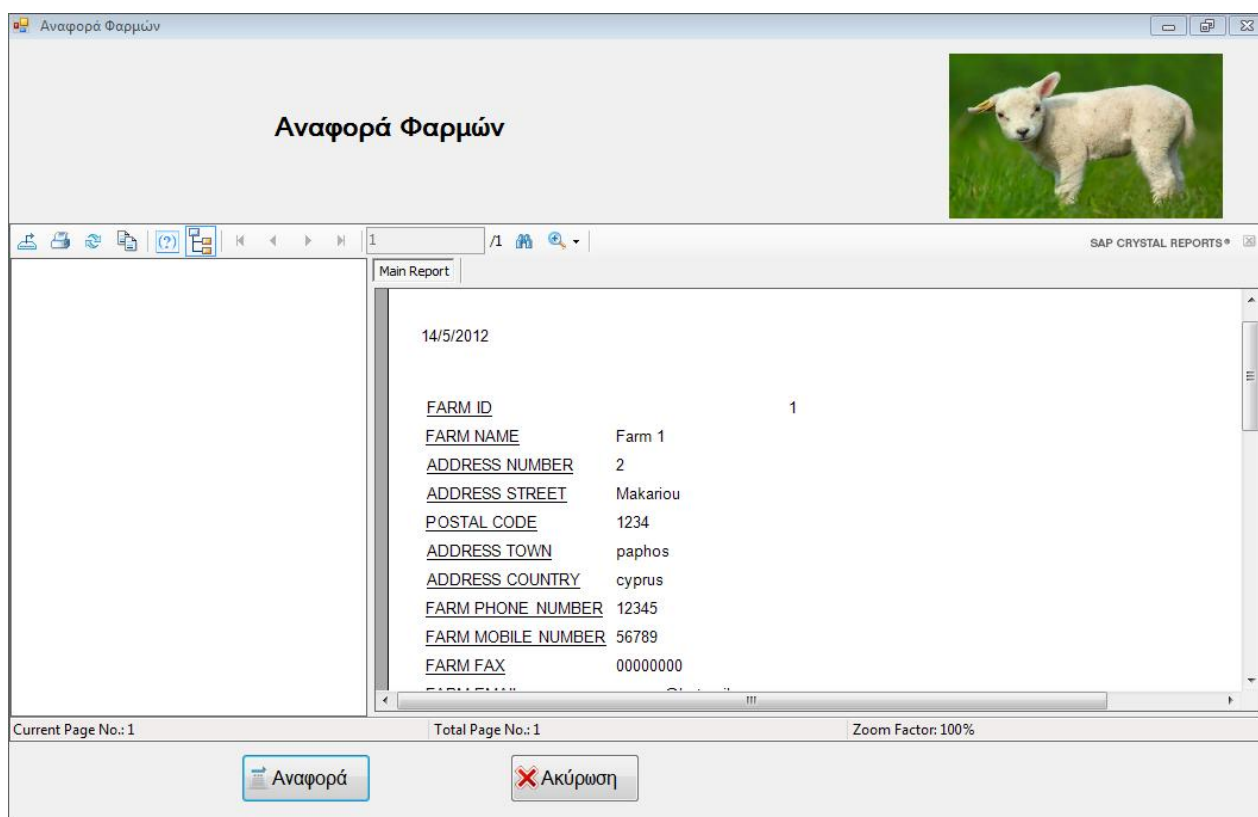


Εικόνα Α-7

Στην πιο πάνω εικόνα, παρουσιάζεται η διαδικασία διαγραφή κάποιας προϋπάρχουσας φάρμας από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις κτηνοτροφικές μονάδες. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που

εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων της συγκεκριμένης φάρμας-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

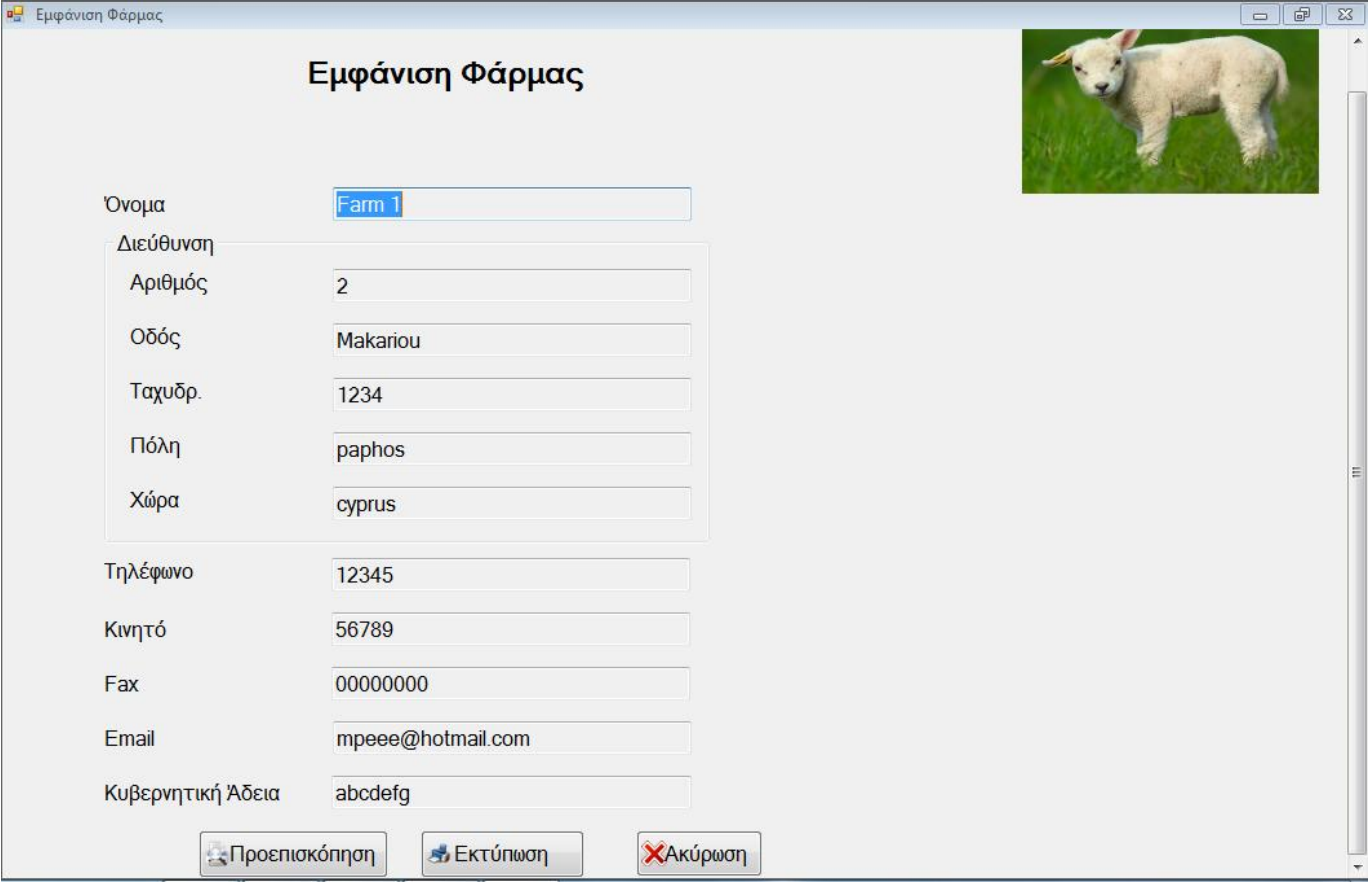
3.5 Αναφορά



Εικόνα A-8

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα αναφορών για τις φάρμες που είναι ήδη καταχωρημένες στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία από τις φάρμες πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

3.6 Εμφάνιση στοιχείων



Εμφάνιση Φάρμας

Όνομα Farm 1

Διεύθυνση

Αριθμός 2

Οδός Makariou

Ταχυδρ. 1234

Πόλη paphos

Χώρα cyprus

Τηλέφωνο 12345

Κινητό 56789

Fax 00000000

Email mpeeee@hotmail.com

Κυβερνητική Άδεια abcdefg

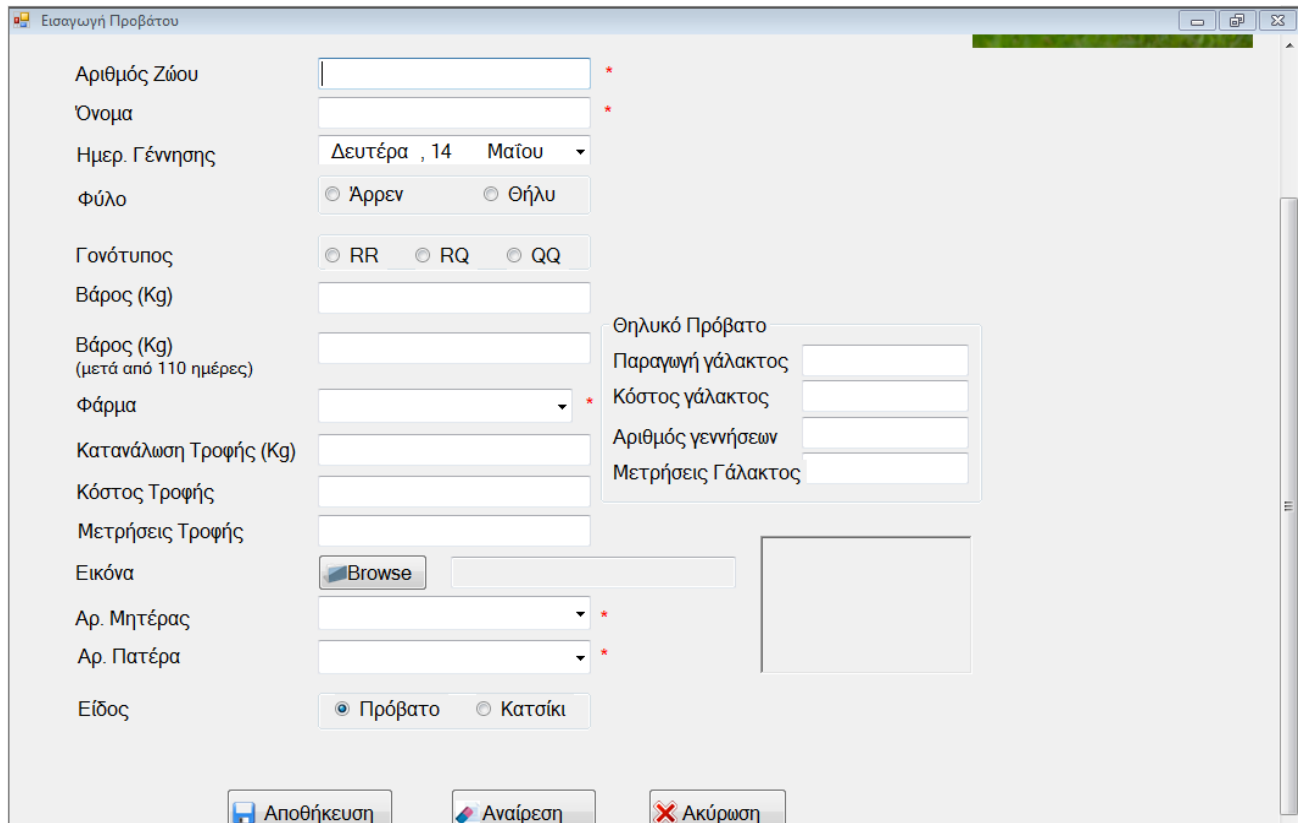
Προεπισκόπηση Εκτύπωση Ακύρωση

Εικόνα A-9

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα παρουσίασης των στοιχείων της φάρμας. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για την συγκεκριμένη φάρμα. Επιλέγοντας το κουμπί: Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

4. Κατηγορία Πρόβατο

4.1 Εισαγωγή προβάτου



Εισαγωγή Προβάτου

Αριθμός Ζώου *

Όνομα *

Ημερ. Γέννησης Δευτέρα , 14 Μαΐου

Φύλο ☐ Άρρεν ☐ Θήλυ

Γονότυπος ☐ RR ☐ RQ ☐ QQ

Βάρος (Kg)

Βάρος (Kg) (μετά από 110 ημέρες)

Φάρμα *

Κατανάλωση Τροφής (Kg)

Κόστος Τροφής

Μετρήσεις Τροφής

Εικόνα

Αρ. Μητέρας *

Αρ. Πατέρα *

Είδος ☒ Πρόβατο ☐ Κατσίκι

Θηλυκό Πρόβατο

Παραγωγή γάλακτος

Κόστος γάλακτος

Αριθμός γεννήσεων

Μετρήσεις Γάλακτος

Εικόνα A-10

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα εισαγωγής των στοιχείων ενός καινούργιου προβάτου στην βάση δεδομένων, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα SHEEP της βάσης. . Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να εισάγει τον αριθμό του ζώου ο οποίος δίνεται από τις κρατικές κτηνιατρικές υπηρεσίες, το όνομα του προβάτου, την φάρμα στην οποία ανήκει καθώς επίσης και τον αριθμό ζώου που έχουν η μητέρα και ο πατέρας του καινούργιου ζώου. Προαιρετικά, ο χρήστης μπορεί να εισάγει την ημερομηνία γέννησης, το φύλο και τον γονότυπο του ζώου, το βάρος του και το βάρος που έχει 110 ημέρες μετά την γέννηση του, το βάρος της τροφής που έχει καταναλώσει και το αντίστοιχο κόστος, καθώς και το πόσες μετρήσεις φαγητού έχουν γίνει για το συγκεκριμένο ζώο με βάση το οποίο προέκυψαν και οι προηγούμενοι αριθμοί. Ο χρήστης μπορεί μέσω της επιλογής του κουμπιού: Browse, κατά την οποία εμφανίζεται ένα καινούργιο παράθυρο πλοήγησης στα αρχεία του υπολογιστή, ο χρήστης μπορεί να επιλέξει την εικόνα η οποία αντιστοιχεί στο συγκεκριμένο ζώο, αν αυτή υπάρχει και το είδος του ζώου, δηλαδή αν είναι πρόβατο ή κατσίκι. Αν το πρόβατο είναι γένους

