

Ατομική Διπλωματική Εργασία

ΕΞΥΠΝΟ ΠΟΡΤΟΦΟΛΙ

Γιώργος Αντωνιάδης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ιούλιος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Έξυπνο Πορτοφόλι

Γιώργος Αντωνιάδης

Επιβλέπων Καθηγητής
Αντώνης Κάκας

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

Ιούλιος 2019

Ευχαριστίες

Με την παρούσα διπλωματική εργασία ολοκληρώνεται ένα μακρύ και αναπόσπαστο κομμάτι της ζωής μου, αυτό των προπτυχιακών μου σπουδών στο Πανεπιστήμιο Κύπρου. Ρίχνοντας μια ματιά πίσω στα τελευταία χρόνια βλέπω πόσο πικρό αλλά συνάμα και πόσο επικοδομητικό ήταν αυτό το ταξίδι για εμένα καθώς μου έδωσε τόσες πολλές γνώσεις αλλά πιο σημαντικό είναι πως μου έδωσε τα εργαλεία που χρειαζόμουν για να συνεχίσω να μαθαίνω και να γίνομαι καλύτερος καθ' όλη τη διάρκεια της ζωής μου. Φυσικά ένα τέτοιο ταξίδι δεν θα μπορούσα ποτέ να το ολοκληρώσω μόνος μου και για αυτό θα ήθελα να προσφέρω μερικά λόγια για να ευχαριστήσω τα άτομα που με στήριξαν σε αυτό το ταξίδι και συνεχίζουν να με στηρίζουν στο ακόμα μεγαλύτερο ταξίδι, αυτό που λέγεται «ζωή».

Πρώτα από όλα, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, Κύριο Αντώνη Κάκα ο οποίος στήριξε την ιδέα πίσω από τη δημιουργία αυτής της Διπλωματικής Εργασίας και μου έδωσε την ευκαιρία να την υλοποιήσω. Επίσης θα ήθελα να τον ευχαριστήσω για την υπομονή που έχει δείξει κατά τη διάρκεια της υλοποίησης αυτής της εργασίας καθώς επίσης και για την καθοδήγηση του προς την επίτευξη της.

Επίσης θα ήθελα να ευχαριστήσω τον φίλο μου Ζαχαρία Γεωργίου, ο οποίος με τις τεχνικές του γνώσεις προσέφερε τα μέγιστα ούτως ώστε να εξικoiωθώ με κάποια από τα εργαλεία τα οποία ήταν απαραίτητα για την ανάπτυξη και υλοποίηση του συστήματος.

Έπειτα, θα ήθελα να ευχαριστήσω τον ξάδελφο μου Νικόλα Νίκου, ο οποίος με τη στήριξη του κατά τη διάρκεια της συγκατοίκησης μας με βοήθησε να συγκεντρωθώ στον στόχο μου, δηλαδή την απόκτηση του πτυχίου και να μην αποκλίνω από αυτόν όσο δύσκολες και δυσίονες έμοιαζαν οι καταστάσεις κατά καιρούς.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου, Κυριάκο και Κυριακούλλα, καθώς επίσης και την αδελφή μου Μαρία, για όλη την υποστήριξη που μου έχουν προσφέρει ανά τα χρόνια και που πάντα βρίσκονταν δίπλα μου και πίστευαν σε εμένα.

Ευχαριστώ,

Γιώργος Αντωνιάδης

Περίληψη

Ο διπλωματική εργασία έχει τίτλο Έξυπνο Πορτοφόλι, και όπως υπονοεί ο τίτλος σχετίζεται με τα χρήματα του κάθε ατόμου και την διαχείριση αυτών των χρημάτων. Πιο συγκεκριμένα η εφαρμογή ζητάει από τον χρήστη να καθορίσει ένα προϋπολογισμό για τον ίδιο και να δώσει μερικές κατευθυντήριες γραμμές ούτως ώστε να γνωρίζει σε γενικές γραμμές τις προτιμήσεις του χρήστη όσον αφορά το ξόδεμα των χρημάτων του. Κατά τη διάρκεια της αλληλεπίδρασης του χρήστη με την εφαρμογή, ο τελευταίος λαμβάνει εισηγήσεις για τον τρόπο με τον οποίο μπορεί να ξοδέψει τα χρήματα του και αναπόκειται στον ίδιο να αποφασίσει με ποιο τρόπο θα το πράξει.

Για την υλοποίηση του συγκεκριμένου συστήματος, αρχικά χρειάστηκε να κατανοήσουμε τον τρόπο με τον οποίο η πλειοψηφία του κόσμου επιλέγει να ξοδέψει τα χρήματα του. Ως εκ τούτου ήταν αναγκαίο να ενημερωθούμε για τις σύγχρονες τάσεις και ανάγκες ξοδέματος χρημάτων, καθώς επίσης και για τους μοχλούς πίεσης πίσω από αυτές τις τάσεις και ανάγκες. Αυτό επιτεύχθηκε διαβάζοντας διάφορα σχετικά άρθρα και ποιο συγκεκριμένα το βιβλίο Nudge των Richard H. Thaler και Cass R. Sunstein. Μέσα από την έρευνα μας οδηγηθήκαμε στη δημιουργία πέντε κατηγοριών, οι οποίες είναι η Ένδυση, Ψυχαγωγία, Σίτιση, Υποχρεώσεις και τέλος τα Καθημερινά έξοδα.

Εν συνεχεία, ψάξαμε και εντοπίσαμε τις τεχνολογίες οι οποίες ήταν απαραίτητες για την υλοποίηση του συστήματος μας και εκπαιδευτήκαμε στη χρήση τους. Τέλος το σύστημα δοκιμάστηκε από πραγματικούς χρήστες ούτως ώστε να διασαφηνίσουμε εάν η εφαρμογή είναι λειτουργική και εύκολη στη χρήση για τον απλό άνθρωπο. Παράλληλα λάβαμε ενημέρωση από τους χρήστες για δικές τους ιδέες και τυχόν αλλαγές οι οποίες οι ίδιοι πιστεύουν ότι θα έκανε την εφαρμογή ποιο λειτουργική και θα προσέφερε καλύτερες υπηρεσίες.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Γενική Εισαγωγή	1
	1.2 Στόχος Διπλωματικής Εργασίας	2
	1.3 Δομή Διπλωματικής Εργασίας	2
Κεφάλαιο 2	Περιγραφή Προβλήματος.....	4
	2.1 Εισαγωγή	4
	2.2 Περιγραφή Προβλήματος	4
	2.3 Υπάρχον Συστήματα	5
	2.4 Μειονεκτήματα Υπάρχων Συστημάτων	6
Κεφάλαιο 3	Ανάλυση Απαιτήσεων-Προδιαγραφών.....	25
	3.1 Εισαγωγή	25
	3.2 Ανάλυση Απαιτήσεων-Προδιαγραφών	30
Κεφάλαιο 4	Γοργίας.....	25
	4.1 Εισαγωγή	25
	4.2 Το σύστημα Γοργίας	30
Κεφάλαιο 5	Τεχνολογίες.....	44
	5.1 Εισαγωγή	44
	5.2 PHP	47
	5.3 JavaScript	52
	5.4 HTML/CSS	54
	5.5 SQL	55
	5.6 Docker	56

Κεφάλαιο 6	Αρχιτεκτονική Συστήματος.....	57
	6.1 Εισαγωγή	57
	6.2 Μεθοδολογία Ανάπτυξης Συστήματος	58
	6.3 Μοντελοποίηση Συστήματος	59
	6.4 Υλοποίηση Συστήματος	60
 ΚΕΦΑΛΑΙΟ 7	 Αξιολόγηση Συστήματος.....	 70
	7.1 Εισαγωγή	70
	7.2 Έλεγχος Συστήματος	73
	7.3 Αξιολόγηση-Αποτελέσματα	70
 ΚΕΦΑΛΑΙΟ 8	 Συμπεράσματα-Μελλοντική Εργασία.....	 70
	8.1 Εισαγωγή	70
	8.2 Μελλοντική Εργασία	73
 Βιβλιογραφία		 80
 Παράρτημα Α		 A-1
 Παράρτημα Β		 B-3

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Εισαγωγή	1
1.2 Στόχος Διπλωματικής Εργασίας	2
1.3 Δομή Διπλωματικής Εργασίας	2

1.1 Γενική Εισαγωγή

Στην εποχή στην οποία ζούμε, ο κάθε άνθρωπος καλείται να λάβει αποφάσεις επί καθημερινής βάσεως οι οποίες σχετίζονται με τη διάθεση των χρημάτων του για την εξυπηρέτηση των αναγκών του, είτε αυτές είναι βραχυπρόθεσμες όπως μια απλή αγορά ενός ποτηριού καφέ, είτε αυτές είναι μακροπρόθεσμες όπως η αποπληρωμή δανείων και λογαριασμών. Λόγω και της πρόσφατης οικονομικής κρίσης η οποία τα τελευταία χρόνια έχει επηρεάσει ολόκληρη την υφήλιο με σφοδρές συνέπειες, τόσο σε ατομικό αλλά και σε εθνικό επίπεδο, η ανάγκη για καλύτερη διαχείριση των χρημάτων μας είναι πλέον επιτακτική. Μάλιστα, αρκετές είναι οι εταιρείες και τράπεζες οι οποίες προσφέρουν πλέον ως κομμάτι των υπηρεσιών τους, αναλυτικά δεδομένα στους πελάτες τους ούτως ώστε οι τελευταίοι να μπορούν να βλέπουν καλύτερα πως ξοδεύουν τα χρήματά τους, καθώς επίσης και μεθόδους καλύτερης αποταμίευσης αυτών των χρημάτων.

Μέσα από την παρούσα διπλωματική εργασία αποσκοπούμε στην δημιουργία μιας διαδικτυακής εφαρμογής μέσα από την οποία οι χρήστες να μπορούν να επιλέξουν ένα προϋπολογισμό για τον οποίο οι ίδιοι νιώθουν άνετα και να κάνουν την καλύτερη για τους ίδιους διαχείριση αυτού του προϋπολογισμού. Για να επιτευχθεί αυτό, η εφαρμογή προτείνει στον κάθε χρήστη τρόπους με τους οποίους μπορεί να ξοδέψει τα χρήματά του λαμβάνοντας υπόψη τις δικές του προτιμήσεις και ανάγκες χωρίς όμως να λαμβάνει τις αποφάσεις για τον ίδιο. Με αυτό τον τρόπο επιτυγχάνουμε την έννοια ενός προσωπικού βοηθού για την

καλύτερη διαχείριση των χρημάτων του κάθε χρήστη και όχι μιας μηχανής η οποία είναι απόλυτη και λαμβάνει αποφάσεις για τον χρήστη.

1.2 Στόχος Διπλωματικής Εργασίας

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η υλοποίηση μιας διαδικτυακής εφαρμογής η οποία θα βοηθά τους χρήστες να διαχειρίζονται καλύτερα τα χρήματά τους. Δηλαδή, μέσα από την εφαρμογή αυτή οι χρήστες να βοηθηθούν στην καλύτερη διάθεση των χρημάτων τους ως προς την κάλυψη των αναγκών διαβίωσης τους και προσωπικών τους αναγκών. Με αυτό το τρόπο η εφαρμογή αυτή βοηθά τον χρήστη να μην κάνει άσκοπα έξοδα, τα οποία ουσιαστικά δεν χρειάζεται και δεν θέλει, αλλά αυτά που χρειάζεται και ικανοποιούν τις πραγματικές του ανάγκες.

Για να επιτευχθεί αυτό, ο κάθε χρήστης συνδέεται στην εφαρμογή δίνοντας αρχικά ένα προϋπολογισμό τον οποίο ο ίδιος θέλει να ακολουθήσει και μερικές κατευθυντήριες γραμμές ως προς τις προτιμήσεις του. Έπειτα, όταν ο χρήστης θέλει να ξοδέψει χρήματα για οποιοδήποτε λόγο μπορεί να αλληλοεπιδράσει με το σύστημα το οποίο λαμβάνοντας υπόψη τις προτιμήσεις του χρήστη θα επιστρέψει πιθανούς τρόπους με τους οποίους μπορεί να διαθέσει τα χρήματά του.

1.3 Δομή Διπλωματικής Εργασίας

Στο πρώτο κεφάλαιο κάνουμε μια γενική εισαγωγή στο θέμα της διπλωματικής εργασίας, στο σκοπό της καθώς επίσης και στη δομή του παρόντος εγγράφου.

Στο δεύτερο κεφάλαιο προχωρούμε στην περιγραφή του προβλήματος και πώς αυτό επηρεάζει τη σύγχρονη κοινωνία. Επίσης κάνουμε αναφορά σε υπάρχον συστήματα τα οποία παρουσιάζουν παρόμοια συμπεριφορά με την δική μας εφαρμογή ή που απευθύνονται στο ίδιο πρόβλημα με άλλους τρόπους. Τέλος, γίνεται αναφορά στα μειονεκτήματα αυτών των συστημάτων.

Στο τρίτο κεφάλαιο, μέσα από την έρευνα μας εντοπίζουμε τις απαιτήσεις του συστήματος και ορίζουμε τον τρόπο με τον οποίο αυτές πρέπει να υλοποιηθούν.

Στο τέταρτο κεφάλαιο θα αναλύσουμε τις τεχνολογίες τις οποίες χρησιμοποιήσαμε για την υλοποίηση του συστήματος, καθώς επίσης και τους λόγους που επιλέχθηκαν.

Στο πέμπτο κεφάλαιο αναφερόμαστε στην αρχιτεκτονική του συστήματος και πιο συγκεκριμένα την μεθοδολογία με την οποία αναπτύχθηκε, την μοντελοποίηση και τελική υλοποίηση του.

