

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΤΟΜΙΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μάιος 2021

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΦΟΡΑΣ ΠΕΡΙΣΤΑΤΙΚΩΝ
ΣΤΟΥΣ ΔΗΜΟΥΣ**

Χρύσω Μακρή

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Συστήματος Αναφοράς Περιστατικών στους Δήμους

Χρύσω Μακρή

Επιβλέπων Καθηγητής

Γιώργος Παπαδόπουλος

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

Μάιος 2021

Ευχαριστίες

Θα ήθελα με την ολοκλήρωση της διπλωματικής εργασίας μου να ευχαριστήσω τα άτομα που με βοήθησαν στο να εκπληρώσω τον στόχο αυτό.

Καταρχήν, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Γιώργο Παπαδόπουλο, που μου έδωσε με την ευκαιρία να ασχοληθώ με την ανάπτυξη ενός τέτοιου συστήματος, κάτι που θα με βοηθήσει στην μετέπειτα επαγγελματική μου καριέρα.

Επίσης, θα ήθελα να πω ένα μεγάλο ευχαριστώ στον Δρ. Αχιλλέα Αχιλλέως (Συνεπιβλέπων), που βρισκόταν δίπλα μου, μου έδινε πολύτιμες συμβουλές και με καθοδηγούσε καθ' όλη τη διάρκεια της εργασίας μου.

Τέλος, το πιο μεγάλο ευχαριστώ το οφείλω στους φίλους και στην οικογένειά μου, και κυρίως στους γονείς μου, που με βοήθησαν και με στήριξαν κατά τη διάρκεια της εργασίας αυτής!

Περίληψη

Στην εποχή που ζούμε, η τεχνολογία έχει αναπτυχθεί αρκετά και τα πάντα πλέον, έχουν αυτοματοποιηθεί. Κάθε πρόβλημα που εντοπίζεται στην κοινωνία, επιλύεται με κύριο μέσο το διαδίκτυο και την τεχνολογία, έτσι ώστε να παρέχεται στον κόσμο μία γρήγορη, εύκολη και προσβάσιμη από παντού, λύση.

Το πρόβλημα που εντοπίστηκε, και κλήθηκα να επιλύσω, ήταν πως στους διάφορους δήμους της Κύπρου παρατηρούνται, συχνά, αρκετές προβληματικές καταστάσεις, που επηρεάζουν αρνητικά τη ζωή των πολιτών. Π.χ. πεζοδρόμια ή δρόμοι που χρειάζονται επισκευή, γκράφιτι σε τοίχους που πρέπει να αφαιρεθούν, σκουπίδια πεταμένα στους δρόμους και πολλά άλλα προβλήματα, που παραμένουν άλυτα. Οι υπάλληλοι των δήμων δεν μπορούν να γνωρίζουν όλα αυτά τα περιστατικά, ούτε την ακριβή τους τοποθεσία. Παράλληλα, οι πολίτες χρειάζονται έναν εύκολο τρόπο για να αναφέρουν στους αρμόδιους κάποιο πρόβλημα. Έτσι διαφάνηκε πως, η διαδικασία αναφοράς ενός προβλήματος, μέχρι την επίλυσή του, χρειάζεται να αυτοματοποιηθεί άμεσα.

Ο σκοπός της διπλωματικής εργασίας μου, ήταν να υλοποιήσω ένα σύστημα, το οποίο θα χρησιμοποιείται τόσο από τους πολίτες, όσο και από τους υπάλληλους των δήμων. Οι πολίτες θα χρησιμοποιούν το σύστημα κυρίως για να αναφέρουν προβληματικά περιστατικά, επιλέγοντας τον τύπο και την τοποθεσία του περιστατικού, αναρτώντας φωτογραφίες του προβλήματος και γράφοντας μία μικρή περιγραφή για αυτό. Οι υπάλληλοι των δήμων θα μπορούν να δουν τις διάφορες αναφορές των πολιτών του δήμου τους, για να μπορέσουν να εντοπίσουν το πρόβλημα και να το επιλύσουν.