θηλυκού, τότε ο χρήστης μπορεί προαιρετικά να εισάγει την παραγωγή γάλακτος του, το κέρδος που είχε από την παραγωγή αυτή, τον αριθμό γεννήσεων που είχε το πρόβατο και τις αριθμό μετρήσεων που λήφθηκαν για την παραγωγή γάλακτος. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

4.2 Αναζήτηση Προβάτου

Κατά την διαδικασία αναζήτησης κάποιου προϋπάρχον προβάτου από την βάση δεδομένων, αναζητούνται τα στοιχεία των προβάτων τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα πρόβατα. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία-κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων του επιλεγμένου προβάτου, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία του συγκεκριμένου προβάτου και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

4.3 Ενημέρωση Προβάτου

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων ενός υπάρχον προβάτου του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία του προβάτου τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής ενός καινούργιου προβάτου. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των στοιχείων του συγκεκριμένου προβάτου στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

4.4 Διαγραφή Προβάτου

Διαγραφή Προβάτου

Αριθμός Ζώου

Όνομα

Ημερομηνία Δευτέρα, 14 Μαΐου ▼

Φύλο ☒ Άρρεν ☐ Θήλυ

Γονότυπος ☒ RR ☐ RQ ☐ QQ

Βάρος

Φάρμα

Είδος ☒ Πρόβατο ☐ Κατσίκι

Επιλεγμένο Πρόβατο

Αναζήτηση Αναίρεση

Κατάσταση ▼

Διαγραφή Ακύρωση

Εικόνα A-11

Στην πιο πάνω εικόνα, παρουσιάζεται η διαδικασία διαγραφής κάποιου υπάρχων προβάτου από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα πρόβατα. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια θα πρέπει να επιλέξει ένα από τα στοιχεία του πεδίου κατάσταση, το οποίο δηλώνει τον λόγο διαγραφής του προβάτου από την φόρμα. Έπειτα, μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων της συγκεκριμένης φόρμας-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

4.5 Αναφορά Προβάτων

Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τα πρόβατα που είναι ήδη καταχωρημένα στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των προβάτων πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

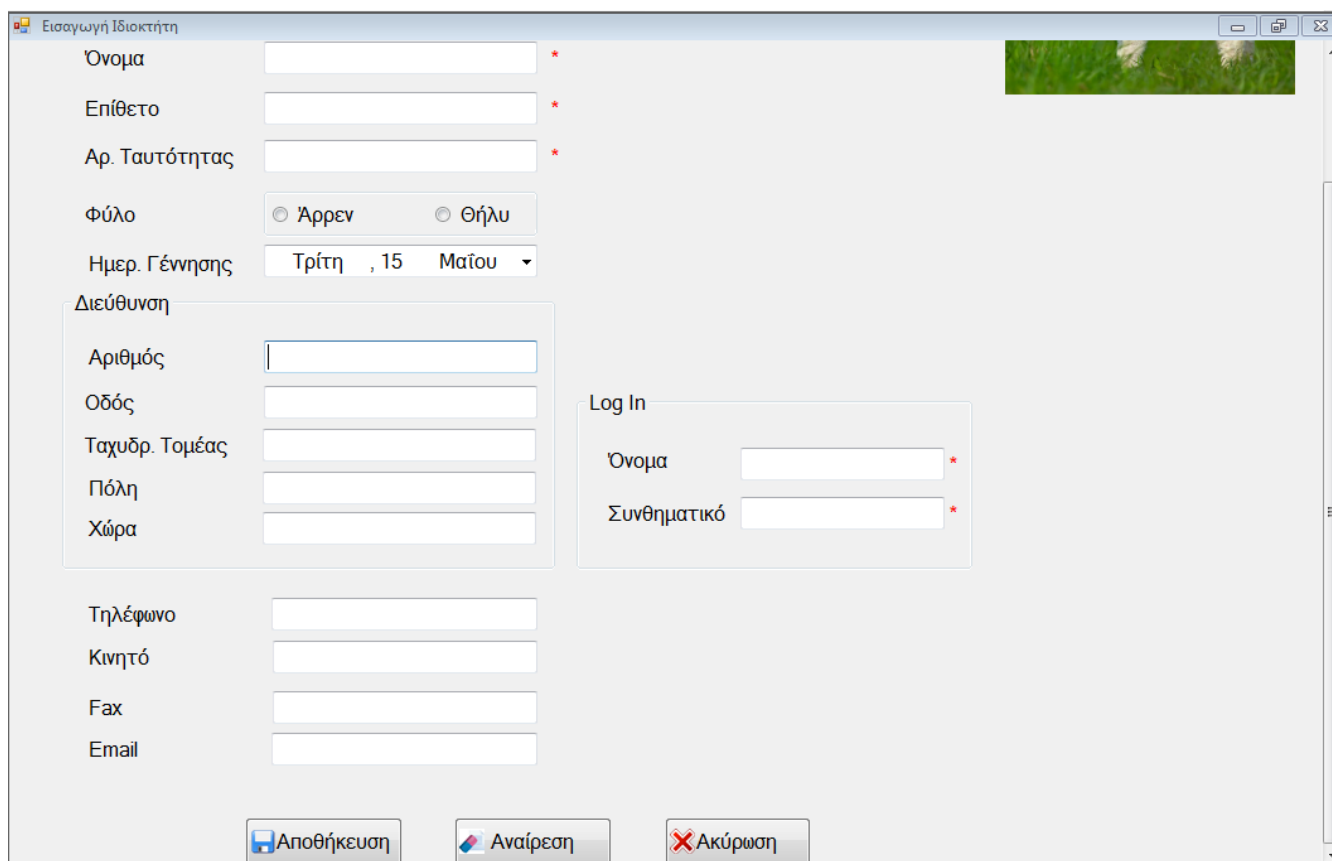
4.6 Παρουσίαση στοιχείων προβάτου

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για ένα συγκεκριμένο πρόβατο το οποίο έχει επιλεγθεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για το συγκεκριμένο πρόβατο. Επιλέγοντας το κουμπί:

Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

5. Κατηγορία Ιδιοκτήτης

5.1 Εισαγωγή ιδιοκτήτη



Εισαγωγή Ιδιοκτήτη

Όνομα *

Επίθετο *

Αρ. Ταυτότητας *

Φύλο ☐ Άρρεν ☐ Θήλυ

Ημερ. Γέννησης Τρίτη , 15 Μαΐου

Διεύθυνση

Αριθμός

Οδός

Ταχυδρ. Τομέας

Πόλη

Χώρα

Τηλέφωνο

Κινητό

Fax

Email

Log In

Όνομα *

Συνθηματικό *

Αποθήκευση Αναίρεση Ακύρωση

Εικόνα A-12

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Ιδιοκτήτης, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα OWNERS, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να εισάγει το όνομα, το επίθετο και τον αριθμό ταυτότητας του ιδιοκτήτη, το συνθηματικό όνομα και κωδικό με τα οποία θα μπορεί να εισέρχεται στο σύστημα. Προαιρετικά, ο χρήστης μπορεί να εισάγει την ημερομηνία γέννησης, το φύλο, την πλήρη διεύθυνση του ιδιοκτήτη (αριθμός, οδός, ταχυδρομικός τομέας, πόλη και χώρα), στοιχεία επικοινωνίας όπως τηλέφωνο, κινητό, Fax και e-mail. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των

στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

5.2 Αναζήτηση Ιδιοκτήτη

Κατά την διαδικασία αναζήτησης κάποιου προϋπάρχον ιδιοκτήτη από την βάση δεδομένων, αναζητούνται τα στοιχεία των ιδιοκτητών τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τους ιδιοκτήτες. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία-κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων του επιλεγμένου ιδιοκτήτη, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία του συγκεκριμένου ιδιοκτήτη και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

5.3 Ενημέρωση Ιδιοκτήτη

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων ενός υπάρχον ιδιοκτήτη του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία του ιδιοκτήτη τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα

στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής ενός καινούργιου ιδιοκτήτη. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των στοιχείων του συγκεκριμένου ιδιοκτήτη στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

5.4 Διαγραφή Ιδιοκτήτη

Η πιο πάνω λειτουργία αναφέρεται στην διαδικασία διαγραφής κάποιου υπάρχων ιδιοκτήτη από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τους ιδιοκτήτες. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία-κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων του συγκεκριμένου ιδιοκτήτη-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

5.5 Αναφορά Ιδιοκτήτη

Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τους ιδιοκτήτες που είναι ήδη καταχωρημένοι στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των

ιδιοκτητών πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

5.6 Παρουσίαση στοιχείων Ιδιοκτήτη

Εμφάνιση Ιδιοκτήτη

Όνομα: argyro

Επίθετο: kyriacou

Αρ. Ταυτότητας: 1231122

Φύλο: ☒ Άρρεν ☐ Θήλυ

Ημερ. Γέννησης: Τρίτη, 1 Μαΐου 2

Διεύθυνση

Αριθμός:

Οδός:

Ταχυδρ. Τομέας:

Πόλη:

Χώρα:

Log In

Όνομα: user

Συνθηματικό:

Τηλέφωνο:

Κινητό:

Fax:

Email:

Προεπισκόπηση Εκτύπωση Ακύρωση

Εικόνα A-13

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για ένα συγκεκριμένο ιδιοκτήτη ο οποίος έχει επιλεγθεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για το συγκεκριμένο ιδιοκτήτη. Επιλέγοντας το κουμπί: Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

6. Κατηγορία Υπάλληλος

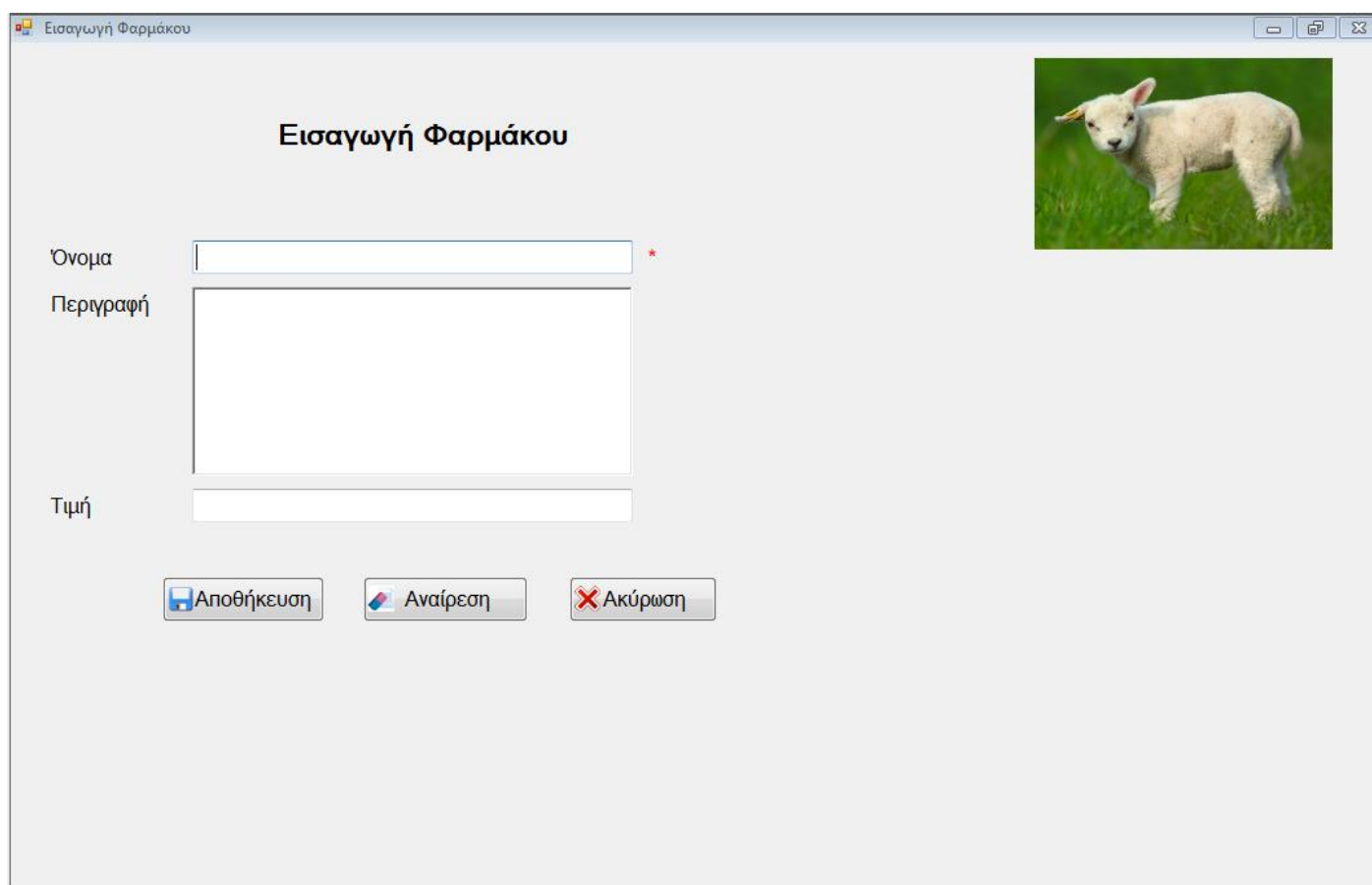
Οι λειτουργίες της εισαγωγής, αναζήτησης, ενημέρωσης, διαγραφής, αναφοράς και παρουσίασης στοιχείων είναι ίδιες με τις αντίστοιχες λειτουργίες της κατηγορίας Ιδιοκτήτης (παράγραφος 5).