Στο έκτο κεφάλαιο κάνουμε την αξιολόγηση του συστήματος μας. Δηλαδή δοκιμάζουμε διάφορα σενάρια και λαμβάνουμε πληροφορίες από χρήστες που το έχουν δοκιμάσει.

Στο έβδομο κεφάλαιο αναφερόμαστε στα συμπεράσματα τα οποία έχουμε εξαγάγει και στην μελλοντική εργασία ούτως ώστε να αναβαθμιστεί το σύστημα που έχουμε δημιουργήσει.

Κεφάλαιο 2

Περιγραφή Προβλήματος

2.1 Εισαγωγή	4
2.2 Περιγραφή Προβλήματος	4
2.3 Υπάρχον Συστήματα	5
2.4 Μειονεκτήματα Υπάρχων Συστημάτων	6

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα προχωρήσουμε σε μια αναλυτική περιγραφή του προβλήματος. Δηλαδή, θα εξηγήσουμε ποιο είναι το πρόβλημα το οποίο καλούμαστε να λύσουμε μέσα από την εφαρμογή που αναπτύξαμε και τους λόγους για τους οποίους αυτό το πρόβλημα πρέπει να λάβει την απαραίτητη προσοχή ούτως ώστε να αντιμετωπισθεί. Επίσης θα ανατρέξουμε σε παρόμοια συστήματα για να παρατηρήσουμε τις διαφορές τους με τη δική μας εφαρμογή, καθώς επίσης και τα μειονεκτήματά τους. Τέλος θα αναλύσουμε πως μπορούμε να ξεπεράσουμε αυτά τα μειονεκτήματα μέσα από το δικό μας σύστημα, εάν αυτό είναι δυνατόν.

2.2 Περιγραφή Προβλήματος

Το πρόβλημα με το οποίο ασχολείται η παρούσα διπλωματική εργασία είναι ο τρόπος με τον οποίο οι άνθρωποι διαθέτουν τα χρήματά τους για την καλύτερη εξυπηρέτηση των αναγκών τους. Αυτό είναι ένα πρόβλημα το οποίο ταλανίζει την ανθρωπότητα για αρκετούς αιώνες καθώς όλοι οι άνθρωποι έχουν την συνήθεια να ξοδεύουν χρήματα σε υλικά αγαθά και απολαύσεις τα οποία είτε δεν χρειάζονται, είτε δεν θέλουν, όμως ακολουθούν την μάζα η οποία τους ωθεί σε τέτοιες αγορές, ενώ παράλληλα αφήνουν στο πλάι άλλες ανάγκες, ποιο επιτακτικές όπως ενοίκια και λογαριασμούς. Αυτές οι επιτακτικές ανάγκες με τη σειρά τους επηρεάζουν την διαβίωση των ατόμων με καταστροφικές συνέπειες, τόσο για τα άτομα όσο και για την κοινωνία γενικότερα, όπως έχουμε διαπιστώσει άλλωστε ιδίως όμματι μετά και την πρόσφατη οικονομική κρίση η οποία έφερε όλες μας τις οικονομικές αδυναμίες στο φως

και οδήγησε πάρα πολλά άτομα, οργανισμούς, ακόμα και κράτη στα πρόθυρα της πτώχευσης και της οικονομικής εξαθλίωσης.

Μέσα από την διαδικτυακή μας εφαρμογή ο κάθε χρήστης έχει την δυνατότητα να επιλέξει ένα προϋπολογισμό για τον ίδιο και να μας καθοδηγήσει ως προς τις βασικές προτιμήσεις του για τη διάθεση αυτού του προϋπολογισμού σε σχέση με πέντε κύριες κατηγορίες. Αυτές οι κατηγορίες είναι η Ένδυση, Ψυχαγωγία, Σίτιση, Υποχρεώσεις και τέλος τα Καθημερινά έξοδα. Φυσικά κάθε μία από αυτές τις κατηγορίες περιλαμβάνει ένα αριθμό υποκατηγοριών, όπως για παράδειγμα η Ψυχαγωγία περιλαμβάνει Αθλήματα, Διακοπές, Θέατρο κτλ., ενώ στις Υποχρεώσεις συμπεριλαμβάνονται Λογαριασμοί, Φόροι, Ενοίκια κτλ. Αφού γίνει η δημιουργία του λογαριασμού του χρήστη, τότε ο τελευταίος έχει τη δυνατότητα να αλληλοεπιδρά με το σύστημα με μια μορφή συνομιλίας. Ο χρήστης μπορεί να ρωτά ουσιαστικά για πιθανές αγορές ή έξοδα τα οποία θέλει να κάνει και το σύστημα να του επιστρέφει ως απάντηση μια λίστα με πιθανούς τρόπους διάθεσης των χρημάτων του. Οι πιθανές λύσεις που προσφέρονται απορρέουν μέσα από τις προτιμήσεις του χρήστη ενώ πάντα εναπόκειται στον ίδιο ποια επιλογή θα ακολουθήσει. Όποια και αν είναι η επιλογή του χρήστη, ο ίδιος καλείται να ενημερώσει το σύστημα για αυτήν έτσι ώστε η εφαρμογή να μπορεί να καταγράφει την κατάσταση του προϋπολογισμού του και τις προτιμήσεις του.

2.3 Υπάρχων Συστήματα

Είδη, αρκετοί είναι οι οργανισμοί οι οποίοι έχουν αντιληφθεί την ανάγκη να παρέχουν στους πελάτες τους υπηρεσίες, μέσα από τις οποίες οι τελευταίοι να έχουν την δυνατότητα να διαχειρίζονται καλύτερα τα χρήματά τους. Πέντε τέτοιοι οργανισμοί είναι οι N26, Revolut, Starling, Monese και Cleo.

Οι πρώτοι τέσσερις οργανισμοί αποτελούν, ουσιαστικά, χρηματοπιστωτικές εταιρίες. Στα συστήματα τους δίνεται η δυνατότητα στους χρήστες να δημιουργήσουν τραπεζικούς λογαριασμούς και να εκδώσουν πιστωτικές κάρτες για αυτούς τους λογαριασμούς. Γενικά αυτά τα συστήματα χρησιμοποιούνται κατά κύριο λόγο, για διεκπεραίωση τραπεζικών συναλλαγών όπως πληρωμές και μεταφορές χρημάτων με ελαφρύτερες χρεώσεις από τις παραδοσιακές εθνικές τράπεζες. Ωστόσο, μέσα από τα συστήματά τους, παρέχεται η δυνατότητα στο χρήστη να εξοικονομήσει χρήματα, είτε διαμέσου δημιουργίας βοηθητικών λογαριασμών στους οποίους μπορεί να αποθηκεύσει μέρος των χρημάτων του, είτε με καθορισμό συγκεκριμένου προϋπολογισμού τον οποίο δεν θα έχει τη δυνατότητα να ξεπεράσει. Επίσης δίνεται στον χρήστη η επιλογή να θέσει στόχους, βραχυπρόθεσμους και μακροπρόθεσμους, όπου μόνο για αυτούς μπορεί να διαθέσει μέρος των χρημάτων του.

Τέλος, προσφέρεται η δυνατότητα ενημέρωσης του πελάτη με λίστες αναφοράς όσον αφορά τους τρόπους με τους οποίους ξόδεψε τα χρήματα του ανά μήνα.

Το πέμπτο σύστημα το οποίο αναλύουμε(Cleo) λειτουργεί διαφορετικά από τα προηγούμενα τέσσερα. Καταρχήν δεν πρόκειται για χρηματοπιστωτικό ίδρυμα, αλλά βασική προϋπόθεση για να δουλέψει είναι η σύνδεση του μαζί με τους τραπεζικούς λογαριασμούς του χρήστη. Αφού αυτό επιτευχθεί αλληλοεπιδρά με το χρήστη διαμέσου messenger ή text ενώ οι κύριες υπηρεσίες που προσφέρει σχετίζονται με τον έλεγχο των εξόδων του χρήστη ανά κατηγορία, τον έλεγχο πρόσφατων συναλλαγών, την άμεση πρόσβαση στα περιεχόμενα των λογαριασμών του χρήστη και τέλος τον έλεγχο του προϋπολογισμού του χρήστη. Αυτός μπορεί να είναι ημερήσιος, εβδομαδιαίος ή και μηνιαίος.

2.4 Μειονεκτήματα Υπάρχων Συστημάτων

Τα πρώτα τέσσερα συστήματα είναι προϊόντα εταιριών οι οποίες παρέχουν τραπεζικές υπηρεσίες με κύριο σκοπό τις χρηματικές συναλλαγές, μετατροπές χρημάτων σε διάφορα άλλα συναλλάγματα ή και cryptocurrency και αποταμίευση των χρημάτων αυτών. Επίσης βασική προϋπόθεση για τη χρήση αυτών των συστημάτων είναι η δημιουργία τραπεζικών λογαριασμών στις εταιρείες που τα παρέχουν.

Από την άλλη το πέμπτο σύστημα(Cleo) προϋποθέτει από τους χρήστες του να συνδέσουν τους τραπεζικούς λογαριασμούς στους οποίους αυτό θα έχει πρόσβαση. Ως εκ τούτου οι χρήστες είναι υποχρεωμένοι να παρέχουν ευαίσθητα προσωπικά στοιχεία, κάτι το οποίο πολλοί από αυτούς προτιμούν να αποφύγουν είτε λόγω έλλειψης εμπιστοσύνης είτε λόγω ανασφάλειας σχετικά με τις διαδικτυακές συναλλαγές.

Τέλος, κανένα από τα πιο πάνω συστήματα δεν λαμβάνει υπόψη τις προσωπικές προτιμήσεις και ανάγκες των ατόμων, ούτε κάνουν εισηγήσεις ως προς την διάθεση των χρημάτων τους. Τα υπάρχον συστήματα απλώς παρέχουν στατιστικά και πληροφορίες σχετικά με συναλλαγές που έχουν ήδη πραγματοποιηθεί καθώς επίσης και μεθόδους αποταμίευσης των χρημάτων των χρηστών.

Κεφάλαιο 3

Ανάλυση Απαιτήσεων-Προδιαγραφών

3.1 Εισαγωγή	7
3.2 Ανάλυση Απαιτήσεων-Προδιαγραφών	7

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται η ανάλυση των απαιτήσεων και των προδιαγραφών τα οποία πρέπει να πληροί το σύστημα ούτως ώστε να δίνει λύση στο πρόβλημα διαχείρισης των χρημάτων του κάθε ατόμου, χωρίς όμως να λαμβάνει το ίδιο το σύστημα την τελική απόφαση αλλά αυτή να ανήκει αποκλειστικά στο άτομο-χρήστη της εφαρμογής.

3.2 Ανάλυση Απαιτήσεων-Προδιαγραφών

3.2.1 Αξιοπιστία-Ασφάλεια

Η κυριότερη ανησυχία του κόσμου όσον αφορά τα προσωπικά και οικονομικά του στοιχεία είναι πρώτα από όλα αυτά να είναι όσο το δυνατόν καλύτερα προστατευμένα. Παρόλο που στο παρών στάδιο η εφαρμογή μας δεν είναι απευθείας συνδεδεμένη με τραπεζικούς λογαριασμούς ή πιστωτικές κάρτες των χρηστών το ποσό το οποίο καλούνται να δώσουν ως εισόδο στην εφαρμογή μας ενδέχεται να είναι ενδεικτικό της οικονομικής τους κατάστασης και για αυτό πρέπει να προστατεύεται. Επίσης οι καταστάσεις με τις αποφάσεις διάθεσης των χρημάτων των χρηστών είναι ενδεικτικές των προτιμήσεων τους και πρέπει να προστατεύονται από τρίτα πρόσωπα τα οποία ενδέχεται να θέλουν να τις εκμεταλλευτούν για προσωπικό τους κέρδος. Για αυτό το σκοπό, κάθε χρήστης έχει δικό του λογαριασμό με τον προσωπικό του κωδικό, τον οποίο θα γνωρίζει μόνο ο ίδιος. Τέλος, μόνο ο διαχειριστής του συστήματος έχει πρόσβαση στα προσωπικά δεδομένα των χρηστών και με βάση τον νόμο είναι υπόχρεος να τα διαφυλάσσει και με κανένα τρόπο να τα μεταβιβάσει σε οποιοδήποτε τρίτο πρόσωπο ή οργανισμό.

3.2.2 Διαθεσιμότητα

Από τη στιγμή που η εφαρμογή θα αφηθεί για χρήση στο ευρύ κοινό πρέπει να είναι πάντα διαθέσιμη στους χρήστες για να μπορούν να βλέπουν τον υπάρχων προϋπολογισμό τους ή να λάβουν κάποια εισήγηση από το σύστημα. Η εφαρμογή μπορεί να μην είναι διαθέσιμη μόνο σε περίπτωση συντήρησης ή αναβάθμισης, όπου η πλείστη προεργασία πρέπει να λάβει χώρα νωρίτερα ούτως ώστε να μειωθεί ο χρόνος που η εφαρμογή δεν θα είναι σε λειτουργία.