Στο τέλος της ανάπτυξης του συστήματος, έγινε και μία μικρή έρευνα κατά πόσο αυτό το σύστημα είναι εύχρηστο και προσφέρει μία ευχάριστη εμπειρία στο χρήστη. Το σύστημα αξιολογήθηκε από μία μικρή ομάδα πολιτών, έτσι ώστε να ελεγχθεί κατά πόσο οι χρήστες είναι ικανοποιημένοι από αυτό και κατά πόσο είναι εύκολο να χρησιμοποιηθεί για το σκοπό που αναφέρθηκε και πριν.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Εισαγωγή	1
	1.2 Στόχος Διπλωματικής Εργασίας	2
	1.3 Δομή Διπλωματικής Εργασίας	3
Κεφάλαιο 2	Προαπαιτούμενη Γνώση και Παρόμοια Συστήματα	5
	2.1 Crowdsourcing	5
	2.2 Internet of Things	6
	2.3 Smart City	7
	2.4 Mobile Computing	9
	2.5 Παρόμοια Συστήματα	10
	2.5.1 Snap Send Solve – Αυστραλία & Νέα Ζηλανδία	10
	2.5.2 Improve My City – Δήμος Θέρμης (Ελλάδα)	11
Κεφάλαιο 3	Προδιαγραφές Συστήματος.....	14
	3.1 Απαιτήσεις Συστήματος	14
	3.2 Αρχιτεκτονική Συστήματος	16
	3.3 Απαιτούμενες Τεχνολογίες	17
	3.1.1 Laravel	17
	3.1.2 Composer	19
	3.1.3 PHP	19
	3.1.4 HTML	20
	3.1.5 CSS	21
	3.1.6 JavaScript	21
	3.1.7 MySQL	22
	3.2 Σχεδιασμός Βάσης Δεδομένων	23

Κεφάλαιο 4	Υλοποίηση Συστήματος.....	27
4.1	Guest – Επισκέπτης	27
4.1.1	Αρχική σελίδα	28
4.1.2	Contact Us	31
4.1.3	Sign Up	32
4.1.4	Login	33
4.2	Simple User – Πολίτης	37
4.2.1	Αρχική Σελίδα	37
4.2.2	Report A Problem	38
4.2.3	User’s Details	40
4.2.4	Report History	42
4.3	Operator – Υπάλληλος του Δήμου	43
4.3.1	Management	43
4.3.1.1	Reports	44
4.3.1.2	Users	45
4.4	Administrator - Διαχειριστής Συστήματος	47
Κεφάλαιο 5	Αξιολόγηση Συστήματος.....	49
5.1	User Experience Questionnaire	49
5.2	Αποτελέσματα Ερωτηματολογίου	52
Κεφάλαιο 6	Μελλοντική Εργασία και Συμπεράσματα	57
6.1	Μελλοντική Εργασία	57
6.2	Συμπεράσματα	58
Βιβλιογραφία		59
Παράρτημα Α.....		A-1
Παράρτημα Β.....		B-1
Παράρτημα Γ.....		Γ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή	1
1.2 Στόχος Διπλωματικής Εργασίας	2
1.3 Δομή Διπλωματικής Εργασίας	3

1.1 Εισαγωγή

Τα τελευταία χρόνια η ραγδαία ανάπτυξη της επιστήμης της πληροφορικής, έχει συμβάλει θετικά στην εξέλιξη της ανθρωπότητας. Ο κάθε άνθρωπος έχει εντάξει τη σύγχρονη τεχνολογία στην καθημερινή του ζωή, είτε για ψυχαγωγία, είτε για ενημέρωση, είτε για να λύσει κάποιο θέμα που τον απασχολεί. Γενικότερα, η ζωή μας έχει βελτιωθεί σε μεγάλο βαθμό και η κοινωνία έχει προοδεύσει αρκετά. Συνεπώς, είναι λογικό πολλές λειτουργίες να έχουν αυτοματοποιηθεί, έτσι ώστε να προσφέρουν στους ανθρώπους εύκολες και γρήγορες λύσεις στα προβλήματά τους.