7. Κατηγορία Κτηνίατρος

Οι λειτουργίες της εισαγωγής, αναζήτησης, ενημέρωσης, διαγραφής, αναφοράς και παρουσίασης στοιχείων είναι ίδιες με τις αντίστοιχες λειτουργίες της κατηγορίας Ιδιοκτήτης (παράγραφος 5).

8. Κατηγορία Φάρμακα

8.1 Εισαγωγή Φαρμάκου

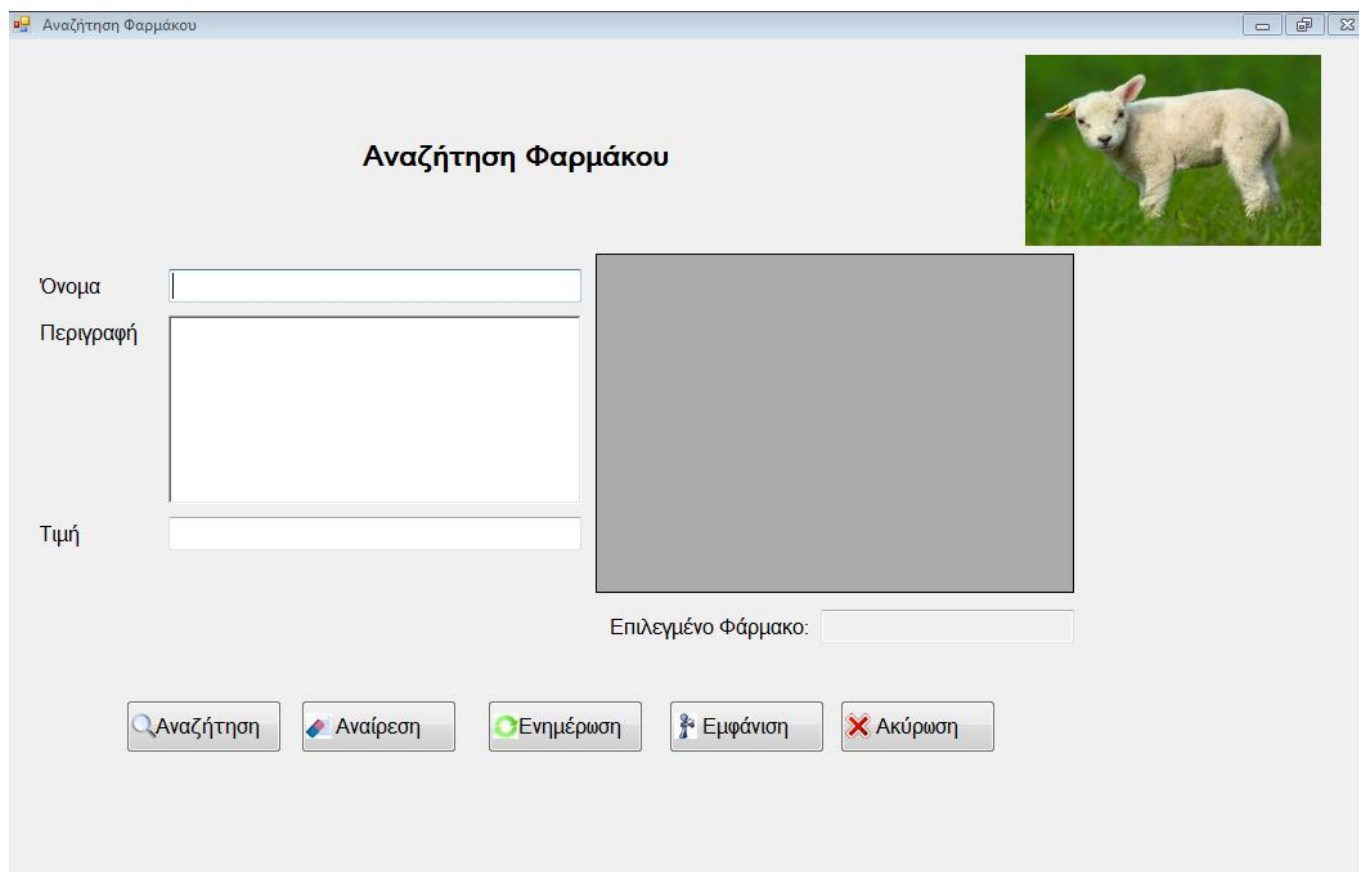


Εικόνα Α-14

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Φάρμακα, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα MEDICINES, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να εισάγει το όνομα του φαρμάκου. Προαιρετικά, ο

χρήστης μπορεί να εισάγει μία περιγραφή για το φάρμακο, καθώς και την τιμή του. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

8.2 Αναζήτηση Φαρμάκου



Εικόνα A-15

Κατά την διαδικασία αναζήτησης κάποιου προϋπάρχον φαρμάκου από την βάση δεδομένων, αναζητούνται τα στοιχεία των φαρμάκων τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα φάρμακα. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία-κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων του επιλεγμένου φαρμάκου, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία του συγκεκριμένου φαρμάκου και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

8.3 Ενημέρωση Φαρμάκου

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων ενός υπάρχον φαρμάκου του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία του φαρμάκου τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής ενός καινούργιου φαρμάκου. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των στοιχείων του συγκεκριμένου φαρμάκου στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

8.4 Αναφορά Φαρμάκων

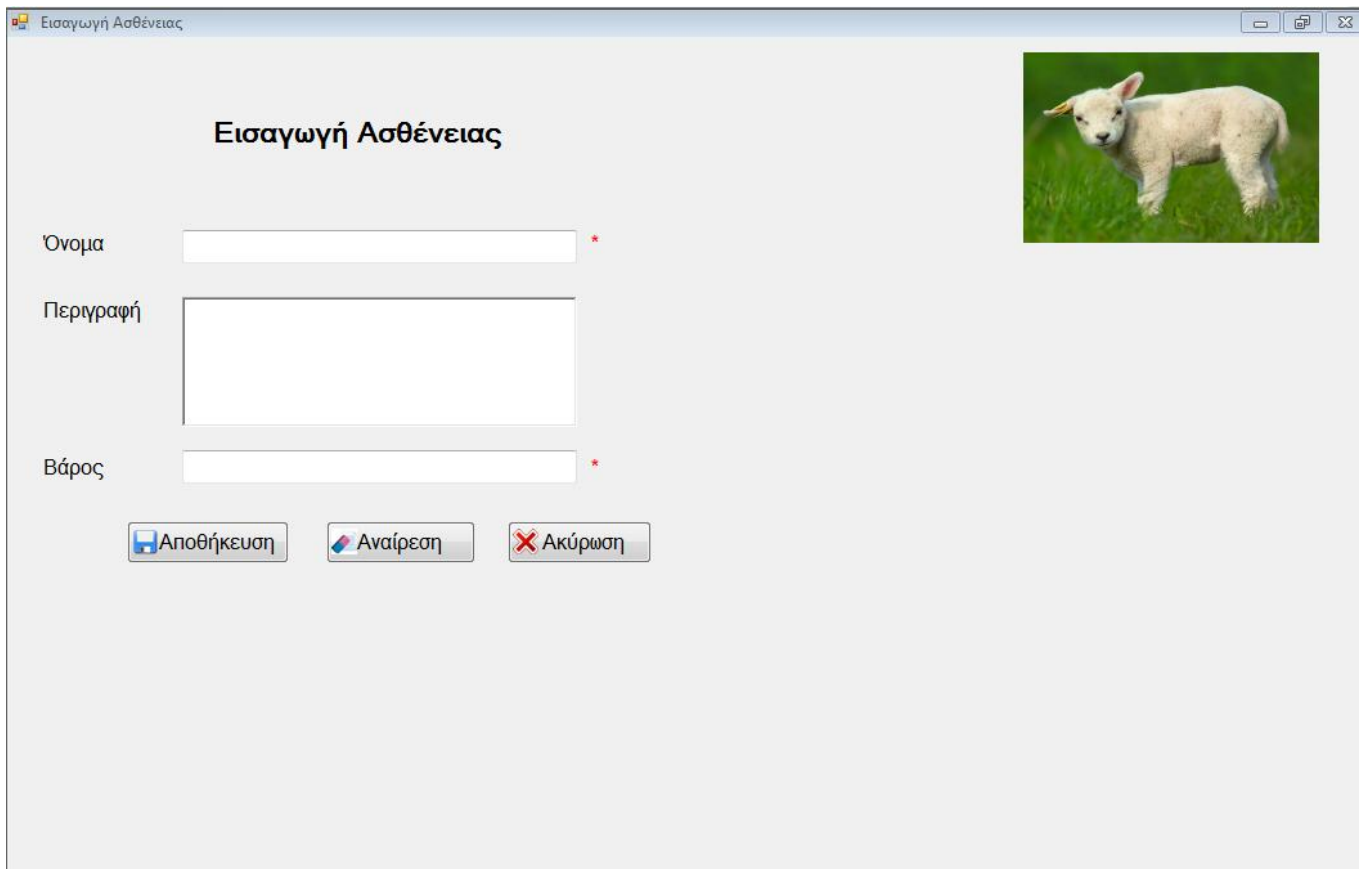
Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τα φάρμακα που είναι ήδη καταχωρημένα στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των φαρμάκων πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

8.5 Παρουσίαση στοιχείων Φαρμάκου

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για ένα συγκεκριμένο φάρμακο το οποίο έχει επιλεγθεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για το συγκεκριμένο φάρμακο. Επιλέγοντας το κουμπί: Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

9. Κατηγορία Ασθένεια

9.1 Εισαγωγή Ασθένειας



Εικόνα A-16

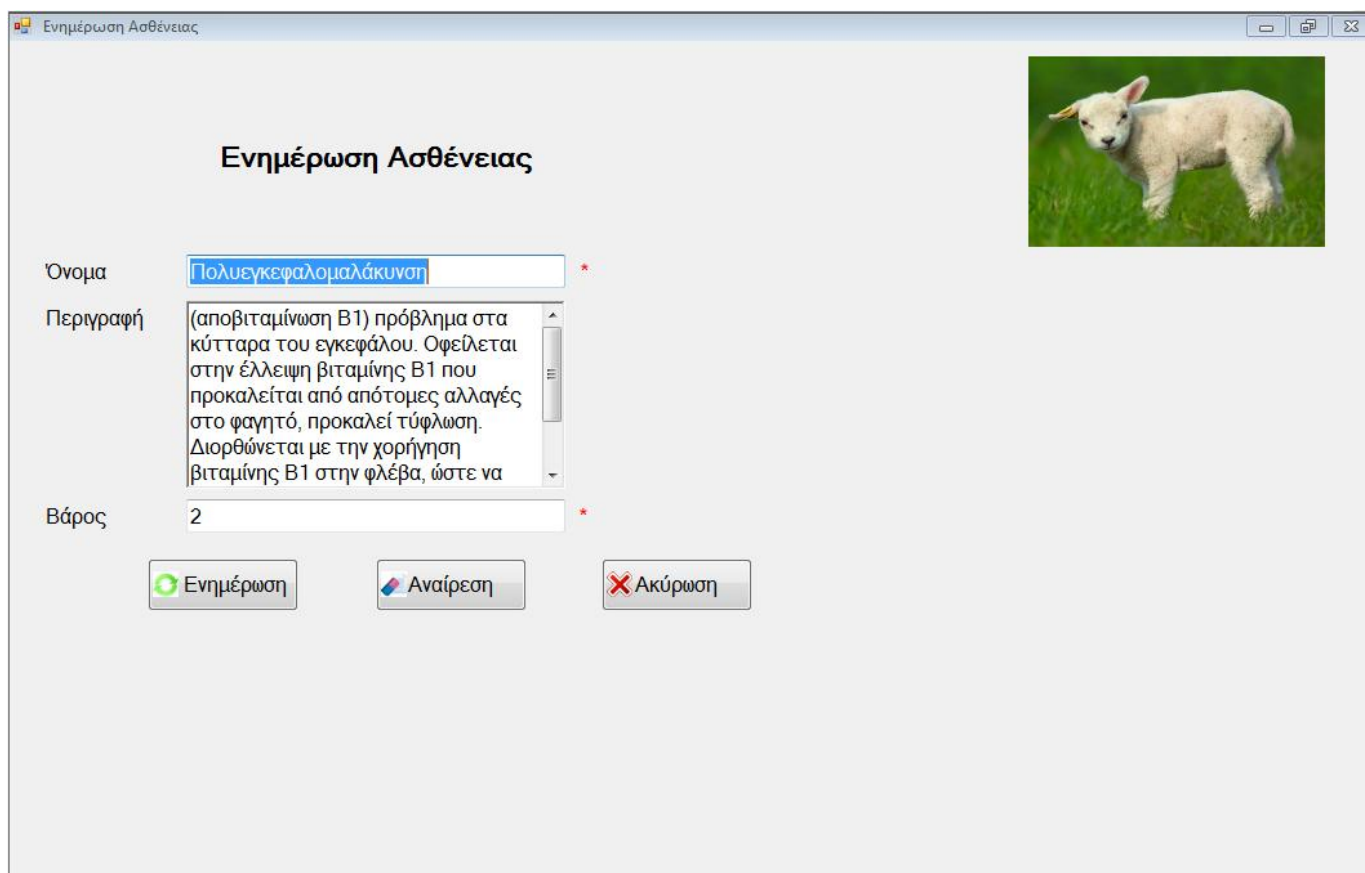
Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Ασθένεια, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα ILLNESS, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να εισάγει το όνομα της ασθένειας, καθώς επίσης και το βάρος που έχει, δηλαδή την σημαντικότητα της ασθένειας. Το στοιχείο αυτό είναι απαραίτητο καθώς θα χρησιμοποιηθεί αργότερα στην αξιολόγηση των προβάτων. Προαιρετικά, ο χρήστης μπορεί να εισάγει μία περιγραφή για την ασθένεια.. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση,

όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

9.2 Αναζήτηση Ασθένειας

Κατά την διαδικασία αναζήτησης κάποιας προϋπάρχον ασθένειας από την βάση δεδομένων, αναζητούνται τα στοιχεία των ασθενειών τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις ασθένειες. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία-κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων της επιλεγμένης ασθένειας, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία της συγκεκριμένης ασθένειας και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

9.3 Ενημέρωση Ασθένειας



Ενημέρωση Ασθένειας

Όνομα: Πολυεγκεφαλομαλάκυνση *

Περιγραφή: (αποβιταμίνωση B1) πρόβλημα στα κύτταρα του εγκεφάλου. Οφείλεται στην έλλειψη βιταμίνης B1 που προκαλείται από απότομες αλλαγές στο φαγητό, προκαλεί τύφλωση. Διορθώνεται με την χορήγηση βιταμίνης B1 στην φλέβα, ώστε να

Βάρος: 2 *

Ενημέρωση Αναίρεση Ακύρωση

Εικόνα A-17

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων μίας υπάρχουσας ασθένειας του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία της ασθένειας τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής μίας καινούργιας ασθένειας. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των της συγκεκριμένης ασθένειας στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

9.4 Αναφορά Ασθένειας

Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τις ασθένειες που είναι ήδη καταχωρημένες στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των

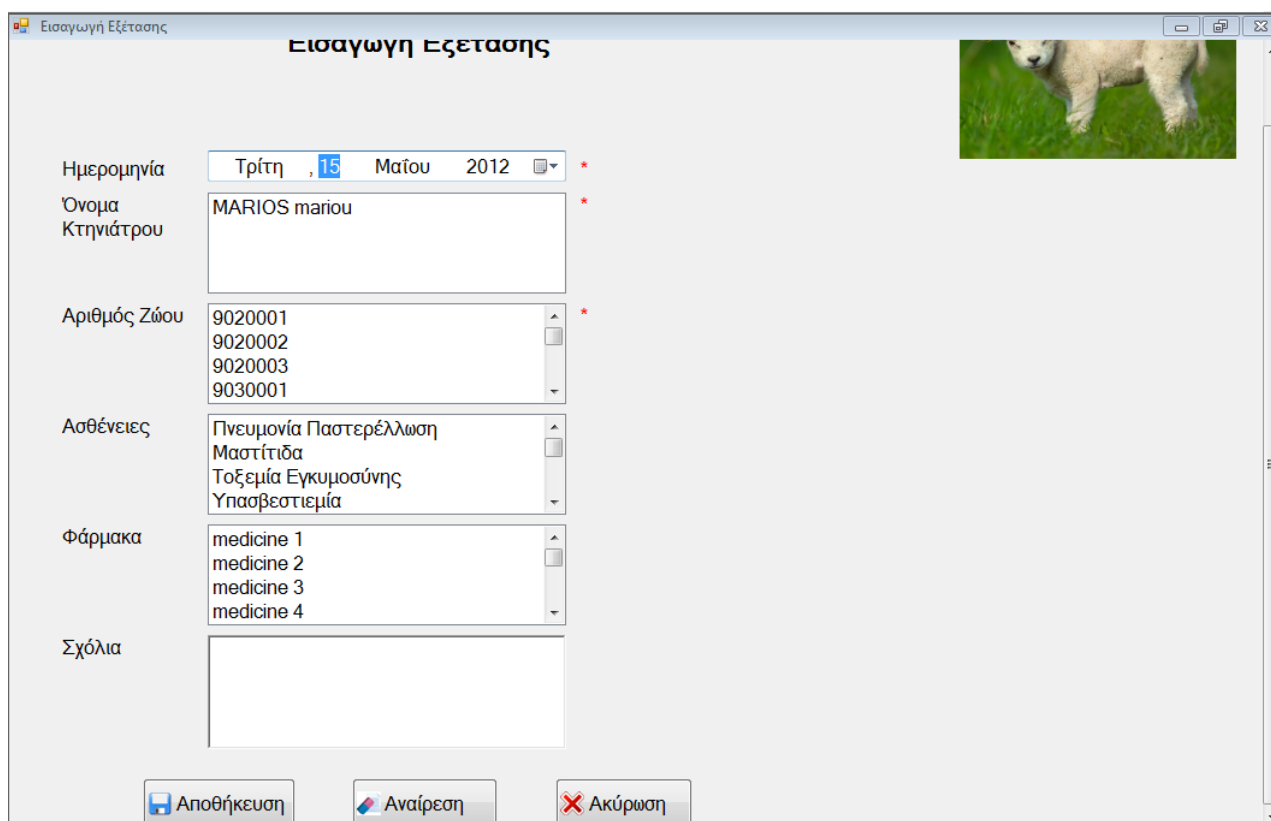
ασθενειών πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

9.5 Παρουσίαση στοιχείων Ασθένειας

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για μία συγκεκριμένη ασθένεια η οποία έχει επιλεγθεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για την συγκεκριμένη ασθένεια. Επιλέγοντας το κουμπί: Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

10. Κατηγορία Κτηνιατρικής Εξέτασης

10.1 Εισαγωγή Εξέτασης



Εισαγωγή Εξέτασης

Ημερομηνία: Τρίτη, 15 Μαΐου 2012 *

Όνομα Κτηνιάτρου: MARIOS mariou *

Αριθμός Ζώου: 9020001, 9020002, 9020003, 9030001 *

Ασθένειες: Πνευμονία Παστερέλλωση, Μαστίτιδα, Τοξεμία Εγκυμοσύνης, Υπασβεστιεμία

Φάρμακα: medicine 1, medicine 2, medicine 3, medicine 4

Σχόλια

Αποθήκευση, Αναίρεση, Ακύρωση

Εικόνα A-18

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Εξέταση, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα EXAMINATION, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να εισάγει την ημερομηνία της εξέτασης, το όνομα του κτηνιάτρου, καθώς και τον αριθμό του ζώου στο οποίο έγινε η εξέταση. Προαιρετικά, ο χρήστης μπορεί να επιλέξει από την λίστα των ασθενειών, τις ασθένειες οι οποίες διαγνώσθηκαν στο συγκεκριμένο πρόβατο, καθώς και να επιλέξει από την λίστα των φαρμάκων, τα φάρμακα τα οποία χορηγήθηκαν στο πρόβατο. Τέλος, ο χρήστης έχει την δυνατότητα να εισάγει ένα σχόλιο για την εξέταση. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε εμφανίζονται τα κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

10.2 Αναζήτηση Εξέτασης

Κατά την διαδικασία αναζήτησης κάποιας προϋπάρχον εξέτασης από την βάση δεδομένων, αναζητούνται τα στοιχεία των εξετάσεων τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις κτηνιατρικές εξετάσεις. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα

αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων της επιλεγμένης εξέτασης, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία της συγκεκριμένης εξέτασης και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

10.3 Ενημέρωση Εξέτασης

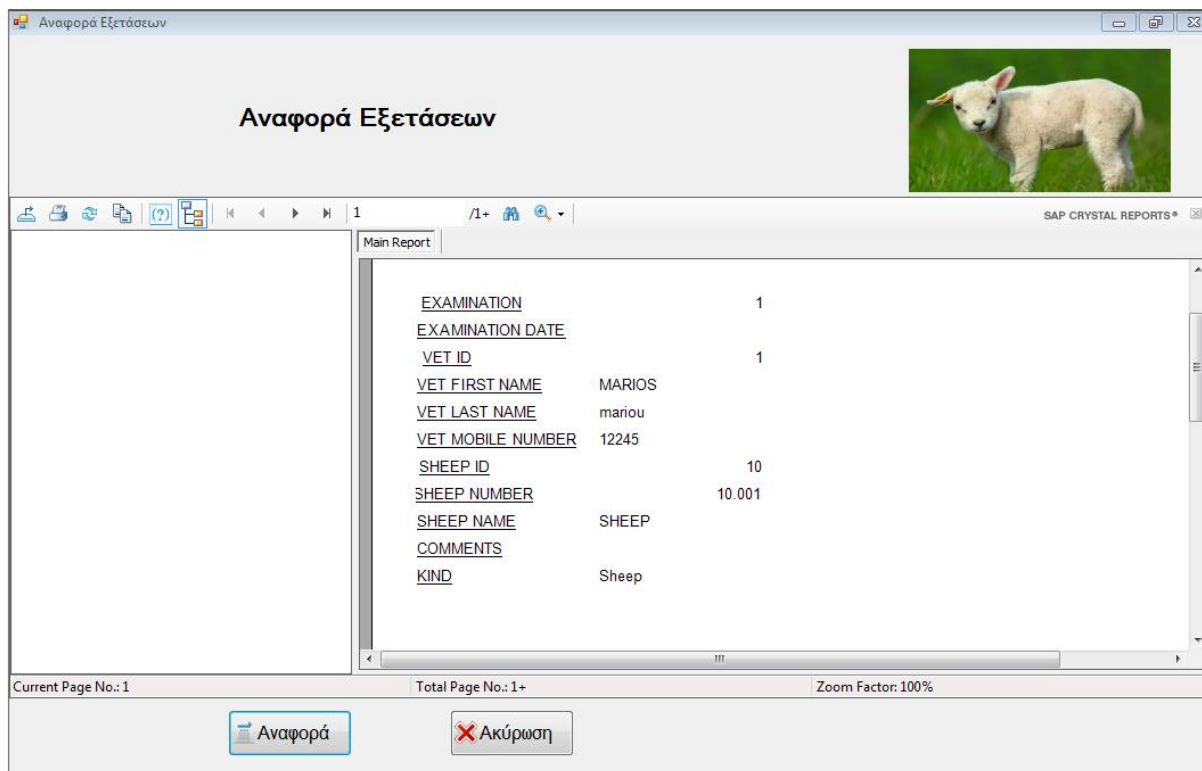
Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων μίας υπάρχουσας εξέτασης του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία της εξέτασης τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής μίας καινούργιας εξέτασης. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των της συγκεκριμένης εξέτασης στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

10.4 Διαγραφή Εξέτασης

Η πιο πάνω λειτουργία αναφέρεται στην διαδικασία διαγραφής κάποιας υπάρχουσας εξέτασης από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις κτηνιατρικές εξετάσεις. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα

αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων της συγκεκριμένης εξέτασης-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

10.5 Αναφορά Εξετάσεων



Αναφορά Εξετάσεων

EXAMINATION	1
EXAMINATION DATE	
VET ID	1
VET FIRST NAME	MARIOS
VET LAST NAME	mariou
VET MOBILE NUMBER	12245
SHEEP ID	10
SHEEP NUMBER	10.001
SHEEP NAME	SHEEP
COMMENTS	
KIND	Sheep