3.2.3 Συντήρηση-Αναβάθμιση

Η συντήρηση και αναβάθμιση της εφαρμογής τόσο κατά τη διάρκεια ανάπτυξης της, όσο και μετά την διάθεση της στο ευρύ κοινό αποτελεί κομμάτι τεράστιας σημασίας. Αυτό οφείλεται στο γεγονός ότι οι τρόποι με τους οποίους οι άνθρωποι επιλέγουν να ξοδεύουν τα χρήματα είναι όλο ένα και ποιο μεταβλητός καθώς οι σύγχρονες τάσεις αλλά και οι οικονομικές ανάγκες της σύγχρονης εποχής μεταβάλλονται συνεχώς. Επίσης καθώς οι χρήστες της διαδικτυακής εφαρμογής αυξάνονται, πολύ πιθανό είναι να φέρνουν μαζί τους νέες ανάγκες τις οποίες η εφαρμογή θα πρέπει να λαμβάνει υπόψη και να διαχειρίζεται.

3.2.4 Χρήστης

Το σύστημα περιλαμβάνει την οντότητα του Χρήστη, η οποία περιλαμβάνει τις εξής ιδιότητες και δυνατότητες.

Λειτουργία 1: Δημιουργία Λογαριασμού

Ο χρήστης πρέπει να δημιουργήσει ένα προσωπικό λογαριασμό παρέχοντας ένα username και password για να έχει τη δυνατότητα να χρησιμοποιήσει την εφαρμογή.

Λειτουργία 2: Καθορισμός αρχικού προϋπολογισμού και δυνατότητα αλλαγής αυτού του προϋπολογισμού.

Ο χρήστης πρέπει να ορίσει ένα προϋπολογισμό τον οποίο το σύστημα θα λαμβάνει υπόψη και θα βασίζεται όταν κάνει πιθανές προτάσεις στον χρήστη. Ο χρήστης έχει τη δυνατότητα να αλλάξει αυτό το ποσό.

Λειτουργία 3: Παροχή Αρχικών Προτιμήσεων

Όταν ο χρήστης δημιουργήσει τον λογαριασμό του, θα κληθεί να απαντήσει σε μερικές απλές ερωτήσεις οι οποίες χρησιμοποιούνται ως κατευθυντήριες γραμμές για μια αρχική κατανόηση των προτιμήσεων του από το σύστημα.

Λειτουργία 4: Ερωτήσεις ως προς την εφαρμογή και επιλογή απάντησης

Ο χρήστης μπορεί να αλληλοεπιδράσει με το σύστημα, ουσιαστικά ρωτώντας το πως ο ίδιος μπορεί να ξοδέψει τα χρήματά του, ή αν μπορεί να κάνει μια συγκεκριμένη αγορά, δίνοντας ως περαιτέρω είσοδο μερικές παραμέτρους όπως παραδείγματος χάρη την τρέχων χρονική περίοδο ή και προτίμηση ως προς κάποιο έξοδο.

Λειτουργία 5: Έλεγχος τρέχων ισοζυγίου προϋπολογισμού

Ο χρήστης έχει τη δυνατότητα ποια είναι η κατάσταση του ισοζυγίου των χρημάτων του.

Λειτουργία 6: Έλεγχος αποφάσεων

Ο χρήστης μπορεί να ανατρέξει σε αποφάσεις τις οποίες ο ίδιος έχει πάρει στο παρελθόν για να διαπιστώσει καλύτερα πως ξοδεύει τα χρήματά του.

Λειτουργία 7: Ενημέρωση Εφαρμογής

Ο χρήστης ενημερώνει το σύστημα για αποφάσεις που έχει πάρει έτσι ώστε να κρατείται ενήμερο το ισοζύγιο του προϋπολογισμού του καθώς επίσης και οι προτιμήσεις του. Αυτό είναι πολύ σημαντικό για να δίνεται η δυνατότητα στην εφαρμογή να του προσφέρει τις καλύτερες δυνατές απαντήσεις.

Λειτουργία 8: Αναφορά Σφαλμάτων

Ο χρήστης πρέπει να αναφέρει στον διαχειριστή / διαχειριστές του συστήματος τυχόν προβλήματα ή σφάλματα τα οποία θα παρουσιαστούν ή θα έρθουν εις γνώση του.

Λειτουργία 9: Προτάσεις Βελτίωσης

Ο χρήστης μπορεί να ενημερώσει την ομάδα διαχείρισης τους συστήματος για τυχόν ιδέες οι οποίες θεωρεί ο ίδιος πως θα βελτίωναν την εφαρμογή και την χρήση της.

3.2.5 Διαχειριστής

Ο Διαχειριστής του συστήματος είναι υπεύθυνος για την ομαλή λειτουργία του.

Λειτουργία 1: Έλεγχος Σφαλμάτων

Σε περίπτωση κάποιου σφάλματος, έχει την ευθύνη να το αναφέρει στην ομάδα ανάπτυξης της εφαρμογής και να ενημερώσει τους χρήστες ότι αυτό βρίσκεται υπό διερεύνηση και θα λυθεί το συντομότερο δυνατόν.

Λειτουργία 2: Έλεγχος Προτάσεων

Ο Διαχειριστής / Διαχειριστές του συστήματος πρέπει να προωθούν προτάσεις των χρηστών στην ομάδα ανάπτυξης της εφαρμογής, οι οποίες μπορούν να βελτιώσουν την εφαρμογή και την χρήση της.

Λειτουργία 3: Δημιουργία Αναφορών και Ανάλυσης Προτιμήσεων

Ο Διαχειριστής / Διαχειριστές του συστήματος είναι υπεύθυνοι για τη συλλογή στατιστικών στοιχείων σχετικά με τις προτιμήσεις και τα έξοδα των χρηστών και την ανάλυση τους. Αυτό γίνεται για σκοπούς μελλοντικής ανάπτυξης της εφαρμογής καθώς επίσης και για ερευνητικούς σκοπούς ούτως ώστε να καθοριστούν καλύτερα οι προτιμήσεις και τα λεγόμενα trends της εποχής(προτιμήσεις ανά καιρούς οι οποίες επηρεάζονται από εξωγενείς παράγοντες, όπως παραδείγματος χάρη ένας χρήστης ακολουθεί τις αγοραστικές προτιμήσεις κάποιου προτύπου του πχ κάποιου τραγουδιστή).

Λειτουργία 4: Αλλαγή Κωδικού Χρήστη

Ο Διαχειριστής / Διαχειριστές έχει τη δυνατότητα να αλλάξει τον κωδικό του χρήστη εάν αυτό του ζητηθεί από τον τελευταίο ή εάν ο ίδιος το κρίνει απαραίτητο(σε περίπτωση που υποπέσει στην αντίληψη του ότι ο λογαριασμός έχει εκτεθεί σε τρίτο πρόσωπο).

3.2.6 Αυτοματοποιημένες Λειτουργίες Εφαρμογής

Λειτουργία 1: Απαντήσεις στον Χρήστη

Το σύστημα πρέπει να δίνει πιθανές επιλογές διάθεσης των χρημάτων του χρήστη όταν αυτό ζητηθεί από τον τελευταίο.

Λειτουργία 2: Ενημέρωση Προϋπολογισμού

Το σύστημα πρέπει να κρατά ενήμερο το τρέχον ισοζύγιο του χρήστη ούτως ώστε ο τελευταίος να είναι πάντα ενήμερος για το διαθέσιμο ποσό που μπορεί να ξοδέψει.

Λειτουργία 3: Ενημέρωση Προτιμήσεων Χρήστη

Το σύστημα πρέπει να διατηρεί βάση με τις προτιμήσεις του χρήστη για να μπορεί να του δώσει καλύτερες προτάσεις και επίσης για να μπορεί ο χρήστης να ανατρέξει σε αυτές εάν ο ίδιος το επιθυμεί.

Κεφάλαιο 4

ΓΟΡΓΙΑΣ

4.1 Εισαγωγή	12
4.2 Το Σύστημα Γοργίας	12

4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα ασχοληθούμε με το σύστημα ΓΟΡΓΙΑΣ. Πιο συγκεκριμένα θα αναλύσουμε τις δυνατότητες του συγκεκριμένου συστήματος για να εξηγήσουμε πως εμείς τις έχουμε εκμεταλλευτεί για να υλοποιήσουμε το δικό μας σύστημα.

4.2 Το Σύστημα Γοργίας

Ο Γοργίας είναι ένα γενικό πλαίσιο επιχειρηματολογίας που συνδυάζει τις ιδέες της λογικής της προτίμησης και της απαγωγής με τρόπο που διατηρεί τα οφέλη και των δύο. Μπορεί να αποτελέσει τη βάση της Λογικής σχετικά με τις προσαρμόσιμες πολιτικές προτιμήσεων ενόψει ελλειπών πληροφοριών από δυναμικά και εξελισσόμενα περιβάλλοντα. Για παράδειγμα, κατά την αγορά ενός αυτοκινήτου, κάποιος μπορεί να προτιμά ορισμένα χαρακτηριστικά έναντι των άλλων. κατά τον προγραμματισμό, η τήρηση ορισμένων προθεσμιών μπορεί να είναι πιο σημαντική από την ικανοποίηση άλλων. Κάνοντας αναφορά στο δικό μας σύστημα κάποια άτομα έχουν διαφορετικές προτιμήσεις ως προς την διάθεση των χρημάτων τους από άλλα. Για παράδειγμα κάποιος μπορεί να δίνει περισσότερη έμφαση στην κάλυψη των υποχρεώσεων του(δάνεια, λογαριασμοί, ενοίκια κτλ.) ενώ κάποιος άλλος να δίνει έμφαση στη διασκέδαση του.

4.1.1 Prolog

Η Prolog είναι μια λογική γλώσσα προγραμματισμού που σχετίζεται με την τεχνητή νοημοσύνη και την υπολογιστική γλωσσολογία. Πρόκειται για μια δηλωτική γλώσσα προγραμματισμού, όπου η λογική του προγράμματος εκφράζεται με όρους σχέσεων που εκπροσωπούνται ως γεγονότα και κανόνες. Για να υλοποιηθεί ένας υπολογισμός γίνεται εκτέλεση ενός ερωτήματος πάνω στις σχέσεις που έχουμε προαναφέρει πιο πάνω.

4.1.2 Υλοποίηση συστημάτων ΓΟΡΓΙΑ σε Prolog

Κάθε αρχείο ξεκινά με τις δύο πιο κάτω προτάσεις.

```
:-compile ('../lib/Gorgias.pl').
```

```
:-compile ('../ext/lpwnf.pl').
```

Η πρώτη γραμμή φορτώνει το σύστημα του ΓΟΡΓΙΑ ενώ η δεύτερη φορτώνει μια συλλογή από κανόνες που καθορίζουν μια σχέση προσόντων μεταξύ των επιχειρημάτων που εκμεταλλεύεται η σχέση επίθεσης για να κωδικοποιήσει τη σχετική ισχύ των επιχειρημάτων.

Για την υλοποίηση του κώδικα χρησιμοποιούμε όρους Prolog που σχηματίζονται από σύμβολα κατηγοριοποίησης για να υποδηλώνουν κανόνες, συγκρούσεις και προτιμήσεις μεταξύ των κανόνων. Η σύνταξη τέτοιων κανόνων έχει ως εξής:

Rule (Label, Head, Body)

Όπου το κεφάλι (Head) αποτελεί ένα literal, το σώμα (Body) είναι μια λίστα από literals και η ετικέτα (Label) είναι ένας σύνθετος όρος που αποτελείται από ένα όνομα κανόνα και επιλεγμένες μεταβλητές από το κεφάλι και το σώμα.

Παράδειγμα:

```
rule (r1(X), fly(X), [bird(X)]).
```

```
rule (r2(X), neg(fly(X)), [penguin(X)]).
```

```
rule (pr1(X), prefer(r2(X), r1(X)), []).
```

Ο πιο πάνω κανόνας $r1(X)$ υποδηλώνει ότι κάτι πετά εάν είναι πουλί ενώ ο κανόνας $r2(X)$ υποδηλώνει ότι όταν κάτι δεν πετά είναι πικουίνος. Ο κανόνας $pr1(X)$ υποδηλώνει ότι ο κανόνας $r2(X)$ έχει μεγαλύτερη προτεραιότητα από τον κανόνα $r1(X)$ και άρα εάν κάτι είναι και πουλί και πικουίνος, η δήλωση «δεν πετά» θα υπερισχύσει.

Ο υπολογισμός ενός αποδεκτού επιχειρήματος είναι μια διεμπλοκή δύο φάσεων: Στην πρώτη φάση ένας στόχος περιορίζεται σε ένα κλειστό σύνολο που αποδεικνύει το στόχο. Στη συνέχεια, αυτό το αρχικό επιχείρημα επεκτείνεται με κατάλληλες άμυνες για κάθε επίθεση εναντίον του αρχικού συνόλου. Ωστόσο, μετά την ανάπτυξη του αρχικού επιχειρήματος, θα προέκυπταν νέες συγκρούσεις και έτσι το σύστημα επαναλαμβάνει ολόκληρη τη διαδικασία έως ότου δεν υπάρξει υπεράσπιση για επίθεση ενάντια στο βασικό επιχείρημα (δηλαδή αδυναμία εύρεσης αποδεκτού συνόλου) ή μέχρι να μην υπάρξουν άλλες επιθέσεις (δηλαδή την επιτυχή εξαγωγή ενός αποδεκτού επιχειρήματος).

Οι ερωτήσεις υποβάλλονται στο σύστημα ως εξής:

Prove(Goals, Delta).

Όπου οι Στόχοι (Goals) είναι μια λίστα από θετικά ή αρνητικά literals, ενώ το Delta είναι ένα αποδεκτό όρισμα για το δεδομένο ερώτημα.

4.1.3 Παράδειγμα Υλοποίησης Συστήματος σε ΓΟΡΓΙΑ

Σε αυτή την υπό-ενότητα θα χρησιμοποιήσουμε μερικές από τις γραμμές κώδικα που έχουμε παραθέσει πιο κάτω στο Παράρτημα Α για να εξηγήσουμε την χρήση του ΓΟΡΓΙΑ στο δικό μας σύστημα.