Αρκετά συχνά, καλούμαστε να αντιμετωπίσουμε διάφορες προκλήσεις, και μέσα σε αυτές συμπεριλαμβάνονται και διάφορα προβληματικά περιστατικά που παρατηρούνται στους δήμους της Κύπρου. Σίγουρα, δεν είναι ευχάριστο να συναντούμε σκουπίδια στη γειτονιά μας, προβλήματα στο οδικό δίκτυο ή ανεπαρκή φωτισμό στους δρόμους. Είναι σημαντικό, τέτοιου είδους θέματα, να αναφέρονται άμεσα και να επιλύονται, λόγω του ότι εκτός από την άσχημη εικόνα που προσδίδουν, είναι μερικές φορές και επικίνδυνα για τους πολίτες.

Παρόλη τη σοβαρότητα όμως αυτών των περιστατικών, παρατηρείται πως παραμένουν άλυτα για μεγάλο χρονικό διάστημα πριν εντοπιστούν από τους αρμόδιους για να επιλυθούν. Το έργο των υπαλλήλων των δήμων είναι αρκετά δύσκολο, αφού δεν μπορούν να γνωρίζουν πότε ένα τέτοιο πρόβλημα εμφανίζεται και σε ποια τοποθεσία. Επιπρόσθετα, οι πολίτες όταν εντοπίσουν κάποιο θέμα που θέλουν να αναφέρουν χρειάζεται να το κάνουν είτε τηλεφωνικώς, είτε μέσω email, είτε

μέσω κάποιας επιστολής που θα στείλουν στον δήμο. Οι 3 αυτές λύσεις δεν είναι αρκετά γρήγορες, εύκολες και προσβάσιμες, με αποτέλεσμα τα προβλήματα που εντοπίζονται να μην αναφέρονται στους αρμόδιους. Ως επακόλουθο των παραπάνω, τα προβλήματα στους δήμους παραμένουν άλυτα και δυσχεραίνουν τις ζωές των πολιτών. Αυτό είναι εντελώς αντίθετο από την εικόνα της σύγχρονης εποχής, που θα έπρεπε να συμβαδίζει με την ανάπτυξη της πληροφορικής και της τεχνολογίας.

Επομένως, θεώρησα σημαντικό, αφού έχουμε τα μέσα και τη γνώση, να αναπτύξω ένα σύστημα που θα αυτοματοποιεί την διαδικασία αναφοράς προβληματικών περιστατικών στους δήμους. Θα ήθελα πολύ να συμβάλω στην ευημερία των συμπολιτών μου και στη βελτίωση της ποιότητας της ζωής τους.

1.2 Στόχος Διπλωματικής Εργασίας

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος το οποίο θα αυτοματοποιεί τη διαδικασία αναφοράς προβληματικών περιστατικών στους δήμους. Για το σκοπό αυτό, θα δημιουργηθεί μία ιστοσελίδα που θα αξιοποιείται και από τους πολίτες αλλά και από τους υπαλλήλους των δήμων.

Οι πολίτες με την δημιουργία της ιστοσελίδας θα ήθελα να έχουν τη δυνατότητα να αναφέρουν τα περιστατικά που συναντούν, σε πολύ σύντομο χρονικό διάστημα, εύκολα και από οπουδήποτε, φτάνει να έχουν πρόσβαση στο διαδίκτυο. Η αναφορά του περιστατικού θα περιλαμβάνει την κατηγορία του περιστατικού, μία μικρή περιγραφή, κάποιες φωτογραφίες αλλά και την τοποθεσία του περιστατικού.

Παράλληλα, οι υπάλληλοι των δήμων θα μπορούν να βλέπουν τα περιστατικά του δήμου τους και τις πληροφορίες για αυτά. Επίσης, θα μπορούν να κατατάσσουν το κάθε πρόβλημα σε ομάδες “Unseen”, “Seen”, “In Progress”, “Solved” και “Blocked”, για εύκολη διαχείριση των περιστατικών. Εκτός από αυτό, το σύστημα θα προσφέρει και ένα είδος βαθμολογίας των πολιτών βάση των αναφορών που έχουν κάνει, έτσι ώστε να ενθαρρύνει τους πολίτες να χρησιμοποιούν την εφαρμογή και να κάνουν στοχευμένες αναφορές με περιστατικά, που όντως είναι στην αρμοδιότητα των δήμων για να επιλυθούν.