Current Page No.: 1 Total Page No.: 1+ Zoom Factor: 100%

Εικόνα A-19

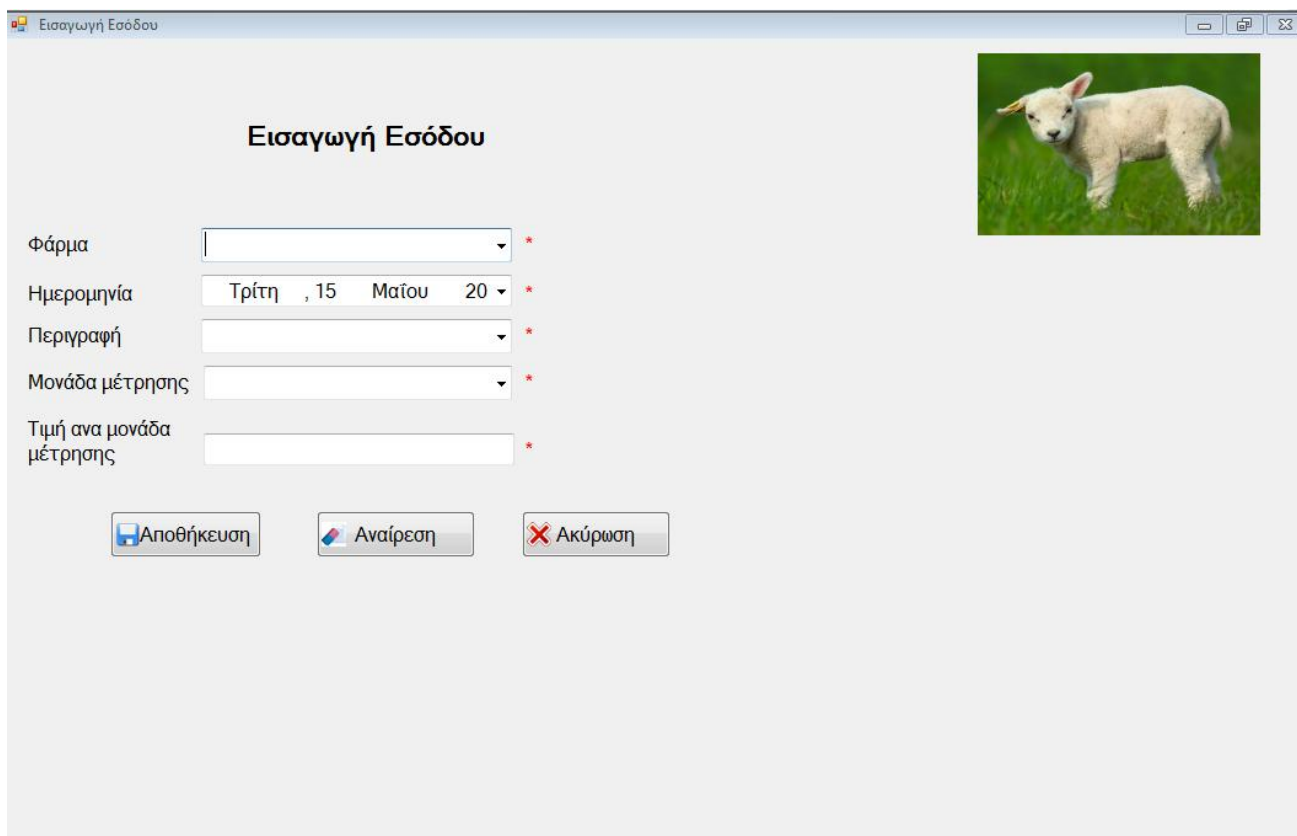
Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τις εξετάσεις που είναι ήδη καταχωρημένες στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των εξετάσεων πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

10.6 Παρουσίαση στοιχείων Εξέτασης

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για μία συγκεκριμένη κτηνιατρική εξέταση η οποία έχει επιλεγθεί από την διαδικασία της

αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για την συγκεκριμένη εξέταση. Επιλέγοντας το κουμπί: Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

11. Κατηγορία Έσοδο



Εικόνα A-20

11.1 Εισαγωγή Εσόδου

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Έσοδο, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα INCOMING, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να επιλέξει την φάρμα στην οποία αναφέρεται το συγκεκριμένο έσοδο, την ημερομηνία εισαγωγής του, την περιγραφή (Γάλα, Κρέας, Τροφή, Σανός, Επιχορήγηση, Μαλλί, Κοπριά) και την μονάδα μέτρησης του, και ασφαλώς την τιμή ανά μονάδα μέτρησης. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το

πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

11.2 Αναζήτηση Εσόδου

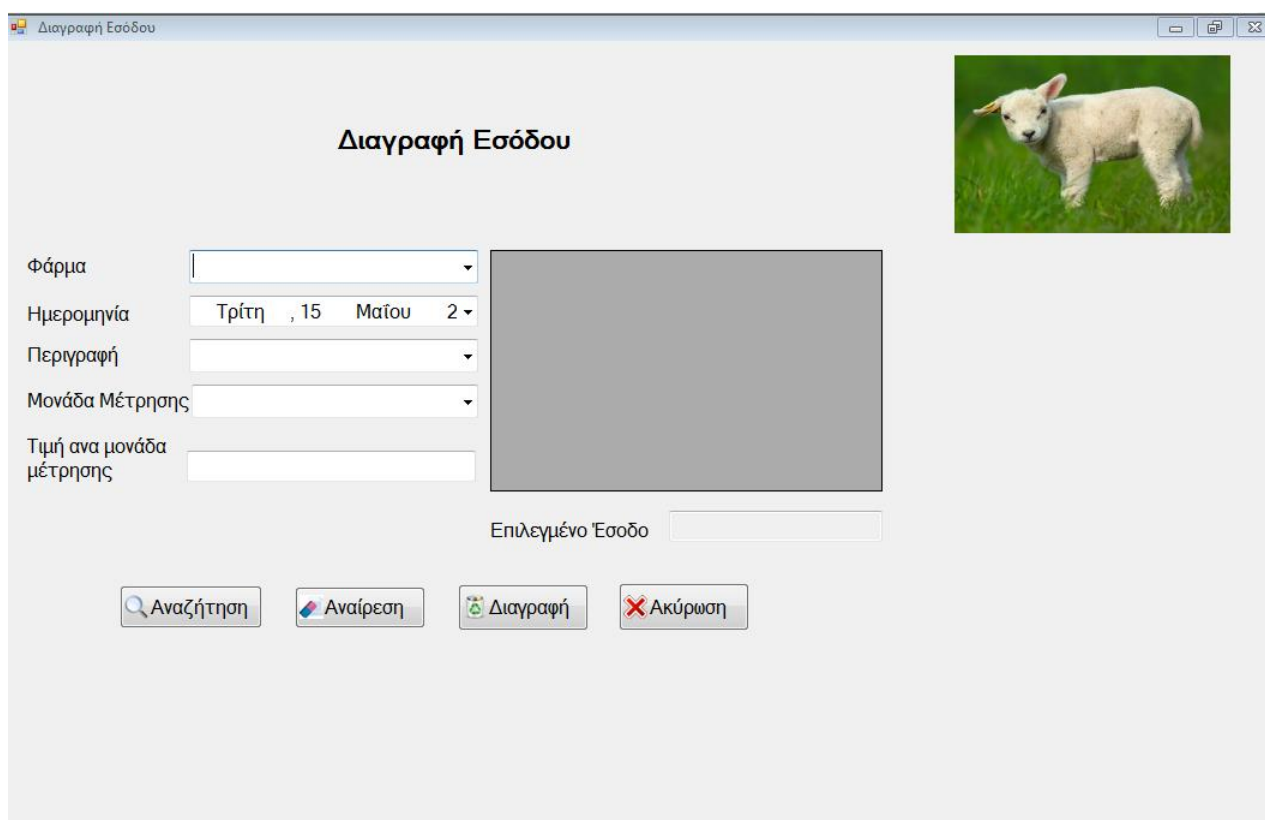
Κατά την διαδικασία αναζήτησης κάποιου προϋπάρχων εσόδου από την βάση δεδομένων, αναζητούνται τα στοιχεία των εσόδων τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα έσοδα. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία-κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων του επιλεγμένου εσόδου, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία του συγκεκριμένου εσόδου και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

11.3 Ενημέρωση Εσόδου

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων ενός υπάρχον εσόδου του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία του εσόδου τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα

στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής ενός καινούργιου εσόδου. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των στοιχείων του συγκεκριμένου εσόδου στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

11.4 Διαγραφή



Εικόνα A-21

Η πιο πάνω λειτουργία αναφέρεται στην διαδικασία διαγραφής κάποιου υπάρχον εσόδου από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα έσοδα. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων του συγκεκριμένου εσόδου-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

11.5 Αναφορά Εσόδων

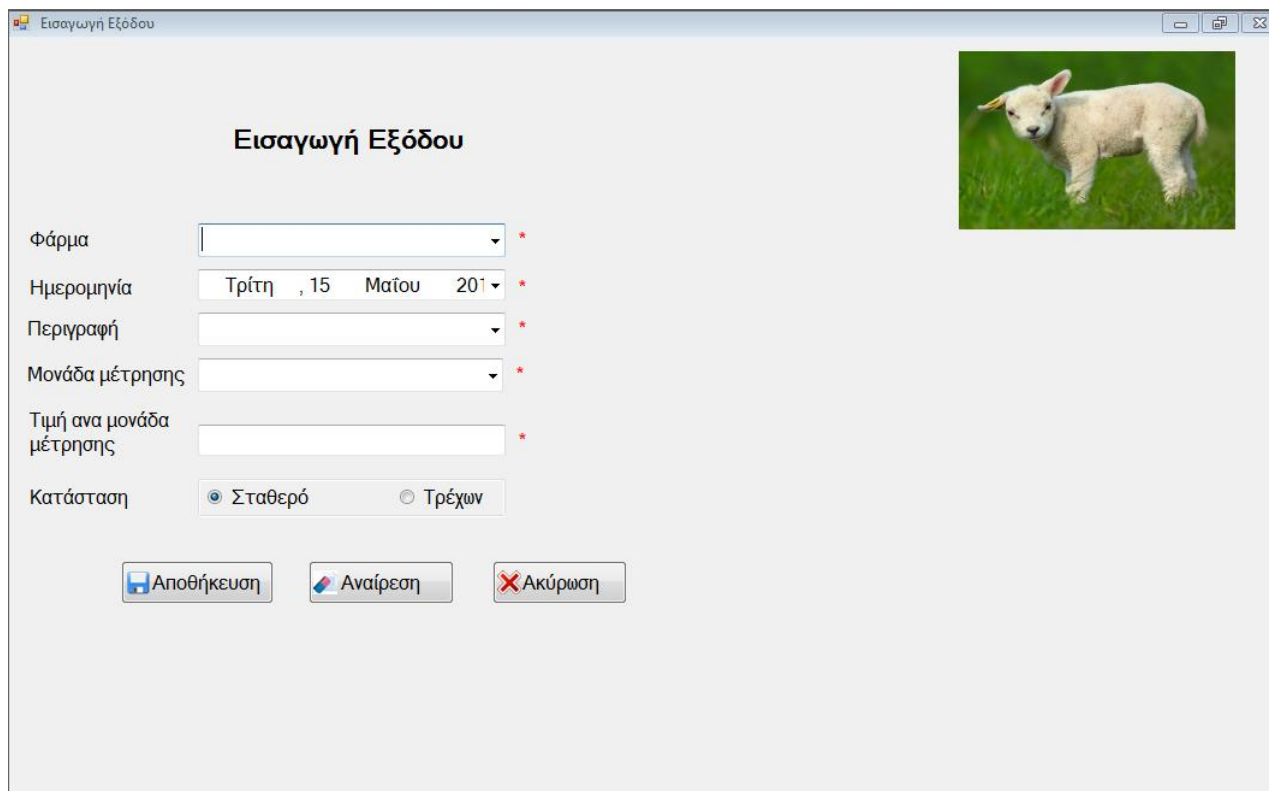
Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τα έσοδα που είναι ήδη καταχωρημένα στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των εσόδων πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

11.6 Παρουσίαση στοιχείων Εσόδου

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για ένα συγκεκριμένο έσοδο το οποίο έχει επιλεγεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για το συγκεκριμένο έσοδο. Επιλέγοντας το κουμπί: Προεπισκόπηση, ο χρήστης μπορεί να δει πώς θα είναι το αρχείο που πρόκειται να εκτυπώσει αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

12. Κατηγορία Έξοδο

12.1 Εισαγωγή Εξόδου



Εισαγωγή Εξόδου

Φάρμα *

Ημερομηνία Τρίτη, 15 Μαΐου 201 *

Περιγραφή *

Μονάδα μέτρησης *

Τιμή ανα μονάδα μέτρησης *

Κατάσταση ☒ Σταθερό ☐ Τρέχων

Εικόνα A-22

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Έξοδο, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα OUTGOING, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να επιλέξει την φάρμα στην οποία αναφέρεται το συγκεκριμένο έξοδο, την ημερομηνία εισαγωγής του, την περιγραφή (Κατασκευή και Υπηρεσίες φάρμας, μισθοί υπαλλήλων, ενοίκιο, πετρέλαιο, κτηνιατρικά έξοδα ..) και την μονάδα μέτρησης του, ασφαλώς την τιμή ανά μονάδα μέτρησης, και τέλος, την κατάσταση του εξόδου. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε εμφανίζονται τα κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την

επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

12.2 Αναζήτηση Εξόδου

Η αναζήτηση Εξόδου ακολουθεί ακριβώς την ίδια διαδικασία με την λειτουργία Αναζήτηση Εσόδου της Κατηγορίας: Έσοδο (παράγραφος 11.2), με την μόνη διαφορά ότι η περιγραφή θα περιλαμβάνει τα έξοδα και όχι τα έσοδα.