Οι πιο κάτω αποτελούν δύο από τους είκοσι βασικούς κανόνες του συστήματος, οι οποίοι αντιστοιχούν σε τρόπους με τους οποίους ένας χρήστης μπορεί να διαθέσει τα χρήματά του, και τις αρνήσεις τους.

```
rule(r1(X), formal_clothes(X), []).
```

```
rule(r2(X), neg(formal_clothes(X)), []).
```

Το επόμενο είδος κανόνα που χρησιμοποιείται αποτελεί τις βασικές προτιμήσεις ανά κατηγορία. Δηλαδή όπως βλέπουμε στους δύο πιο κάτω κανόνες εάν ο χρήστης προτιμά να ξοδεύει λεφτά σε ρούχα, τότε αυτόματα ο κανόνας $r1(X)$ ο οποίος αναφέρεται σε ρούχα παίρνει μεγαλύτερη προτεραιότητα από τον κανόνα $r2(X)$, ο οποίος είναι η άρνηση του προηγούμενου κανόνα. Οπότε, ουσιαστικά ο κανόνας $npr1(X)$ υποδηλώνει ότι εάν ο χρήστης δηλώσει ότι προτιμά να ξοδεύει χρήματα σε ρούχα, τότε $r1(X) > r2(X)$. Επίσης ο κανόνας $r6(X)$ παίρνει προτεραιότητα από τον κανόνα $r5(X)$, καθώς ο τελευταίος δεν σχετίζεται με ρούχα και ως εκ τούτου η άρνηση του παίρνει προτεραιότητα. Δηλαδή ο $npr3(X)$ υποδηλώνει ότι εάν ο χρήστης προτιμά να ξοδεύει τα χρήματά του σε ρούχα τότε προτιμά να μην πηγαίνει σινεμά($r6(X)$) από το να πηγαίνει σινεμά($r5(X)$) το οποίο εμπίπτει στην κατηγορία της διασκέδασης.

```
rule(npr1(X), prefer(r1(X), r2(X)), []):- clothes.
```

```
rule(npr3(X), prefer(r6(X), r5(X)), []):- clothes.
```

Το επόμενο είδος κανόνα το οποίο βλέπουμε είναι οι ειδικές προτεραιότητες, οι οποίες στηρίζονται στις αποκλειστικές προτιμήσεις των χρηστών.

```
rule(spr1(X), prefer(r6(X), r5(X)), []):- entertainment, company(alone).
```

```
rule(spr2(X), prefer(spr1(X), npr13(X)), []):- entertainment, company(alone).
```

Ο κανόνας spr1(X) υποδηλώνει ότι όταν ο χρήστης είναι μόνος και θέλει να διασκεδάσει δεν θέλει να πηγαίνει σινεμά. Αυτός ο κανόνας έρχεται σε αντιπαράθεση με τον κανόνα npr13(X) ο οποίος υποδηλώνει ότι όταν κάποιος θέλει να διασκεδάσει μπορεί να πάει σινεμά. Ως εκ τούτου, δημιουργείται ο κανόνας spr2(X) ο οποίος δίνει μεγαλύτερη προτεραιότητα στον κανόνα spr1(X) από τον κανόνα npr13(X) όταν το σύστημα παίρνει ως είσοδο ότι το άτομο θέλει να διασκεδάσει αλλά είναι μόνο.

Κεφάλαιο 5

Τεχνολογίες

5.1 Εισαγωγή	15
5.2 PHP	15
5.3 Javascript	16
5.4 HTML/CSS	16
5.5 Βάσεις Δεδομένων - SQL	16
5.6 Docker	17

5.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται παρουσίαση των γλωσσών προγραμματισμού και των υπόλοιπων εργαλείων ανάπτυξης λογισμικού τα οποία έχουν χρησιμοποιηθεί για την ανάπτυξη και υλοποίηση του υφιστάμενου συστήματος. Οι πιο κάτω τεχνολογίες έχουν επιλεγεί μετά από έρευνα καθώς έπρεπε να λάβουμε υπόψη τις απαιτήσεις των προδιαγραφών που θέσαμε, καθώς επίσης και τις απαιτήσεις των σύγχρονων διαδικτυακών εφαρμογών, οι οποίες είναι εκ των ων ουκ άνευ για την βέλτιστη και εργονομική λειτουργία του συστήματος μας.

5.2 PHP

Η γλώσσα προγραμματισμού PHP (Hypertext Preprocessor) είναι μια γλώσσα γενικής χρήσης της οποίας ο αρχικός σκοπός ήταν η δημιουργία ιστοσελίδων, διαδικτυακών εφαρμογών και υπηρεσιών κοινωνικών δικτύων. Στο τρέχον σύστημα, η συγκεκριμένη γλώσσα δουλεύει στο υπόβαθρο και ως εκ τούτου οι λειτουργίες τις δεν είναι ορατές σε ένα απλό χρήστη. Χρησιμοποιείται για την επικοινωνία της ιστοσελίδας μας με τη βάση δεδομένων καθώς επίσης και για την εκτέλεση διαφόρων εργασιών σχετικά με τις επιλογές/προτιμήσεις που καταχωρεί ο χρήστης, την μετάφραση τους σε κανόνες ΓΟΡΓΙΑ και έπειτα την μετάφραση των αποτελεσμάτων σε προτάσεις προς τον χρήστη σε μορφή HTML κώδικα.

5.3 JavaScript

Η γλώσσα προγραμματισμού JavaScript είναι μία δυναμική γλώσσα η οποία εφαρμόζει τις έννοιες του αντικειμενοστραφούς προγραμματισμού και χρησιμοποιείται κυρίως για την δημιουργία αλληλεπιδραστικών ιστοσελίδων και διαδικτυακών εφαρμογών. Αυτή είναι και η μεγαλύτερη διαφορά της με την HTML η οποία έχει τη δυνατότητα να υλοποιεί μόνο στατικές ιστοσελίδες.

5.3.1 Vue.js

Η βιβλιοθήκη Vue.js είναι μια δομή(Framework) της JavaScript που χρησιμοποιείται για την δημιουργία και ανάπτυξη δυναμικών εφαρμογών. Η διαφορά μιας δομής(Framework) από άλλες κοινές βιβλιοθήκες είναι το γεγονός πως έχει σχεδιαστεί για να διαχωρίζει μια διαδικτυακή εφαρμογή σε ορθογώνιες μονάδες έτσι ώστε να βελτιώνει την ποιότητα του κώδικα και την συντήρηση της εφαρμογής. Μία από τις κύριες τις λειτουργίες είναι η ανάπτυξη single-page applications(SPAs), δηλαδή διαδικτυακών εφαρμογών-ιστοσελίδων οι οποίες αλληλοεπιδρούν με τους χρήστες κάνοντας δυναμική επανεγγραφή στον εαυτό αντί να φορτώνουν συνεχώς νέες σελίδες από κάποιο εξυπηρετητή(server). Με αυτό το τρόπο η ανταπόκριση της σελίδας βελτιώνεται πολύ σημαντικά προσδίδοντας στην καλύτερη εμπειρία του χρήστη.

5.4 HTML / CSS

Οι γλώσσες HTML και CSS είναι άρρηκτα συνδεδεμένες μεταξύ τους και για αυτό γίνεται αναφορά στις δύο μαζί.

Η HTML(HyperText Markup Language) είναι μια γλώσσα προγραμματισμού υπεύθυνη για τη δημιουργία στατικών ιστοσελίδων, όπως έχουμε αναφέρει και πιο πάνω. Η συγκεκριμένη γλώσσα υλοποιεί τις λειτουργίες της ιστοσελίδας μας οι οποίες είναι ορατές από τον τελικό χρήστη, με άλλα λόγια είναι η γλώσσα με την οποία υλοποιούμε την όψη της σελίδας μας.

Η CSS(Cascading Style Sheets) είναι μια γλώσσα προγραμματισμού, η οποία σε συνδυασμό με την HTML χρησιμοποιείται για τη μορφοποίηση των ιστοσελίδων. Ουσιαστικά μας επιτρέπει να ομορφαίνουμε τις ιστοσελίδες / διαδικτυακές εφαρμογές και να τις κάνουμε πιο φιλικές ως προς τον τελικό χρήστη(user friendly).

5.5 Βάσεις Δεδομένων - SQL

Οι Βάσεις Δεδομένων αποτελούν ολοκληρωμένα συστήματα διαχείρισης, επεξεργασίας και αποθήκευσης πληροφοριών. Οι κύριοι λόγοι που θα χρησιμοποιήσουμε αυτή την τεχνολογία είναι η πιο ασφαλής αποθήκευση των δεδομένων των χρηστών, η καλύτερη και ευκολότερη

διαχείριση των δεδομένων τους και η ευκολότερη αναπαράσταση τους. Επίσης διατηρείται μια ομοιομορφία στα δεδομένα, κάτι που διευκολύνει την επεξεργασία και διαχείριση τους. Η Τεχνολογία SQL (Structured Query Language) χρησιμοποιείται για επικοινωνία με μια Βάση Δεδομένων. Μέσα από την SQL ενημερώνουμε τα δεδομένα των χρηστών και γενικά εκτελούμε διεργασίες επικοινωνίας της βάσης δεδομένων μας με την διαδικτυακή εφαρμογή που έχουμε αναπτύξει.

5.6 Docker

Το Docker είναι ένα εργαλείο σχεδιασμένο για να διευκολύνει τη δημιουργία, την ανάπτυξη και την εκτέλεση εφαρμογών χρησιμοποιώντας δοχεία (Containers). Τα δοχεία επιτρέπουν σε έναν προγραμματιστή να συσκευάσει μια εφαρμογή με όλα τα μέρη που χρειάζονται, όπως βιβλιοθήκες και άλλες εξαρτήσεις, και να τα αποστείλει όλα ως ένα πακέτο. Με τον τρόπο αυτό, χάρη στο δοχείο, ο προγραμματιστής μπορεί να είναι βέβαιος ότι η εφαρμογή θα τρέξει σε οποιοδήποτε άλλο μηχάνημα Linux/Windows, ανεξάρτητα από τυχόν προσαρμοσμένες ρυθμίσεις που μπορεί να έχει αυτό το μηχάνημα, κάτι που μπορεί να διαφέρει από το μηχάνημα που χρησιμοποιείται για τη γραφή και τον έλεγχο του κώδικα.

Ουσιαστικά, το Docker δουλεύει με παρόμοιο τρόπο όπως μια εικονική μηχανή, αλλά αντίθετα από μια εικονική μηχανή, αντί να δημιουργήσει ένα ολόκληρο εικονικό λειτουργικό σύστημα, το Docker επιτρέπει στις εφαρμογές να χρησιμοποιούν τον ίδιο πυρήνα Linux/Windows με το σύστημα στο οποίο εκτελούν και απαιτεί μόνο να στέλνονται εφαρμογές με πράγματα που δεν εκτελούνται ήδη στον κεντρικό υπολογιστή. Αυτό δίνει σημαντική ενίσχυση της απόδοσης και μειώνει το μέγεθος της εφαρμογής.

Για αυτούς τους λόγους έχουμε επιλέξει την συγκεκριμένη τεχνολογία για να υλοποιήσουμε το hosting της σελίδας και της βάσης δεδομένων μας.

Κεφάλαιο 6

Αρχιτεκτονική Συστήματος

6.1 Εισαγωγή	19
6.2 Κύκλος Ζωής Λογισμικού	19
6.3 Μοντελοποίηση Συστήματος	25
6.4 Υλοποίηση Συστήματος	26

6.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα δούμε τον τρόπο με τον οποίο δημιουργήσαμε την διαδικτυακή αυτή εφαρμογή. Συγκεκριμένα θα δούμε πως ξεκίνησε ως ιδέα και σταδιακά πως φτάσαμε μέχρι την υλοποίηση του συστήματος μας, ακολουθώντας τα κριτήρια και τις προδιαγραφές που θέσαμε αρχικά, αλλά και τα προβλήματα και τις νέες ανάγκες οι οποίες έχουν προκύψει κατά τη διάρκεια της υλοποίησης του συστήματος.

6.2 Κύκλος Ζωής Λογισμικού

Ο κύκλος Ζωής Λογισμικού είναι η διαδικασία ανάπτυξης του λογισμικού, η οποία ξεκινά από την αρχική ιδέα και διαμέσου διαφόρων σταδίων καταλήγει μέχρι και την απόσυρση του λογισμικού. Τα κυριότερα στάδια είναι η ανάθεση του συστήματος σε μια ομάδα ειδικών, η δημιουργία και ανάλυση των προδιαγραφών του, η σχεδίαση και ανάπτυξη κώδικα και αφότου ολοκληρωθούν επιτυχώς τα πιο πάνω, η διάθεση του στον τελικό χρήστη. Επίσης ίσως τα σημαντικότερα στάδια του κύκλου ζωής του λογισμικού αφότου δοθεί για ευρεία χρήση, είναι η συντήρηση και αναβάθμιση του ανάλογα με τις μεταβαλλόμενες ανάγκες των χρηστών μέχρι και την τελική του απόσυρση.