Τέλος, στοχεύω στο να κάνω αυτή τη διαδικτυακή εφαρμογή εύχρηστη και κατανοητή. Είναι πολύ σημαντικό, οι χρήστες να καταλαβαίνουν αμέσως πως χρησιμοποιούν την εφαρμογή για να αναφέρουν ένα πρόβλημα, ώστε να μην εγκαταλείψουν την προσπάθεια και να αποφασίσουν να μην χρησιμοποιούν την εφαρμογή ή να χρειαστούν αρκετή ώρα μέχρι να διεκπεραιώσουν αυτή τη λειτουργία. Γι' αυτό, όπως αναφέρθηκε και πριν, στόχος μου είναι να αναπτύξω ένα σύστημα αποδοτικό, εύχρηστο, προσβάσιμο από παντού και εύκολο στην εκμάθηση.

1.3 Δομή Διπλωματικής Εργασίας

Η παρούσα Διπλωματική Εργασία αποτελείται από 6 κεφάλαια. Πιο κάτω δίνεται μία μικρή περιγραφή για το περιεχόμενο στο κάθε κεφάλαιο.

Κεφάλαιο 1 - Εισαγωγή:

Το πρώτο κεφάλαιο περιέχει μία γενική εισαγωγή στο θέμα της Διπλωματικής Εργασίας. Αναφέρεται το πρόβλημα που εντοπίστηκε και η προτεινόμενη λύση σε αυτό. Επίσης, αναφέρεται συνοπτικά ο στόχος της Διπλωματικής Εργασίας και στη δομή γίνεται μία μικρή περίληψη για το κάθε κεφάλαιο.

Κεφάλαιο 2 - Προαπαιτούμενη Γνώση και Παρόμοια Συστήματα:

Στο δεύτερο κεφάλαιο θα αναλυθεί η θεωρία και κάποιοι όροι σχετικοί με το σύστημα που θα δημιουργηθεί. Ακόμα, θα γίνει και μία μικρή περιγραφή κάποιων συστημάτων που ήδη υπάρχουν και μοιάζουν με αυτό που αναπτύχθηκε.

Κεφάλαιο 3 – Προδιαγραφές Συστήματος:

Στο τρίτο κεφάλαιο θα γίνει αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος, την αρχιτεκτονική του και το σχεδιασμό της βάσης δεδομένων. Παράλληλα, στο Παράρτημα Α, θα παρουσιαστεί ο κώδικα δημιουργίας των Migrations.

Κεφάλαιο 4 – Υλοποίηση Συστήματος:

Σε αυτό το κεφάλαιο θα παρουσιαστούν οι διάφορες οθόνες του συστήματος και θα αναλυθούν τα διάφορα βήματα που θα πρέπει να ακολουθήσει ο κάθε τύπος χρήστη για να κάνει τις βασικές

λειτουργίες, όπως εγγραφή ή αναφορά περιστατικού. Παράλληλα, κομμάτια του κώδικα και μερική εξήγηση του θα γίνουν στο Παράρτημα Β.

Κεφάλαιο 5 – Αξιολόγηση Συστήματος:

Στο πέμπτο κεφάλαιο, θα αναλυθεί το ερωτηματολόγιο UEQ που χρησιμοποιήθηκε για αξιολόγηση της ευχρηστίας του συστήματος. Επιπρόσθετα, θα παρουσιαστούν τα αποτελέσματα που εξάχθηκαν από αυτό. Εικόνες από το ερωτηματολόγιο θα παρουσιαστούν στο Παράρτημα Γ.

Κεφάλαιο 6 – Συμπεράσματα και Μελλοντική Εργασία:

Στο έκτο και τελευταίο κεφάλαιο, θα αναλυθούν τα τελικά συμπεράσματα και θα αναφερθούν μερικές ιδέες για το πως μπορεί αυτό το σύστημα να εξελιχτεί μελλοντικά.