12.3 Ενημέρωση Εξόδου

Η Ενημέρωση Εξόδου ακολουθεί ακριβώς την ίδια διαδικασία με την λειτουργία Ενημέρωση Εσόδου της Κατηγορίας: Έσοδο (παράγραφος 11.3), με την μόνη διαφορά ότι η περιγραφή θα περιλαμβάνει τα έξοδα και όχι τα έσοδα.

12.4 Διαγραφή Εξόδου

Η Διαγραφή Εξόδου ακολουθεί ακριβώς την ίδια διαδικασία με την λειτουργία Διαγραφή Εσόδου της Κατηγορίας: Έσοδο (παράγραφος 11.4), με την μόνη διαφορά ότι η περιγραφή θα περιλαμβάνει τα έξοδα και όχι τα έσοδα.

12.5 Αναφορά Εξόδων

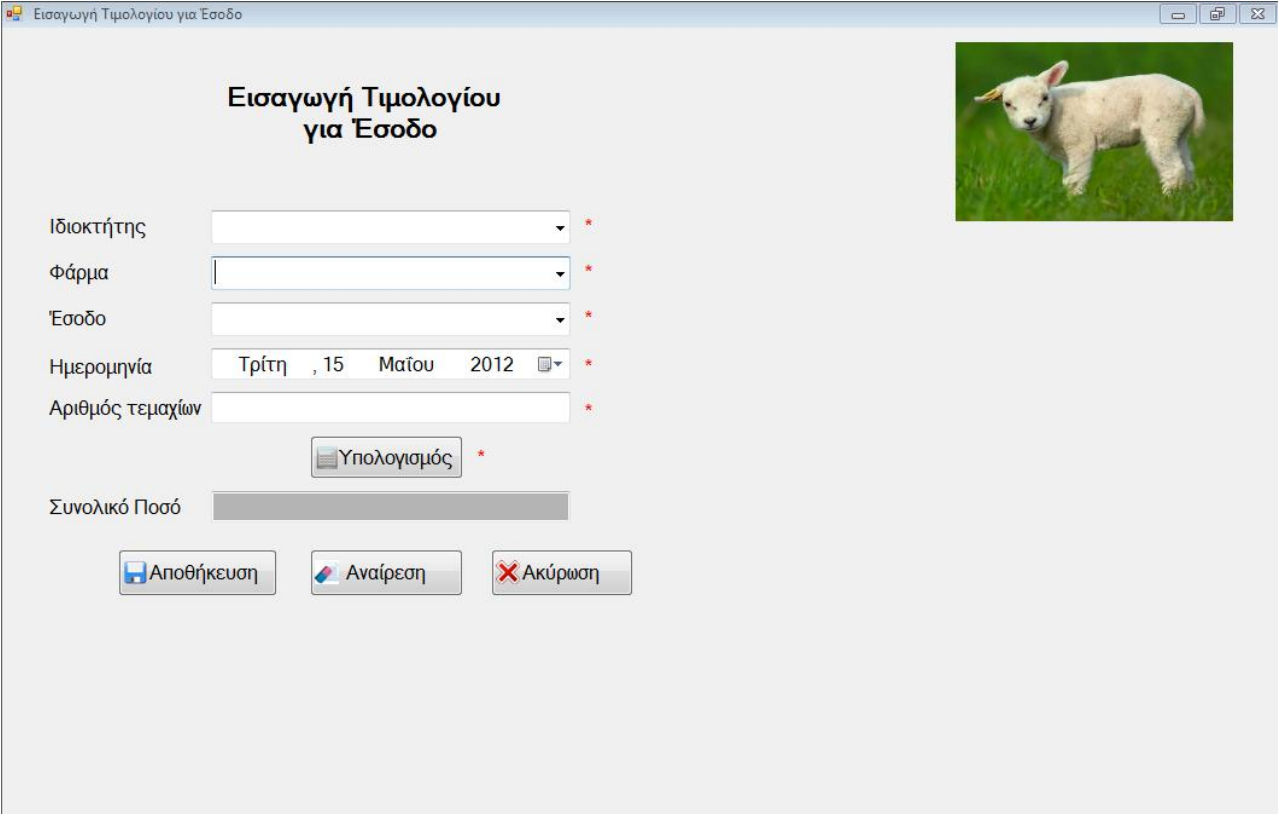
Η Αναφορά Εξόδων ακολουθεί ακριβώς την ίδια διαδικασία με την λειτουργία Αναφορά Εσόδων της Κατηγορίας: Έσοδο (παράγραφος 11.5), με την μόνη διαφορά ότι η περιγραφή θα περιλαμβάνει τα έξοδα και όχι τα έσοδα.

12.6 Παρουσίαση στοιχείων Εξόδου

Η Παρουσίαση στοιχείων Εξόδου ακολουθεί ακριβώς την ίδια διαδικασία με την λειτουργία Παρουσίαση στοιχείων Εσόδου της Κατηγορίας: Έσοδο (παράγραφος 11.6), με την μόνη διαφορά ότι η περιγραφή θα περιλαμβάνει τα έξοδα και όχι τα έσοδα.

13. Κατηγορία Τιμολόγιο Εσόδων

13.1 Εισαγωγή Τιμολογίου Εσόδου



**Εισαγωγή Τιμολογίου
για Έσοδο**

Ιδιοκτήτης *

Φάρμα *

Έσοδο *

Ημερομηνία Τρίτη , 15 Μαΐου 2012 *

Αριθμός τεμαχίων *

*

Συνολικό Ποσό

Εικόνα Α-23

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Τιμολόγιο Εσόδου, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα INVOICE_IN, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να επιλέξει τον ιδιοκτήτη που εξέδωσε το τιμολόγιο, την φάρμα στην οποία αναφέρεται το συγκεκριμένο τιμολόγιο, την ημερομηνία εισαγωγής του, το Έσοδο το οποίο αφορά και τον αριθμό τεμαχίων του συγκεκριμένου εσόδου. Ο χρήστης έπειτα θα πρέπει να επιλέξει το κουμπί: Υπολογισμός, έτσι ώστε να υπολογισθεί αυτόματα το συνολικό ποσό του τιμολογίου που αναφέρεται σε ένα συγκεκριμένο έσοδο. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε τα εμφανίζονται κατάλληλα μηνύματα στον χρήστη για το που βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που

εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

13.2 Αναζήτηση Τιμολογίου Εσόδου

Κατά την διαδικασία αναζήτησης κάποιου προϋπάρχον τιμολογίου εσόδου από την βάση δεδομένων, αναζητούνται τα στοιχεία των τιμολογίων εσόδων τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα τιμολόγια εσόδων. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων του επιλεγμένου τιμολογίου, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία του συγκεκριμένου τιμολογίου και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

13.3 Ενημέρωση Τιμολογίου Εσόδου

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων ενός υπάρχον τιμολογίου εσόδου του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία του τιμολογίου τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής ενός καινούργιου τιμολογίου εσόδου. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εκτελείται ενημέρωση των στοιχείων του συγκεκριμένου τιμολογίου

στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

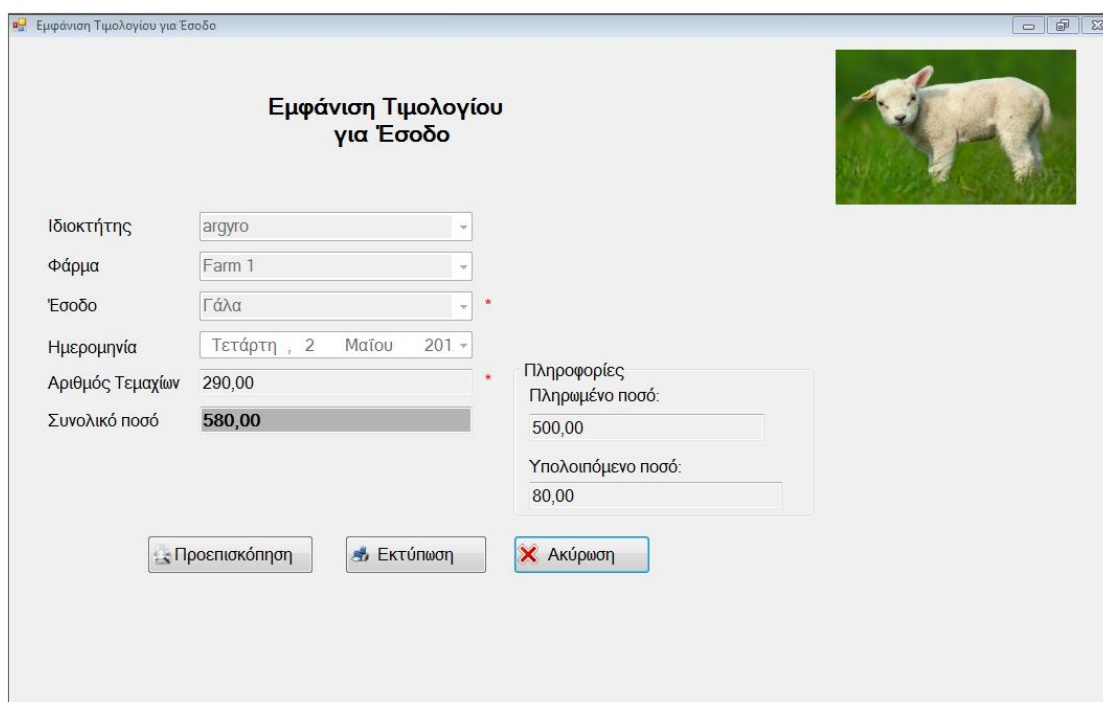
13.4 Διαγραφή Τιμολογίου Εσόδου

Η πιο πάνω λειτουργία αναφέρεται στην διαδικασία διαγραφής κάποιου υπάρχον τιμολογίου εσόδου από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τα τιμολόγια εσόδων. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων του συγκεκριμένου τιμολογίου εσόδου-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

13.5 Αναφορά Τιμολογίων Εσόδων

Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τα τιμολόγια εσόδων που είναι ήδη καταχωρημένα στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των εσόδων πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

13.6 Παρουσίαση στοιχείων Τιμολογίου Εσόδου



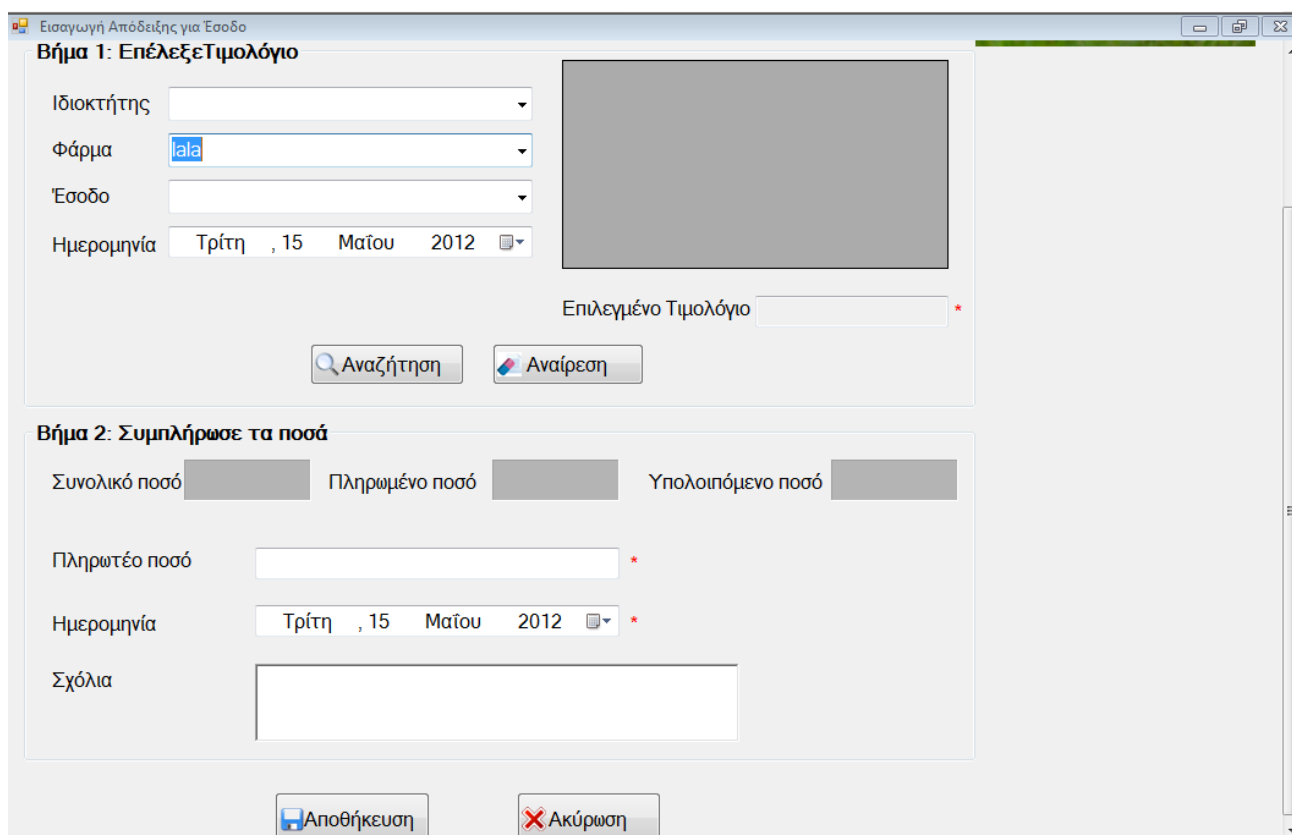
Εικόνα A-24

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για ένα συγκεκριμένο τιμολόγιο εσόδου το οποίο έχει επιλεγθεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για το συγκεκριμένο τιμολόγιο. Επιλέγοντας το κουμπί: Προεπισκόπηση, ανοίγει ένα αρχείο Microsoft Word 2007 στο οποίο ο χρήστης μπορεί να κάνει αλλαγές και να το τυπώσει. Επιπλέον, το Word αρχείο αποθηκεύεται αυτόματα σε συγκεκριμένο φάκελο του υπολογιστή. Ο χρήστης μπορεί να εκτυπώσει την φόρμα παρουσίασης του τιμολογίου όπως αυτό εμφανίζεται από την εφαρμογή, αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

14. Κατηγορία Τιμολόγιο Εξόδου

Οι λειτουργίες της κατηγορίας αυτή είναι ακριβώς ίδιες με τις αντίστοιχες λειτουργίες της κατηγορίας: Τιμολόγιο Εσόδου (παράγραφος 13), με την μόνη διαφορά ότι η κατηγορία αυτή αναφέρεται στα Τιμολόγια Εξόδων και όχι Εσόδων.