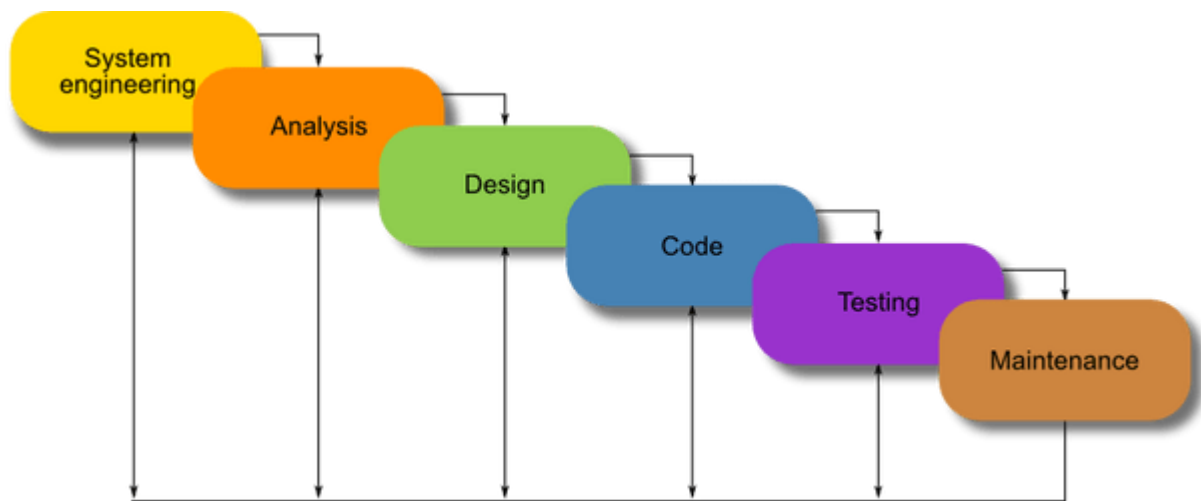
6.2.1 Μοντέλα Κύκλου Ζωής

Τα Μοντέλα Κύκλου Ζωής είναι οι μεθοδολογίες οι οποίες έχουν αναπτυχθεί για την ανάπτυξη και υλοποίηση ενός συστήματος. Κάθε μεθοδολογία διαθέτει κάποια διακριτά στάδια μέσα από τα οποία επιτυγχάνεται η ολοκλήρωση ενός συστήματος. Τα πλείστα από αυτά τα μοντέλα διαθέτουν κοινά στάδια αλλά διαφέρουν στην μεθοδολογία και σειρά υλοποίησης τους. Στη συνέχεια αυτού του κεφαλαίου θα παραθέσουμε τα κυριότερα από αυτά ούτως ώστε να μπορέσουμε να εξηγήσουμε και να δικαιολογήσουμε την μεθοδολογία την οποία ακολουθήσαμε για την υλοποίηση του τρέχοντος συστήματος.

6.2.1.1 Μοντέλο Καταρράκτη

Το Μοντέλο Καταρράκτη υλοποιείται σε στάδια τα οποία βρίσκονται σε σειρά το ένα με το άλλο και κάθε στάδιο είναι εξαρτώμενο από το αμέσως προηγούμενο του, δηλαδή κάθε στάδιο δεν μπορεί να ξεκινήσει εάν δεν ολοκληρωθούν τα απαραίτητα παραδοτέα του προηγούμενου σταδίου. Επίσης σε αυτό το μοντέλο οι απαιτήσεις του συστήματος αποφασίζονται στην αρχή της διαδικασίας, κάνοντας τυχόν αλλαγές τους στην πορεία υλοποίησης του συστήματος πολύ δύσκολο να επιτευχθούν. Χρησιμοποιείται σε κάποια πεδία σχεδιασμού μηχανικής(engineering design) και κυρίως σε μεγάλα συστήματα τα οποία σχετίζονται με υπάρχον συστήματα λογισμικού. Τα στάδια του μπορούν διακριθούν κατά κύριο λόγο(πολλές φορές κάποια από αυτά μπορούν να ενοποιηθούν, δίνοντας μας έτσι λιγότερα στάδια όπως βλέπουμε και στο σχεδιάγραμμα πιο κάτω) ως εξής:

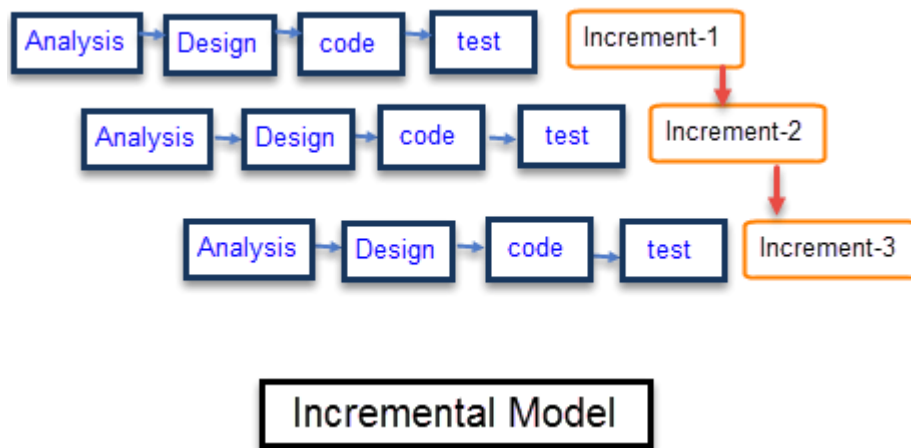
- (i) Σύλληψη Ιδέας
- (ii) Αρχικοποίηση
- (iii) Ανάλυση Συστήματος
- (iv) Σχεδίαση Συστήματος
- (v) Υλοποίηση Συστήματος
- (vi) Δοκιμή Συστήματος
- (vii) Διάθεση Συστήματος
- (viii) Συντήρηση Συστήματος



(Σχήμα 5.1 Μοντέλο Καταρράκτη)

6.2.1.2 Αυξητικό Μοντέλο

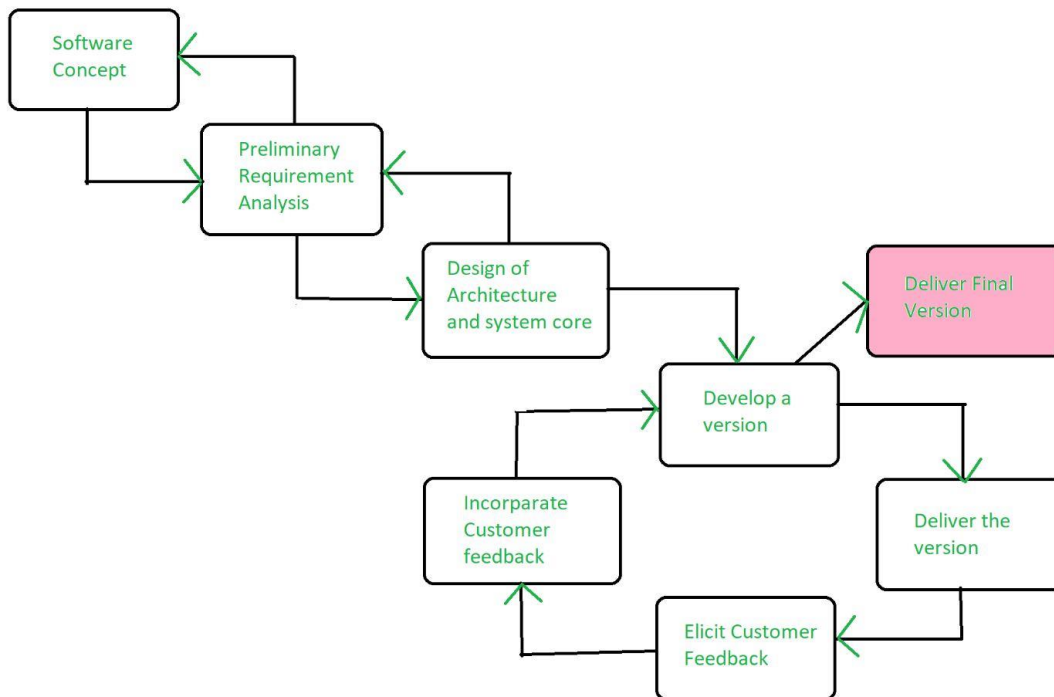
Σε αυτό το μοντέλο, το σύστημα παραδίδεται σε εκδόσεις και μπορεί υλοποιηθεί με 3 τρόπους. Πρώτον, κάθε έκδοση περιλαμβάνει σχεδόν όλα τα στάδια του Μοντέλου του Καταρράκτη και πιο συγκεκριμένα σε κάθε έκδοση βλέπουμε τις φάσεις της Σχεδίασης, Υλοποίησης, Ελέγχου και Εγκατάστασης-Χρήσης. Αυτή η διαδικασία συνεχίζεται με κάθε νέα έκδοση να προσαυξάνεται στην προηγούμενη μέχρι να ολοκληρωθεί το σύστημα. Δεύτερο, οι συνολικές απαιτήσεις καθορίζονται πριν την υλοποίηση μικρών ατομικών προσαυξήσεων τύπου μοντέλου Καταρράκτη. Τρίτον, η αρχικές έννοιες του λογισμικού, η ανάλυση απαιτήσεων και η σχεδίαση του πυρήνα της αρχιτεκτονικής του συστήματος καθορίζονται από Μοντέλο Καταρράκτη και ακολουθείται από προσαυξήσεις που εκτελούνται μέχρι την εγκατάσταση της τελικής έκδοσης. Αυτό το μοντέλο καλύπτει κάποιες από τις αδυναμίες του Μοντέλου Καταρράκτη καθώς είναι πιο ευέλικτο στην αντιμετώπιση αδυναμιών που μπορεί να παρουσιάσει το σύστημα αφού μπορούν να εντοπιστούν αρκετά νωρίς. Παρόλα αυτά οι απαιτήσεις του δεν παύουν να παραμένουν σχετικά σταθερές από τα αρχικά του στάδια.



(Σχήμα 5.2 Αυξητικό Μοντέλο)

6.2.1.3 Εξελικτικό Μοντέλο

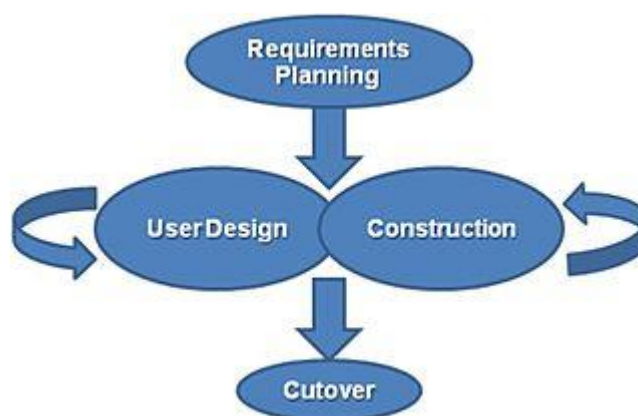
Το Εξελικτικό Μοντέλο αποτελεί, ουσιαστικά, μια παραλλαγή του Αυξητικού μοντέλου. Η μεγάλη του διαφορά και πλεονέκτημα είναι ότι είναι εύκολο να αλλαχθούν οι απαιτήσεις κατά την διάρκεια της ανάπτυξης του συστήματος κάνοντας το έτσι πολύ πιο ευέλικτο. Αυτό το μοντέλο διαχωρίζει τον κύκλο ανάπτυξης σε μικρότερα αυξητικά μοντέλα καταρράκτη, στα οποία έχουν πρόσβαση οι χρήστες μετά την ολοκλήρωσή τους. Έτσι, με την ολοκλήρωση κάθε μικρού μοντέλου αυτοί παρέχουν ανατροφοδότηση την οποία εκμεταλλεύεται η ομάδα ανάπτυξης του συστήματος για να προχωρήσει στην ανάπτυξη του επόμενου βελτιωμένου μοντέλου. Χρησιμοποιείται κυρίως σε μεγάλα έργα, όπου οι χρήστες θέλουν να χρησιμοποιούν τις κύριες λειτουργίες του συστήματος πριν ακόμα αυτό έχει υλοποιηθεί πλήρως ή σε αντικειμενοστραφής ανάπτυξη λογισμικού επειδή το σύστημα έχει τη δυνατότητα να διαχωριστεί σε πιο μικρές μονάδες, δηλαδή τα αντικείμενα που χρησιμοποιούνται κατά τη διάρκεια του προγραμματισμού.



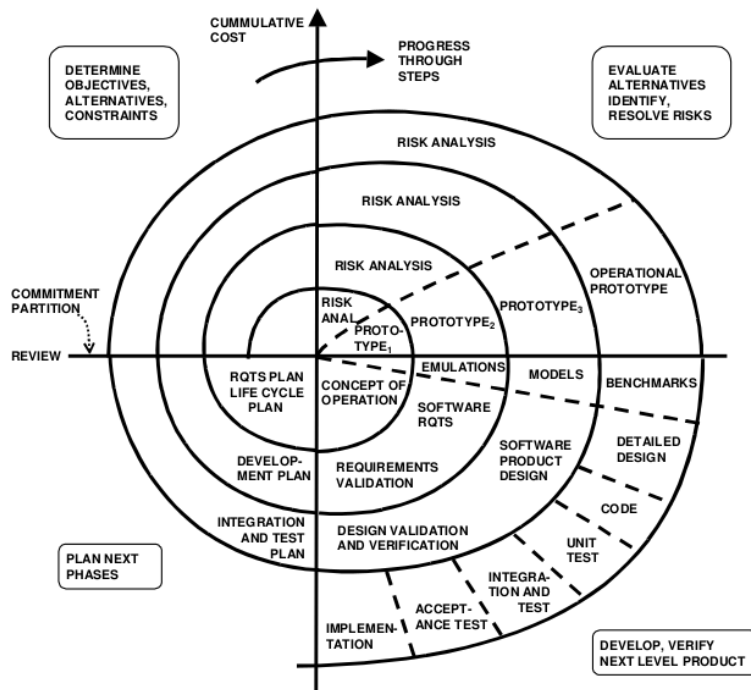
(Σχήμα 5.3 Εξελικτικό Μοντέλο)

6.2.1.4 Σπειροειδές Μοντέλο

Το Σπειροειδές Μοντέλο συνδυάζει μερικές βασικές πτυχές του Μοντέλου Καταρράκτη μαζί με Ανάπτυξη Γρήγορων Εφαρμογών(Rapid-application development), ευρέως γνωστές ως RAD. Δίνει έμφαση σε επαναληπτική ανάλυση ρίσκου, μεθοδολογία η οποία είναι ειδικά κατάλληλη για μεγάλου εύρους πολύπλοκα συστήματα. Αυτό είναι κάτι το οποίο οι περισσότερες μεθοδολογίες έχουν παραμελήσει. Δυστυχώς όμως έχει περισσότερο κόστος από άλλα μοντέλα και μεγαλύτερη πολυπλοκότητα και ως εκ τούτου αποφεύγεται η χρήση του στην ανάπτυξη μικρότερων συστημάτων.



(Σχήμα 5.4 Rapid-Application Development)



(Σχήμα 5.5 Σπειροειδές Μοντέλο)

6.2.1.5 Γενική Αναφορά σε άλλα Μοντέλα

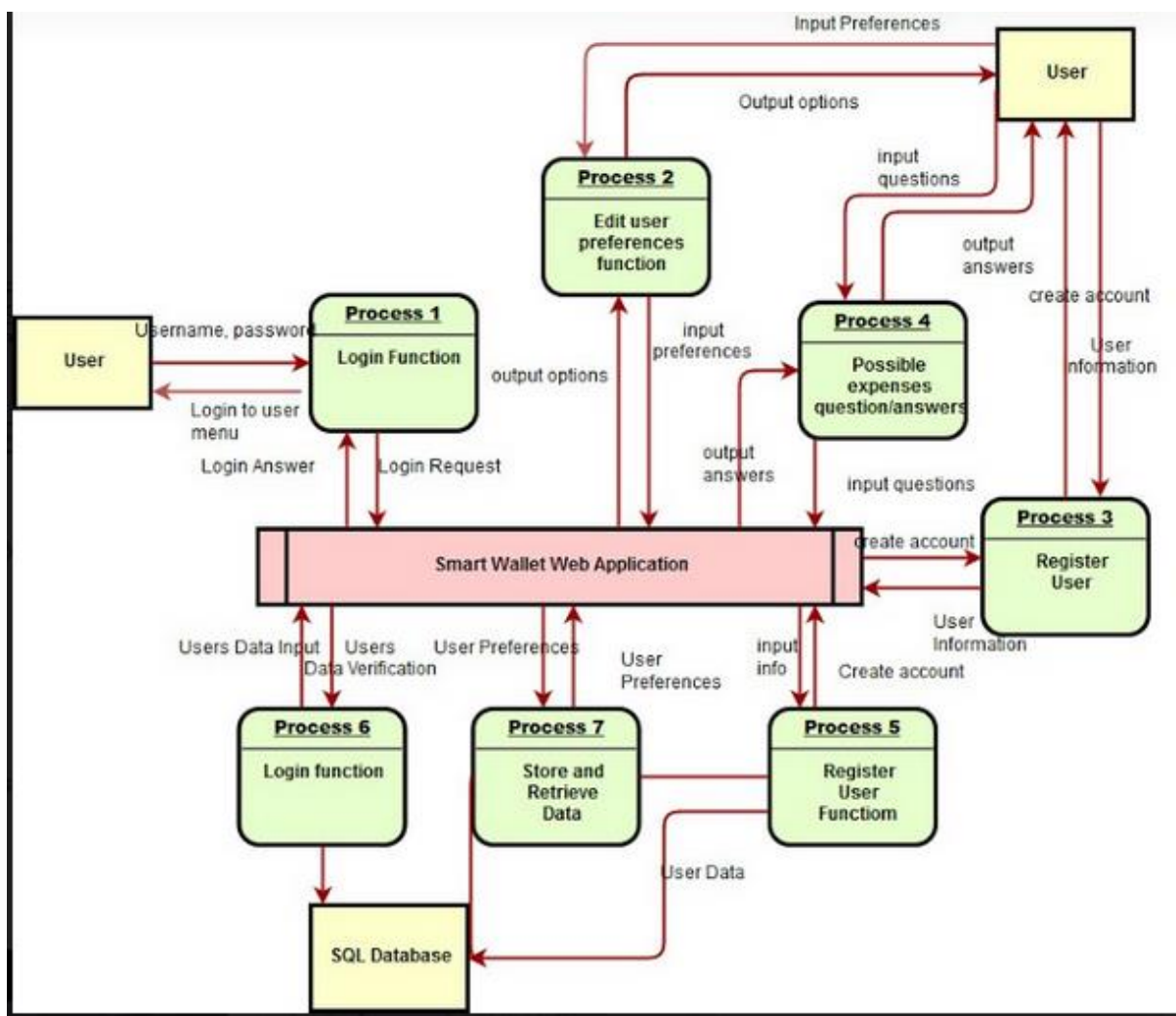
Φυσικά, πέραν των προαναφερθεισών μοντέλων υπάρχουν και άλλα όπως είναι το Μοντέλο του Χάους το οποίο ακολουθεί τον κανόνα επίλυσης του πιο σημαντικού προβλήματος πρώτα, το Μοντέλο V(V-Model) το οποίο αποτελεί μια συγκεκριμένη έκδοση του Μοντέλου Καταρράκτη, το Μοντέλο Συμπεριφοριστικής Ανάπτυξης, το Μοντέλο του Ακραίου Προγραμματισμού(Extreme Programming) και άλλα πολλά, τα οποία όμως θα ήταν αδύνατο να εφαρμοστούν στην ανάπτυξη του δικού μας συστήματος για πρακτικούς ή και οικονομικούς λόγους.

6.2.2 Αξιολόγηση Μοντέλων

Όπως βλέπουμε από τα πιο πάνω μοντέλα υπάρχουν διάφοροι τρόποι ανάπτυξης λογισμικού. Επίσης κανένα μοντέλο δεν μπορεί να μας εγγυηθεί την επιτυχή ολοκλήρωση του συστήματος ή την βέλτιστη διαδικασία ανάπτυξης του. Καθώς το σύστημα μας αποτελεί δική μας ιδέα έχουμε θέσει τις κύριες προδιαγραφές και απαιτήσεις εξ αρχής αλλά έχουμε την δυνατότητα να τροποποιήσουμε ορισμένες από αυτές κατά τη διάρκεια της ανάπτυξης του συστήματος χωρίς όμως να επηρεάσουμε τα βασικά κριτήρια τα οποία θέσαμε, και γύρω από τα οποία περιστρέφεται η ιδέα του συστήματος μας. Πέραν από την βασική ιδέα του συστήματος μας, μερικά σημαντικά κριτήρια είναι, αυτό να παρέχει καλή διαθεσιμότητα, αξιοπιστία και ασφάλεια. Ακόμη πρέπει να είναι εύκολο και ευχάριστο όσον αφορά την χρήση του από οποιοδήποτε απλό χρήστη.

Μέσα από αυτά λοιπόν συμπεράναμε ότι για την ανάπτυξη του συστήματος μας θα έπρεπε να ακολουθήσουμε κάποια βήματα σε σειρά και αφότου παρουσιάσουμε ένα πρότυπο, να συνεχίσουμε να κάνουμε αλλαγές και να το αναπτύσσουμε σταδιακά. Επίσης η άμεση δοκιμή σε χρήστες, πέραν των μελών που αναπτύσσουν το σύστημα, δεν είναι απαραίτητη καθώς πρόκειται για μια καινούργια ιδέα, αλλά πρέπει να δοκιμαστεί από το ευρύ κοινό όταν αυτό βρίσκεται στα τελικά του στάδια.. Για αυτούς τους λόγους έχουμε επιλέξει ένα συνδυασμό Μοντέλου Καταρράκτη και Εξελικτικού Μοντέλου, κρατώντας έτσι κάποιες αρχικές βασικές προδιαγραφές αλλά διατηρώντας την ευελιξία να κάνουμε διάφορες αλλαγές και εν τέλει δοκιμές σε πραγματικό περιβάλλον επιπέδου παραγωγής.

6.3 Μοντελοποίηση Συστήματος



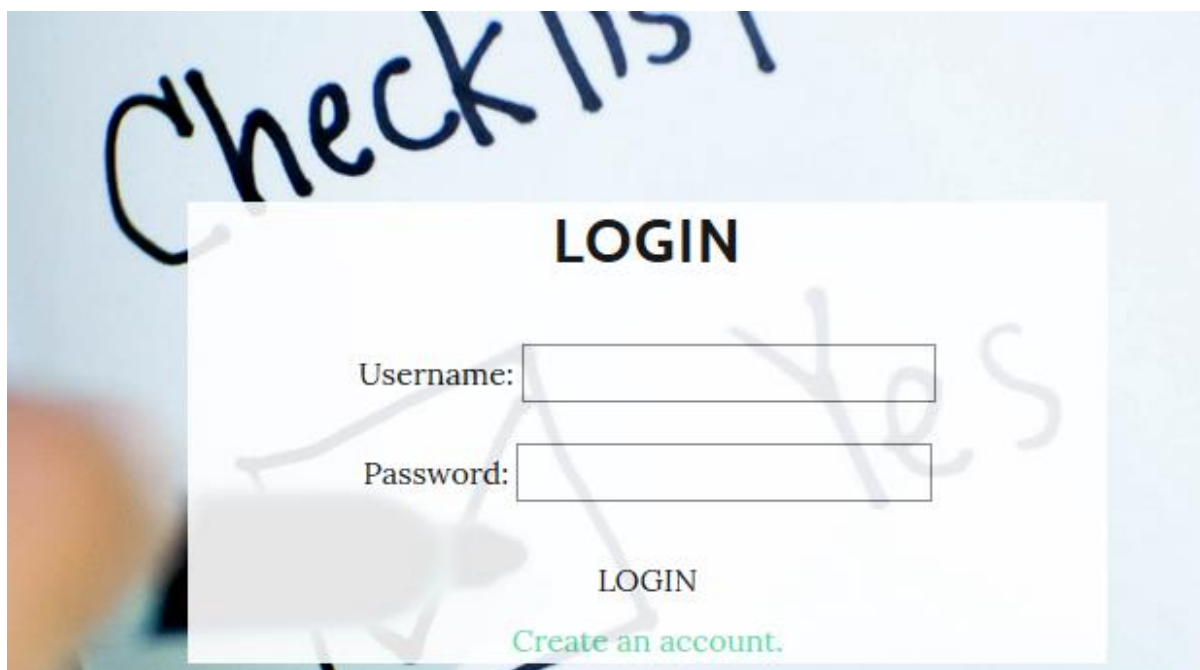
Στο πιο πάνω σχήμα φαίνονται οι κύριες λειτουργίες του συστήματος. Πιο συγκεκριμένα μέσα από τις ροές δεδομένων βλέπουμε τις κύριες λειτουργίες μεταξύ των χρηστών και της εφαρμογής και μεταξύ της εφαρμογής με τη βάση δεδομένων.

6.4 Υλοποίηση Συστήματος

Με την έναρξη του σταδίου της υλοποίησης της διαδικτυακής μας εφαρμογής, θέσαμε ως κύριο στόχο να επιτευχθούν όλοι οι στόχοι υλοποίησης τους οποίους θέσαμε αρχικώς. Σε αυτή την ενότητα θα δούμε ποιες από αυτές έχουν υλοποιηθεί πλήρως και που χρειάστηκε να προχωρήσουμε σε αλλαγές. Επίσης, θα δούμε πως αυτές υλοποιήθηκαν για να φτάσουμε στο τελικό αποτέλεσμα.

6.4.1 Ασφαλής Είσοδος χρήστη στο σύστημα / Εγγραφή

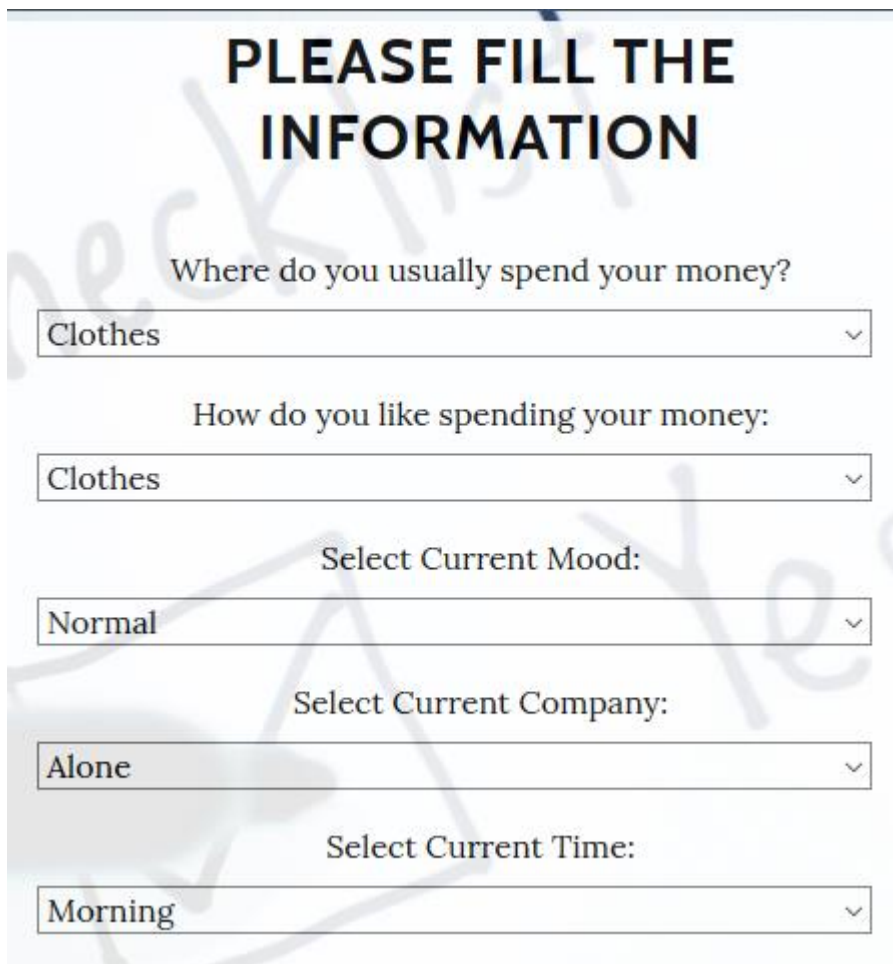
Οι χρήστες πρέπει να δημιουργήσουν προσωπικό λογαριασμό και για να έχουν πρόσβαση σε αυτόν πρέπει να καταχωρούν το μοναδικό τους username και κωδικό. Επίσης κατά τη δημιουργία του λογαριασμού ζητείται από αυτούς να καταχωρήσουν τα προσωπικά τους στοιχεία. Αυτά τα στοιχεία, μαζί με το username και τον κωδικό των χρηστών φυλάσσονται και διατηρούνται στη δική μας βάση δεδομένων.