15. Κατηγορία Απόδειξη Εσόδου



Εισαγωγή Απόδειξης για Έσοδο

Βήμα 1: Επέλεξε Τιμολόγιο

Ιδιοκτήτης

Φάρμα

Έσοδο

Ημερομηνία

Επιλεγμένο Τιμολόγιο

Βήμα 2: Συμπλήρωσε τα ποσά

Συνολικό ποσό Πληρωμένο ποσό Υπολοιπόμενο ποσό

Πληρωτέο ποσό

Ημερομηνία

Σχόλια

Εικόνα Α-25

15.1 Εισαγωγή Απόδειξης Εσόδου

Στην πιο πάνω εικόνα παρουσιάζεται η εισαγωγή νέων στοιχείων για την κατηγορία Απόδειξη Εσόδου, δηλαδή είναι η διαδικασία εισαγωγής μίας καινούργιας εγγραφής στον πίνακα RECEIPT_IN, της βάσης δεδομένων. Συγκεκριμένα, στην φόρμα αυτή ο χρήστης υποχρεωτικά θα πρέπει να επιλέξει το τιμολόγιο εσόδου το οποίο σχετίζεται με την απόδειξη. Το τιμολόγιο μπορεί να το βρει μέσω της αναζήτησης τιμολογίου που υπάρχει στην πιο πάνω φόρμα. Επιπλέον, υποχρεωτικά πεδία είναι το πληρωτέο ποσό και η ημερομηνία έκδοσης της απόδειξης. Ο χρήστης προαιρετικά μπορεί να εισάγει σχόλια σχετικά με την συγκεκριμένη απόδειξη εσόδου. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αποθήκευση, κατά το οποίο γίνεται έλεγχος για την ορθότητα των στοιχείων που καταχωρήθηκαν, και αν ο έλεγχος είναι ορθός τότε τα στοιχεία για την δημιουργία την νέας εγγραφής εισάγονται στην βάση δεδομένων. Αν ο έλεγχος είναι λανθασμένος, τότε εμφανίζονται τα κατάλληλα μηνύματα στον χρήστη για το που

βρίσκονται τα λάθη και ο χρήστης παραμένει στην ίδια φόρμα. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία εισαγωγή των στοιχείων στην βάση δεδομένων, και η φόρμα τερματίζεται.

15.2 Αναζήτηση Απόδειξης Εσόδου

Κατά την διαδικασία αναζήτησης κάποιας προϋπάρχουσας απόδειξης εσόδου από την βάση δεδομένων, αναζητούνται τα στοιχεία των αποδείξεων εσόδων τα οποία πληρούν τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις αποδείξεις εσόδων. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Ενημέρωση, κατά το οποίο εμφανίζεται η φόρμα ενημέρωσης των στοιχείων της επιλεγμένης απόδειξης, ή μπορεί να επιλέξει το κουμπί Εμφάνιση, κατά το οποίο εμφανίζονται τα στοιχεία της συγκεκριμένης απόδειξης και τα οποία μπορεί ο χρήστης να τυπώσει. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

15.3 Ενημέρωση Απόδειξης Εσόδου

Η λειτουργία αυτή αφορά την φόρμα ενημέρωσης των στοιχείων μίας υπάρχουσας απόδειξης εσόδου του συστήματος. Κατά την είσοδο στην φόρμα αυτή, παρουσιάζονται τα στοιχεία της απόδειξης τα οποία είναι ήδη καταχωρημένα στην Βάση δεδομένων. Τα στοιχεία αυτά είναι τα ίδια στοιχεία τα οποία εμφανίζονται στην φόρμα εισαγωγής μίας καινούργιας απόδειξης εσόδου. Ο χρήστης μπορεί να ενημερώσει τα στοιχεία τα οποία επιθυμεί και έπειτα να επιλέξει το κουμπί Ενημέρωση,

κατά το οποίο εκτελείται ενημέρωση των στοιχείων της συγκεκριμένης απόδειξης στην βάση δεδομένων. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

15.4 Διαγραφή Απόδειξης Εσόδου

Εικόνα Α-26

Η πιο πάνω λειτουργία αναφέρεται στην διαδικασία διαγραφής κάποιας υπάρχουσας απόδειξης εσόδου από την βάση δεδομένων, με βάση τα κριτήρια τα οποία εμφανίζονται στα αριστερά της φόρμας. Τα αποτελέσματα της αναζήτησης, εμφανίζονται στον πίνακα που βρίσκεται στα δεξιά της φόρμας. Αν ο χρήστης δεν εισάγει κανένα από τα κριτήρια με το οποίο θα γίνει αναζήτηση, τότε εμφανίζονται στον πίνακα όλα τα στοιχεία τα οποία είναι καταχωρημένα στην βάση δεδομένων σχετικά με τον πίνακα που αφορά τις αποδείξεις εσόδων. Σε οποιοδήποτε πεδίο ο χρήστης πατήσει το πλήκτρο Enter, τότε ενεργοποιείται αυτόματα το κουμπί Αναζήτηση, κατά το οποίο γίνεται η αναζήτηση των στοιχείων που πληρούν τα κριτήρια αναζήτησης. Με την επιλογή του κουμπιού: Αναίρεση, όλα τα στοιχεία που

εμφανίζονται στα πεδία της φόρμας διαγράφονται για να μπορεί ο χρήστης να εισάγει καινούργια στοιχεία- κριτήρια αναζήτησης. Όταν παρουσιαστούν τα αποτελέσματα αναζήτησης στον πίνακα, τότε ο χρήστης μπορεί να επιλέξει κάποιο από αυτά και στην συνέχεια μπορεί να επιλέξει το κουμπί Διαγραφή, κατά το οποίο γίνεται διαγραφή των στοιχείων της συγκεκριμένης απόδειξης εσόδου-εγγραφής από την βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

15.5 Αναφορά Αποδείξεων Εσόδων

Η πιο πάνω λειτουργία αφορά την φόρμα αναφορών για τις αποδείξεις εσόδων που είναι ήδη καταχωρημένες στην βάση δεδομένων. Η δημιουργία αναφορών έγινε με την χρήση του εργαλείου SAP-Crystal Reports Viewer. Για να εμφανιστούν τα στοιχεία των εσόδων πρέπει ο χρήστης να επιλέξει το κουμπί: Αναφορά. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

15.6 Παρουσίαση στοιχείων Απόδειξης Εσόδου

Η πιο πάνω λειτουργία αφορά την φόρμα παρουσίασης των στοιχείων για μία συγκεκριμένη απόδειξη εσόδου η οποία έχει επιλεγεί από την διαδικασία της αναζήτησης. Στην φόρμα αυτή ο χρήστης μπορεί να δει τα στοιχεία που είναι καταχωρημένα στην βάση δεδομένων για την συγκεκριμένη απόδειξη. Επιλέγοντας το κουμπί: Προεπισκόπηση, ανοίγει ένα αρχείο Microsoft Word 2007 στο οποίο ο χρήστης μπορεί να κάνει αλλαγές και να το τυπώσει. Επιπλέον, το Word αρχείο αποθηκεύεται αυτόματα σε συγκεκριμένο φάκελο του υπολογιστή. Ο χρήστης μπορεί να εκτυπώσει την φόρμα παρουσίασης της απόδειξης όπως αυτή εμφανίζεται από την εφαρμογή, αν επιλέξει το κουμπί Εκτύπωση. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

16. Κατηγορία Απόδειξη Εξόδου

Οι λειτουργίες της κατηγορίας αυτή είναι ακριβώς ίδιες με τις αντίστοιχες λειτουργίες της κατηγορίας: Απόδειξη Εσόδου (παράγραφος 15), με την μόνη διαφορά ότι η κατηγορία αυτή αναφέρεται στις Αποδείξεις Εξόδων και όχι Εσόδων.

17. Κατηγορία Απόρριψη

Η λειτουργία αυτή παρέχει στην χρήστη την δυνατότητα, ανάλογα με την φάρμα την οποία επιλέγει, του υποδεικνύει ποια από τα ζώα της συγκεκριμένης φάρμας θα πρέπει να παραμείνουν στην φάρμα, και ποια θα πρέπει να πωληθούν ή να σταλούν στο σφαγείο.

Απόρριψη

Απόρριψη Προβάτου



Φάρμα

Farm 1

	SHEEP ID	ANIMAL NO.	GENDER	GENOTYPE	GROUP	MILK PRODUCTIC (L)	NUM OF BIRTHS	Score	Action
▶	10	10001	F	RQ	\	2,300	1	2,2798382...	Slaughter
	11	10002	F	RR	\	1,400	1	1,6761239...	Slaughter
	12	10003	F	RQ	\	3,000	1	2,9581245...	Stay
	13	10004	F	RQ	\	1,300	1	1,5828823...	Slaughter
	14	10005	F	RR	\	1,500	1	1,6257296...	Slaughter
	15	10006	F	RQ	\	1,100	1	1,5113126...	Slaughter
	16	10007	F	RR	\	2,800	1	2,7285903...	Stay
	17	10008	F	RR	\	1,500	1	1,7570211...	Slaughter
	18	10009	F	RR	\	2,900	1	2,9854650...	Stay
	19	10010	F	RQ	\	1,300	1	1,7133667...	Slaughter
	20	10011	F	RR	\	1,200	1	1,5279401...	Slaughter
	21	10012	F	RR	\	1,100	1	1,6577008...	Slaughter
	22	10013	F	RQ	\	1,800	1	2,6146121...	Stay

Αποθήκευση

Ακύρωση

Εικόνα A-27

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα Απόρριψης, δηλαδή είναι η φόρμα στην οποία παρουσιάζονται ποια από τα ζώα της φάρμας που έχει ήδη επιλεγεί, θα πρέπει να μείνουν στην συγκεκριμένη φάρμα, και ποια θα πρέπει να πωληθούν ή να σταλούν στο σφαγείο. Αν το ζώο παραμένει στην φάρμα, τότε εμφανίζεται με πράσινο, και στο τέλος της γραμμής για το ζώο αυτό παρουσιάζεται ένα Dropdown List, σε περίπτωση που ο χρήστης αποφασίσει να μην κρατήσει το ζώο αυτό, να επιλέξει από την λίστα «Sell» ή «Slaughter», αντί «Stay». Το ίδιο συμβαίνει και με τα ζώα τα οποία θα πρέπει να φύγουν από την φάρμα, και τα οποία παρουσιάζονται με κόκκινο. Αφού ο χρήστης ολοκληρώσει την διαδικασία απόρριψης, μπορεί να επιλέξει το κουμπί: Αποθήκευση, για να καταχωρηθούν οι αλλαγές και στην βάση δεδομένων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

18. Κατηγορία Παραγωγή Γάλακτος

18.1 Εισαγωγή Παραγωγής Γάλακτος

Εικόνα A-28

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα εισαγωγής νέων στοιχείων για την παραγωγή γάλακτος του κάθε προβάτου. Όπως παρατηρείται από πιο πάνω, ο χρήστης μπορεί να επιλέξει τις διαστάσεις που θα έχει το γαλευτήρι-διάγραμμα, καθώς και την ημερομηνία εισαγωγής. Ακολούθως, ανάλογα με την θέση που έχει το πρόβατο στο γαλευτήρι, ο χρήστης μπορεί να το επιλέξει από την λίστα και να εισάγει την ποσότητα γάλακτος που είχε το πρόβατο την συγκεκριμένη ημερομηνία. Όταν ο χρήστης επιλέξει το κουμπί: Αποθήκευση, τότε ενημερώνεται ο πίνακας DAILY_MILK_PRODUCTION της βάσης δεδομένων με τα κατάλληλα στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