(Εικόνα 6.1 Σελίδα Εισόδου)

6.4.2 Εισαγωγή προτιμήσεων / Αλλαγή προτιμήσεων

Οι χρήστες δίνουν ως είσοδο στο σύστημα τις γενικές τους προτιμήσεις. Δηλαδή πως προτιμούν να ξοδεύουν τα χρήματα τους συνήθως, εάν είναι μόνοι τους ή με παρέα, σε ποια χρονική περίοδο βρισκόμαστε, σε ποιο χρονικό σημείο της μέρας βρισκόμαστε και τι είδους διάθεση έχουν. Οι χρήστες μπορούν να τροποποιήσουν τις επιλογές τους με κάθε νέο αίτημα προς την εφαρμογή.



The screenshot shows a form with the title "PLEASE FILL THE INFORMATION". It contains five dropdown menus, each with a label and a selected option:

- Label: "Where do you usually spend your money?"
Selected option: "Clothes"
- Label: "How do you like spending your money:"
Selected option: "Clothes"
- Label: "Select Current Mood:"
Selected option: "Normal"
- Label: "Select Current Company:"
Selected option: "Alone"
- Label: "Select Current Time:"
Selected option: "Morning"

(Εικόνα 6.2 Λήψη Προτιμήσεων)

6.4.3 Εισηγήσεις του Συστήματος

Αφότου οι χρήστες εισαγάγουν τις προτιμήσεις τους, η εφαρμογή θα φυλάξει αυτές τις προτιμήσεις και θα χρησιμοποιήσει τον ΓΟΡΓΙΑ ούτως ώστε να επιστρέψει πιθανούς τρόπους διάθεσης των χρημάτων του χρήστη. Επίσης θα επιστρέψει πιθανούς λόγους για τους οποίους ένας χρήστης θα ήταν καλό να μην προχωρήσει σε κάποια έξοδα τη συγκεκριμένη χρονική στιγμή.

RECCOMENDATION RESULTS

- Why you should buy formal clothes:
 - 1.
- Why you should not buy formal clothes:
 1. We suggest you don't buy formal clothes
- Why you should buy casual clothes:
 - 1.
- Why you should not buy casual clothes:
 1. We suggest you don't buy casual clothes
- Why you should go to the cinema:
 - 1.
- Why you should not go to the cinema:
 1. Whats the point of watching a movie if you cant argue with others after you watch it, get some company to visit the movies
 2. You want to have fun but you are alone, cinema seems a little boring idea
 3. Whats the point of watching a movie if you cant argue with others after you watch it, get some company to visit the movies
 4. We advise not to go to the cinema.
- Why you should go for athletic activities:
 1. You like exercising in the morning

(Εικόνα 6.3 Επιστροφή εισηγήσεων)

Κεφάλαιο 7

Αξιολόγηση Συστήματος

7.1 Εισαγωγή	29
7.2 Έλεγχος Συστήματος	29
7.3 Αξιολόγηση - Αποτελέσματα	30

7.1 Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται αναφορά στην διαδικασία που ακολουθήσαμε για την διεξαγωγή ελέγχου και αξιολόγησης του συστήματος μας. Αρχικά, προχωρήσαμε οι ίδιοι σε διάφορους ελέγχους και δοκιμές του συστήματος μας. Μέσα από αυτές τις δοκιμές εντοπίσαμε αδυναμίες και προβλήματα τα οποία κρίναμε αναγκαία διορθωθούν πριν την αρχική διάθεση της εφαρμογής στους χρήστες. Έπειτα διαθέσαμε την εφαρμογή σε πραγματικούς χρήστες και συγκεντρώσαμε τις απόψεις τους μέσα από ένα ερωτηματολόγιο. Τέλος, προχωρήσαμε σε ορισμένες μικροδιορθώσεις, λαμβάνοντας υπόψη τις απόψεις των χρηστών και σημειώσαμε τις αλλαγές μεγαλύτερου εύρους που μας έχουν ζητηθεί για μελλοντική ανάπτυξη του συστήματος.

7.2 Έλεγχος Συστήματος

Όπως έχουμε προαναφέρει, ο έλεγχος του συστήματος ξεκίνησε πρώτα από εμάς. Αρχικά, δημιουργήσαμε υποθετικούς χρήστες υλοποιώντας μικρά απλά σενάρια για να εξακριβώσουμε την ορθή ανταπόκριση της εφαρμογής μας. Αφότου αυτά ολοκληρώθηκαν με επιτυχία προχωρήσαμε σε πιο σύνθετα σενάρια, τα οποία να ανταποκρίνονται σε πραγματικές καθημερινές συνθήκες και οικονομικές απαιτήσεις ενός σύγχρονου φυσιολογικού ατόμου. Επίσης, για να βεβαιωθούμε ότι οι προσομοιώσεις μας, βρίσκονται

όσο το δυνατόν πιο κοντά σε πραγματικό περιβάλλον επικοινωνίας ενός χρήστη με την εφαρμογή, χρησιμοποιήσαμε πραγματιστικές ερωτήσεις από την πλευρά του χρήστη ως προς την εφαρμογή. Παράλληλα, έχουμε εξερευνήσει και ακραίες περιπτώσεις για να διαπιστώσουμε την ορθή συμπεριφορά του συστήματος μας.

Αφότου ολοκληρώσαμε αυτές τις δοκιμές, προχωρήσαμε σε δοκιμές ως προς την ορθή και ομαλή λειτουργία της εφαρμογής, εξερευνώντας όλες επιλογές που προσφέρονται από την διαδικτυακή εφαρμογή.

Έπειτα, προχωρήσαμε σε έλεγχο της ασφάλειας του συστήματος μας. Ελέγξαμε τόσο την ασφάλεια σύνδεσης του χρήστη στην εφαρμογή, όσο και την ασφαλή σύνδεση και μεταφορά από και προς την Βάση Δεδομένων μας. Τέλος, ελέγξαμε την ορθή και ασφαλή αποθήκευση των δεδομένων στην Βάση μας, πριν να διαθέσουμε την εφαρμογή για χρήση από πραγματικούς χρήστες.

Ως τελευταίο βήμα του ελέγχου της εφαρμογής μας, δόθηκε σε πραγματικούς χρήστες για να το δοκιμάσουν και ζητήθηκε από αυτούς να απαντήσουν ένα σύντομο ερωτηματολόγιο ούτως ώστε να διαπιστώσουμε την απήχηση του συστήματος στον κόσμο. Ακόμη μέσα από το ερωτηματολόγιο ζητήθηκε από τους χρήστες να σχολιάσουν τυχόν προβλήματα που έχουν εντοπίσει και να μας παραθέσουν προτάσεις, τις οποίες οι ίδιοι θεωρούν ότι θα βελτίωνε την εφαρμογή.

7.3 Αξιολόγηση – Αποτελέσματα

Η αξιολόγηση του συστήματος έγινε μέσα από τη χρήση ερωτηματολογίου, το οποίο παρατίθεται αυτούσιο στη συνέχεια. Οι ερωτήσεις του ερωτηματολογίου έχουν ως κύριο σκοπό να δώσουν απαντήσεις ως προς την δομή της εφαρμογής, την ευχρηστία της και την ικανοποίηση των απαιτήσεων που θα μπορούσε να έχει ο κάθε χρήστης από μια εφαρμογή τέτοιου είδους. Περισσότερο από όλα αυτά όμως, θέλαμε να εξακριβώσουμε εάν οι χρήστες αντιλαμβάνονται την ιδέα πίσω από αυτή την εφαρμογή και εάν προτίθενται να την χρησιμοποιούν στην καθημερινή τους ζωή.

Ερωτηματολόγιο:

Questionnaire

1. Age:

Choose ▼

2. Gender

☐ Male

☐ Female

3. How did you find the graphical environment of the website?

	1	2	3	4	5	
Easy to use	☐	☐	☐	☐	☐	Difficult to use

4. How difficult was it to create an account?

	1	2	3	4	5	
Very Easy	☐	☐	☐	☐	☐	Very Difficult

5. How difficult was the communication with the application

	1	2	3	4	5	
Very Easy	☐	☐	☐	☐	☐	Very Difficult

6. Would you use the application in your day to day purchases

- ☐ NO
- ☐ Some of them
- ☐ Most of them
- ☐ Yes

7. How did you find the answers provided by the application?

- ☐ Irrelevant
- ☐ Slightly Irrelevant
- ☐ Slightly Relevant
- ☐ Relevant

8. How did you find the concept of the application?

Your answer

9. What would you change?

Your answer

10. What would you like to see added to the application?

Your answer

Μέσα από τις απαντήσεις που λάβαμε καταλήξαμε στα εξής συμπεράσματα:

1. Η πλειοψηφία των ατόμων κάτω των 41 ετών βρήκε την εφαρμογή αρκετά εύκολη στη χρήση ενώ τα άτομα μεγαλύτερης ηλικίας αντιμετώπισαν μερικές δυσκολίες, γεγονός που είναι αναμενόμενο λόγω και της έλλειψης τεχνογνωσίας.
2. Οι χρήστες ζήτησαν να προστεθούν κάποιες εικόνες στην εφαρμογή ούτως ώστε να γίνει πιο ευχάριστη στην όψη, κάτι το οποίο βρισκόταν στα αρχικά μας πλάνα και θα πραγματοποιηθεί πριν την τελική διάθεση της εφαρμογής στο κοινό.
3. Η πλειοψηφία των χρηστών απάντησε ότι τα αποτελέσματα που έλαβαν στις ερωτήσεις τους από την εφαρμογή ήταν σχετικές με τα ενδιαφέροντα τους ή αρκετά κοντά σε αυτές.
4. Όλοι οι χρήστες βρήκαν την όλη ιδέα της εφαρμογής ιδιαίτερα ενδιαφέρουσα και πιστεύουν ότι θα μπορούσαν να την εντάξουν και να την εκμεταλλευτούν στην καθημερινότητά τους.
5. Οι χρήστες έδωσαν διάφορες ιδέες με τις οποίες θα μπορούσαμε να εμπλουτίσουμε τις επιλογές της εφαρμογής και να την αναπτύξουμε έτσι ώστε να παρέχει περισσότερες υπηρεσίες και να καλύπτει μεγαλύτερο εύρος πιθανών τρόπων διάθεσης των χρημάτων τους.

Κεφάλαιο 8

Συμπεράσματα - Μελλοντική Εργασία

8.1 Εισαγωγή	35
8.2 Συμπεράσματα	35
8.3 Μελλοντική Εργασία	36

8.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα εξάγουμε τα συμπεράσματα μας μέσα από τις απαντήσεις που πήραμε από τους χρήστες. Επίσης θα κάνουμε αναφορά στην μελλοντική εργασία που μπορεί να γίνει. Η μελλοντική εργασία έχει καθοριστεί μέσα από τις δικές μας βλέψεις αλλά και μέσα από τις απαντήσεις που πήραμε από τους χρήστες στο ερωτηματολόγιο που τους παραθέσαμε.

8.2 Συμπεράσματα

Καταρχήν, μέσα από την θετική ανταπόκριση των χρηστών έχουμε πετύχει την πρώτη και κυριότερη απαίτηση την οποία θέσαμε στην αρχή αυτού του έργου. Δηλαδή την αποδοχή της ιδέας του Έξυπνου Πορτοφολιού και την πιθανή ενσωμάτωση του στην καθημερινή ζωή των χρηστών. Επίσης έχουμε πετύχει, κατά κύριο λόγο, την ευχρηστία της εφαρμογής μας, καθώς η πλειοψηφία των χρηστών ανέφεραν πως είναι αρκετά εύκολη στη χρήση. Ακόμη καταφέραμε να υλοποιήσουμε μια εφαρμογή η οποία προσφέρει αρκετά καλές εισηγήσεις, προτρέποντας τους να πάρουν ορισμένες επιλογές, χωρίς όμως να τις παίρνει η ίδια εφαρμογή για τους ίδιους ή να τους αναγκάζει να πάρουν μία συγκεκριμένη επιλογή.

8.3 Μελλοντική Εργασία

1. Πρώτα από όλα, θα προχωρήσουμε στην προσθήκη περισσότερων υποκατηγοριών όσο αφορά τους τρόπους διάθεσης των χρημάτων των χρηστών ούτως ώστε η εφαρμογή μας να μπορεί να τους δίνει περισσότερες επιλογές.
2. Δεύτερον, η εφαρμογή να παρέχει τη δυνατότητα σύνδεσης της με τραπεζικούς λογαριασμούς. Αυτή η εργασία έχει ζητηθεί από την πλειοψηφία των χρηστών και πέραν της σύνδεσης της εφαρμογής με τραπεζικά συστήματα απαιτεί την ενσωμάτωση περαιτέρω στρωμάτων ασφαλείας στην εφαρμογής μας ούτως ώστε να μπορούμε να εγγυηθούμε την ασφάλεια των προσωπικών τους στοιχείων και λογαριασμών.
3. Τρίτον, έπειτα από αίτημα των χρηστών, η εφαρμογή θα συνδεθεί με λογαριασμούς από άλλες εφαρμογές ή και κοινωνικά δίκτυα. Αυτή η ανάγκη προκύπτει από το γεγονός ότι, όλες οι εφαρμογές σταδιακά προχωρούν στην υλοποίηση αυτής της επιλογής και οι χρήστες το βρίσκουν πιο εύκολο να διαχειρίζονται τις εφαρμογές τους από κάποιο κεντρικό λογαριασμό.
4. Τέταρτον, να δημιουργηθεί αντίστοιχη εφαρμογή κινητού(mobile application) η οποία να προσφέρει τις ίδιες δυνατότητες.
5. Η εφαρμογή θα πρέπει να υλοποιεί δυναμικά τα αρχεία Γοργία των χρηστών. Επίσης θα πρέπει να έχει τη δυνατότητα να επιβάλλει αλλαγές σε αυτά τα αρχεία ανάλογα με τις αλλαγές στις προτιμήσεις των χρηστών.

Βιβλιογραφία

- [1] Antonis C. Kakas, Pavlos Moraitis, Nikolaos I. Spanoudakis, “GORGIAS: Applying Argumentation”.
- [2] Theodoros Mitsikas, Nikolaos Spanoudakis, Petros Stefaneas, Antonis Kakas, “From Natural Language to Argumentation and Cognitive Systems”.
- [3] Stella Heras, Javier Palanca and Carlos Ivan Chesnevar, “Argumentation-Based Personal Assistants for Ambient Assisted Living”.
- [4] Richard H. Thaler, Cass R. Sunstein, “Nudge”.
- [5] Σημειώσεις Μαθήματος «ΕΠΛ361 – Τεχνολογία Λογισμικού», Καπιτσάκη Γ. Τμήμα Πληροφορικής Πανεπιστημίου Κύπρου, 2015-2016.
- [6] Σημειώσεις Μαθήματος «ΕΠΛ434 – Λογικός Προγραμματισμός και Τεχνητή Νοημοσύνη», Κάκας Α. Τμήμα Πληροφορικής Πανεπιστημίου Κύπρου, 2016-2017.
- [7] Business Insider, Here’s where Americans are spending their extra money, <https://www.businessinsider.com/where-consumers-spend-their-extra-money-2016-6>
- [8] PersonalFinance, How young people spend their money, <https://www.iol.co.za/personal-finance/my-money/how-young-people-spend-their-money-18478663>
- [9] HTML Tutorials, <https://www.w3schools.com/html/>
- [10] JavaScript Tutorials, <https://www.w3schools.com/js/>
- [11] SQL Tutorials, <https://www.w3schools.com/sql/>
- [12] CSS Tutorials, <https://www.w3schools.com/css/>
- [13] PHP Tutorials, <https://www.w3schools.com/php/>

Παράρτημα Α: Παράδειγμα Προτιμήσεων Χρήστη σε ΓΟΡΓΙΑ

```
:- dynamic entertainment/0, clothes/0, food/0, obligations/0, random/0, current_time/1,  
mood/1, company/1.
```

```
:- compile('../gorgias/lib/gorgias.pl').
```

```
:- compile('../gorgias/ext/lpwnf.pl').
```

```
% Arguments for Options
```

```
%clothing
```

```
rule(r1(X), formal_clothes(X), []).
```

```
rule(r2(X), neg(formal_clothes(X)), []).
```

```
rule(r3(X), casual_clothes(X), []).
```

```
rule(r4(X), neg(casual_clothes(X)), []).
```

```
%fun
```

```
rule(r5(X), cinema(X), []).
```

```
rule(r6(X), neg(cinema(X)), []).
```

```
rule(r7(X), athletic_activities(X), []).
```

```
rule(r8(X), neg(athletic_activities(X)), []).
```

```
rule(r9(X), park(X), []).
```

```
rule(r10(X), neg(park(X)), []).
```

```
rule(r11(X), gaming(X), []).
```

```
rule(r12(X), neg(gaming(X)), []).
```

```
%daily_expenses
```

```
rule(r13(X), refreshment(X), []).  
rule(r14(X), neg(refreshment(X)), []).
```

%bills

```
rule(r15(X), rent(X), []).  
rule(r16(X), neg(rent(X)), []).
```

```
rule(r17(X), bills(X), []).  
rule(r18(X), neg(bills(X)), []).
```

%food

```
rule(r19(X), eat(X), []).  
rule(r20(X), neg(eat(X)), []).
```

%priorities "spent money objective"

%priorities "clothes"

```
rule(npr1(X), prefer(r1(X), r2(X)), []):- clothes.  
rule(npr2(X), prefer(r3(X), r4(X)), []):- clothes.  
rule(npr3(X), prefer(r6(X), r5(X)), []):- clothes.  
rule(npr4(X), prefer(r8(X), r7(X)), []):- clothes.  
rule(npr5(X), prefer(r10(X), r9(X)), []):- clothes.  
rule(npr6(X), prefer(r12(X), r11(X)), []):- clothes.  
rule(npr7(X), prefer(r14(X), r13(X)), []):- clothes.  
rule(npr8(X), prefer(r16(X), r15(X)), []):- clothes.  
rule(npr9(X), prefer(r18(X), r17(X)), []):- clothes.  
rule(npr10(X), prefer(r20(X), r19(X)), []):- clothes.
```

%priorities "fun"

```
rule(npr11(X), prefer(r2(X), r1(X)), []):- entertainment.
```

```

rule(npr12(X), prefer(r4(X), r3(X)), []):- entertainment.
rule(npr13(X), prefer(r5(X), r6(X)), []):- entertainment.
rule(npr14(X), prefer(r7(X), r8(X)), []):- entertainment.
rule(npr15(X), prefer(r9(X), r10(X)), []):- entertainment.
rule(npr16(X), prefer(r11(X), r12(X)), []):- entertainment.
rule(npr17(X), prefer(r14(X), r13(X)), []):- entertainment.
rule(npr18(X), prefer(r16(X), r15(X)), []):- entertainment.
rule(npr19(X), prefer(r18(X), r17(X)), []):- entertainment.
rule(npr20(X), prefer(r20(X), r19(X)), []):- entertainment.

```

%priorities "obligations"

```

rule(npr21(X), prefer(r2(X), r1(X)), []):- obligations.
rule(npr22(X), prefer(r4(X), r3(X)), []):- obligations.
rule(npr23(X), prefer(r6(X), r5(X)), []):- obligations.
rule(npr24(X), prefer(r8(X), r7(X)), []):- obligations.
rule(npr25(X), prefer(r10(X), r9(X)), []):- obligations.
rule(npr26(X), prefer(r12(X), r11(X)), []):- obligations.
rule(npr27(X), prefer(r14(X), r13(X)), []):- obligations.
rule(npr28(X), prefer(r15(X), r16(X)), []):- obligations.
rule(npr29(X), prefer(r17(X), r18(X)), []):- obligations.
rule(npr30(X), prefer(r20(X), r19(X)), []):- obligations.

```

%priorities "random"

```

rule(npr31(X), prefer(r2(X), r1(X)), []):- random.
rule(npr32(X), prefer(r4(X), r3(X)), []):- random.
rule(npr33(X), prefer(r6(X), r5(X)), []):- random.
rule(npr34(X), prefer(r8(X), r7(X)), []):- random.
rule(npr35(X), prefer(r10(X), r9(X)), []):- random.
rule(npr36(X), prefer(r12(X), r11(X)), []):- random.
rule(npr37(X), prefer(r13(X), r14(X)), []):- random.
rule(npr38(X), prefer(r16(X), r15(X)), []):- random.
rule(npr39(X), prefer(r18(X), r17(X)), []):- random.
rule(npr40(X), prefer(r19(X), r20(X)), []):- random.

```

%priorities "mood tired"

```

rule(npr41(X), prefer(r2(X), r1(X)), []):- mood(tired).
rule(npr42(X), prefer(r4(X), r3(X)), []):- mood(tired).
rule(npr43(X), prefer(r6(X), r5(X)), []):- mood(tired).
rule(npr44(X), prefer(r8(X), r7(X)), []):- mood(tired).
rule(npr45(X), prefer(r10(X), r9(X)), []):- mood(tired).

```

```

%priorities "mood(hungry)"

```

```

rule(npr46(X), prefer(r2(X), r1(X)), []):- mood(hungry).
rule(npr47(X), prefer(r4(X), r3(X)), []):- mood(hungry).
rule(npr48(X), prefer(r6(X), r5(X)), []):- mood(hungry).
rule(npr49(X), prefer(r8(X), r7(X)), []):- mood(hungry).
rule(npr50(X), prefer(r10(X), r9(X)), []):- mood(hungry).
rule(npr51(X), prefer(r12(X), r11(X)), []):- mood(hungry).
rule(npr52(X), prefer(r14(X), r13(X)), []):- mood(hungry).
rule(npr53(X), prefer(r16(X), r15(X)), []):- mood(hungry).
rule(npr54(X), prefer(r18(X), r17(X)), []):- mood(hungry).
rule(npr55(X), prefer(r19(X), r20(X)), []):- mood(hungry).

```

```

%priorities "mood(thirsty)"

```

```

rule(npr56(X), prefer(r2(X), r1(X)), []):- mood(thirsty).
rule(npr57(X), prefer(r4(X), r3(X)), []):- mood(thirsty).
rule(npr58(X), prefer(r6(X), r5(X)), []):- mood(thirsty).
rule(npr59(X), prefer(r8(X), r7(X)), []):- mood(thirsty).
rule(npr60(X), prefer(r10(X), r9(X)), []):- mood(thirsty).
rule(npr61(X), prefer(r12(X), r11(X)), []):- mood(thirsty).
rule(npr62(X), prefer(r13(X), r14(X)), []):- mood(thirsty).
rule(npr63(X), prefer(r16(X), r15(X)), []):- mood(thirsty).
rule(npr64(X), prefer(r18(X), r17(X)), []):- mood(thirsty).
rule(npr65(X), prefer(r20(X), r19(X)), []):- mood(thirsty).

```

```

%special priorities "company(alone)"

```

```

rule(spr1(X), prefer(r6(X), r5(X)), []):- entertainment, company(alone).
rule(spr2(X), prefer(spr1(X), npr13(X)), []):- entertainment, company(alone).
rule(spr3(X), prefer(r8(X), r7(X)), []):- entertainment, company(alone).
rule(spr4(X), prefer(spr3(X), npr14(X)), []):- entertainment, company(alone).
rule(spr5(X), prefer(r10(X), r9(X)), []):- entertainment, company(alone).

```


rule(spr6(X), prefer(spr5(X), npr15(X)), []):- entertainment, company(alone).

%special priorities "company(friends)"

%special priorities "company(family)"

rule(spr7(X), prefer(r12(X), r11(X)), []):- entertainment, company(family).

rule(spr8(X), prefer(spr7(X), npr16(X)), []):- entertainment, company(family).

%special priorities "current_time(morning)"

rule(spr9(X), prefer(r7(X), r8(X)), []):- current_time(morning).

rule(spr10(X), prefer(r12(X), r11(X)), []):- current_time(morning).

rule(spr11(X), prefer(r13(X), r14(X)), []):- current_time(morning).

rule(spr12(X), prefer(spr10(X), npr16(X)), []):- current_time(morning), entertainment,
company(alone).

Παράρτημα Β: Κώδικας εισόδου στοιχείων χρηστών για δημιουργία Λογαριασμού

```
session_start();

$conn = mysqli_connect("localhost", "root", "", "thesis");

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

//user details
$Name=$_POST['name'];
$Surname=$_POST['surname'];
$UserName=$_POST['username'];
$Password=$_POST['password'];
$City=$_POST['city'];
$Country=$_POST['country'];
>Email=$_POST['email'];
$Budget=$_POST['budget'];

if($_POST['password']===$_POST['p']){
    $Password = $_POST['password'];
}

if (!preg_match("/^[a-zA-Z ]*$/",$Name)) {
    $nameErr = "Only letters and white space allowed";
}

if (!preg_match("/^[a-zA-Z ]*$/",$Surname)) {
    $nameErr = "Only letters and white space allowed";
}

if (!filter_var($Email, FILTER_VALIDATE_EMAIL)) {
    $emailError = "Invalid email format";
}
```

```
}
```

```
$sql = "SELECT * FROM users WHERE UserName ='$Username'";
```

```
$result = mysqli_query($conn, $sql);
```

```
$input_func = "INSERT INTO users (Name,Surname, Username, Password,  
City,Country,Email,Budget) VALUES ($Name','$Surname', '$Username',  
'$Password','$City','$Country','$Email','$Budget')";
```

```
.
```