18.2 Κατηγοριοποίηση προβάτων

Παραγωγή Γάλακτος

Εισαγωγή Κατηγοριοποίηση Αναφορά

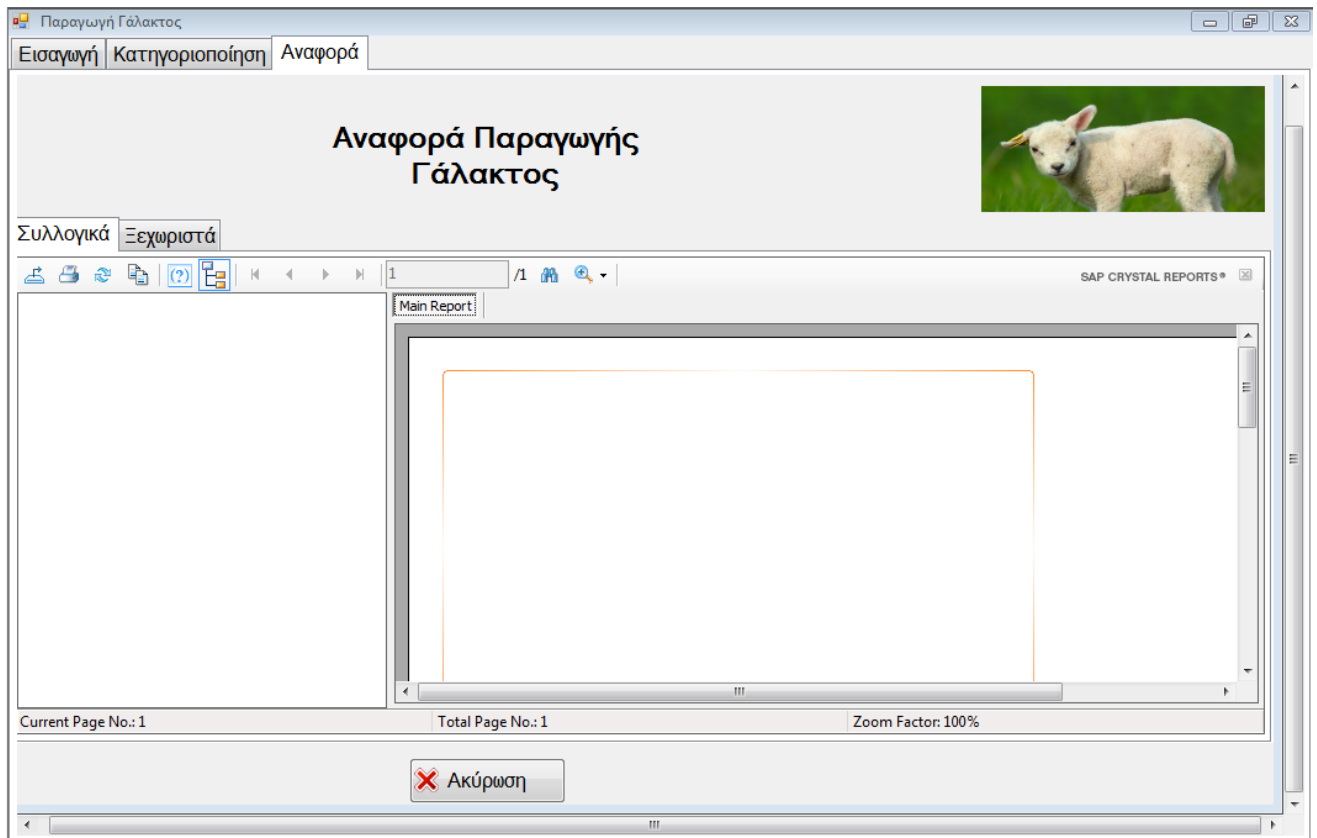
	SHEEP ID	ANIMAL NO.	GENDER	GENOTYPE	GROUP	MILK PRODUCTIO (L)	NUM OF BIRTHS
▶	10	10001	F	RQ	\	2,300	1
	11	10002	F	RR	\	1,400	1
	12	10003	F	RQ	\	3,000	1
	13	10004	F	RQ	\	1,300	1
	14	10005	F	RR	\	1,500	1
	15	10006	F	RQ	\	1,100	1
	16	10007	F	RR	\	2,800	1
	17	10008	F	RR	\	1,500	1
	18	10009	F	RR	\	2,900	1
	19	10010	F	RQ	\	1,300	1
	20	10011	F	RR	\	1,200	1
	21	10012	F	RR	\	1,100	1
	22	10013	F	RQ	\	1,800	1
	23	10014	F	RQ	\	2,600	1
	24	10015	F	RR	\	3,300	1
	25	10016	F	RQ	\	2,700	1
	26	10017	F	RR	\	1,300	1
	27	10018	F	RQ	\	1,900	1
	28	10019	F	RR	\	2,500	1
	29	10020	F	RQ	\	1,700	1
	30	10021	F	RQ	\	1,500	1

Αποθήκευση Ακύρωση

Εικόνα A-29

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα κατηγοριοποίησης των προβάτων, δηλαδή ο χρήστης μπορεί να επιλέξει σε ποιά θέση βρίσκεται το κάθε πρόβατο στο γαλευτήριο, με το να εισάγει την αντίστοιχη θέση στην στήλη “GROUP”, του πίνακα που εμφανίζεται. Καμία από τις άλλες στήλες του πίνακα δεν μπορεί να μετατραπεί. Όταν ο χρήστης επιλέξει το κουμπί: Αποθήκευση, τότε ενημερώνεται ο πίνακας FEMALE_SHEEP της βάσης δεδομένων με τα κατάλληλα στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

18.3 Αναφορά Παραγωγής Γάλακτος



Εικόνα A-30

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα Αναφοράς της Παραγωγής Γάλακτος. Ο χρήστης έχει την δυνατότητα να δει την πρόοδο των ζώων ως προς την παραγωγή γάλακτος, είτε συλλογικά, είτε ξεχωριστά, και έτσι να διευκολυνθεί στην λήψη αποφάσεων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

19. Κατηγορία Κατανάλωση Τροφής

19.1 Εισαγωγή Κατανάλωσης Τροφής

Εικόνα A-31

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα εισαγωγής νέων στοιχείων για την κατανάλωση τροφής για μία ομάδα προβάτων, ανάλογα με την θέση που έχουν στην φάρμα. Όπως παρατηρείται από πιο πάνω, ο χρήστης μπορεί να επιλέξει τις διαστάσεις- κυψελίδες που θα έχει η φάρμα-διάγραμμα, καθώς και την ημερομηνία εισαγωγής. Ακολουθώντας, ανάλογα με την θέση που έχει το πρόβατο στην φάρμα, ο χρήστης μπορεί να τα δει από την λίστα και να εισάγει την ποσότητα τροφής που κατανάλωσαν τα πρόβατα της αντίστοιχης κυψελίδας την συγκεκριμένη ημερομηνία. Όταν ο χρήστης επιλέξει το κουμπί: Αποθήκευση, τότε ενημερώνεται ο πίνακας DAILY_FOOD_CONSUMPTION της βάσης δεδομένων με τα κατάλληλα στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

19.2 Κατηγοριοποίηση Προβάτων

Κατανάλωση Τροφής

Εισαγωγή
Κατηγοριοποίηση
Αναφορά

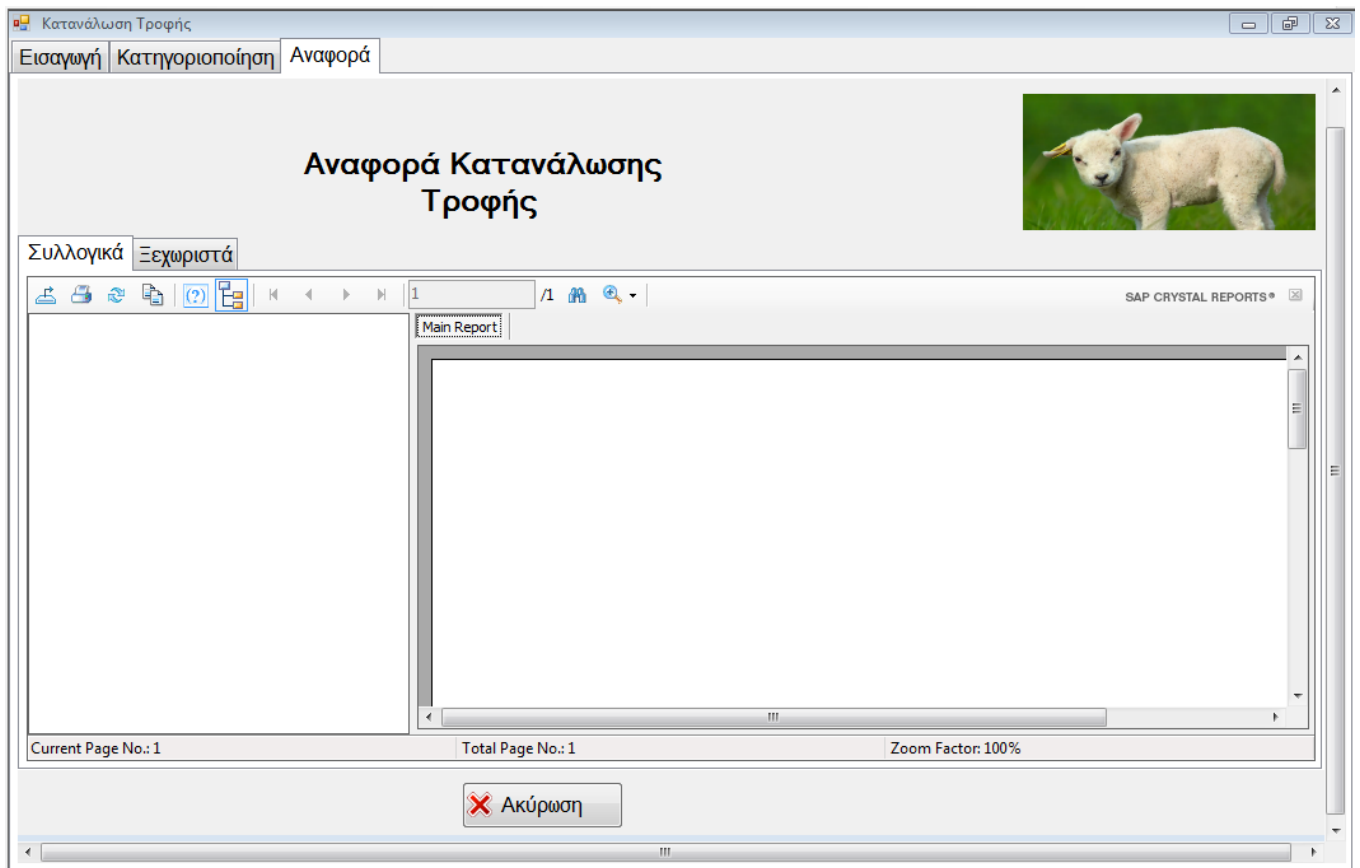
	SHEEP ID	ANIMAL NO.	GENDER	GENOTYPE	GROUP	FOOD CONSUMPT (Kg)	FOOD COST (€)
▶	1	9020001	M	RR	\	1,500	1,000
	2	9020002	M	RQ	\	1,500	1,000
	3	9020003	M	RR	\	1,500	1,000
	4	9030001	M	RR	\	1,500	1,000
	5	9030002	M	RQ	\	1,500	1,000
	6	9030003	M	RQ	\	1,500	1,000
	7	9040001	M	RR	\	1,500	1,000
	8	9040002	M	RQ	\	1,500	1,000
	9	9040003	M	RR	\	1,500	1,000
	10	10001	F	RQ	\	1,500	1,000
	11	10002	F	RR	\	1,500	1,000
	12	10003	F	RQ	\	1,500	1,000
	13	10004	F	RQ	\	1,500	1,000
	14	10005	F	RR	\	1,500	1,000
	15	10006	F	RQ	\	1,500	1,000
	16	10007	F	RR	\	1,500	1,000
	17	10008	F	RR	\	1,500	1,000
	18	10009	F	RR	\	1,500	1,000
	19	10010	F	RQ	\	1,500	1,000
	20	10011	F	RR	\	1,500	1,000
	21	10012	F	RR	\	1,500	1,000

Αποθήκευση
Ακύρωση

Εικόνα A-32

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα κατηγοριοποίησης των προβάτων, δηλαδή ο χρήστης μπορεί να επιλέξει σε ποιά θέση-κυψέλη βρίσκεται το κάθε πρόβατο στην φάρμα, με το να εισάγει την αντίστοιχη θέση στην στήλη “GROUP”, του πίνακα που εμφανίζεται. Καμία από τις άλλες στήλες του πίνακα δεν μπορεί να μετατραπεί. Όταν ο χρήστης επιλέξει το κουμπί: Αποθήκευση, τότε ενημερώνεται ο πίνακας SHEEP της βάσης δεδομένων με τα κατάλληλα στοιχεία. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.

19.3 Αναφορά Κατανάλωσης Τροφής



Εικόνα Α-33

Στην πιο πάνω εικόνα παρουσιάζεται η φόρμα Αναφοράς της Κατανάλωσης Τροφής. Ο χρήστης έχει την δυνατότητα να δει την πρόοδο των ζώων ως προς την κατανάλωση τροφής, είτε συλλογικά, είτε ξεχωριστά, και έτσι να διευκολυνθεί στην λήψη αποφάσεων. Με την επιλογή του κουμπιού: Ακύρωση, δεν γίνεται καμία ενέργεια στην βάση δεδομένων, και η φόρμα τερματίζεται.