Κεφάλαιο 2

Προαπαιτούμενη Γνώση και Παρόμοια Συστήματα

2.1 Crowdsourcing	5
2.2 Internet of Things	6
2.3 Smart City	7
2.4 Mobile Computing	9
2.5 Παρόμοια Συστήματα	10
2.5.1 Snap Send Solve - Αυστραλία & Νέα Ζηλανδία	10
2.5.2 Improve My City - Δήμος Θέρμης (Ελλάδα)	11

2.1 Crowdsourcing

Το crowdsourcing είναι μία ομαδική διαδικτυακή δραστηριότητα, στην οποία εμπλέκεται το πλήθος (crowd), δηλαδή μία μεγάλη ομάδα χρηστών[1]. Στο πλήθος ανατίθενται κάποιες δραστηριότητες, οι οποίες έχουν ένα κοινό σκοπό, είτε την καινοτομία, είτε την αποτελεσματικότητα, είτε την επίλυση ενός προβλήματος. Κατά τη διαδικασία αυτή, αξιοποιείται η γνώση, η εμπειρία, ο χρόνος ή οι πόροι των χρηστών.

Αυτή η ιδέα, να αξιοποιείς το πλήθος ώστε να φτάσεις σε ένα αποτέλεσμα, δεν είναι καινούργια.[2] Το πρώτο παράδειγμα στην ιστορία το συναντάμε στην αρχαία Αθήνα, το 594 π.Χ. Ο Σόλωνας απαίτησε από όλους τους πολίτες να ορκιστούν τήρηση των νόμων, και κάπως έτσι οι πολίτες συμμετείχαν στη σωστή διακυβέρνηση της αρχαίας Αθήνας, δημιουργώντας έτσι την πρώτη δημοκρατική κυβέρνηση. Ακόμα ένα παράδειγμα αποτελεί η συγγραφή του αγγλικού λεξικού της Οξφόρδης, το 1884, όπου 800 εθελοντές βοήθησαν στην ολοκλήρωση αυτού του έργου.

Τώρα, με την βοήθεια του διαδικτύου, όλοι είμαστε συνδεδεμένοι και η ιδέα του crowdsourcing μπορεί να πραγματοποιηθεί πολύ πιο εύκολα. Η δημοφιλής χρήση του διαδικτύου, καθιστά την

επικοινωνία και το συντονισμό των ατόμων και των εργασιών μία πολύ γρήγορη διαδικασία. Επίσης, χάρη στην αυξανόμενη συνδεσιμότητα μας, είναι πιο εύκολο από ποτέ τα άτομα να συνεισφέρουν συλλογικά για ένα σκοπό. Επομένως, μπορούμε να αξιοποιήσουμε αυτήν την ιδέα σε πολλούς τομείς.

Οποιοσδήποτε μπορεί να χρησιμοποιήσει το crowdsourcing. Οι εταιρείες μπορούν να αξιοποιήσουν τις δεξιότητες των εργαζομένων τους και να τελειώσουν μία εργασία πιο γρήγορα.[3] Επίσης, οργανισμοί μπορούν να διακρίνουν τις επιθυμίες των πελατών τους και να ζητήσουν τη γνώμη τους, ώστε να ελέγξουν αν τους αρέσουν αυτά που τους προσφέρουν, για να βελτιωθούν. Ακόμη, οι κυβερνήσεις δίνουν φωνή στους πολίτες εφαρμόζοντας το crowdsourcing. Τέλος, στον τομέα της εκπαίδευσης και της επιστήμης γίνονται πολλές έρευνες και αναπτυσσόμαστε με ραγδαίο ρυθμό.

Σε αυτή τη διπλωματική εργασία θα αξιοποιηθεί η ιδέα του crowdsourcing. Το σύστημα αναφοράς περιστατικών θα βασίζεται στο κοινό, το οποίο θα αναφέρει τα προβληματικά περιστατικά στους δήμους της Κύπρου. Επίσης, οι υπάλληλοι του κάθε δήμου θα είναι υπεύθυνοι να επιλύουν τα προβλήματα που εμφανίζονται στο δικό τους δήμο. Κάπως έτσι, θα εντοπίζουμε τα περιστατικά, αξιοποιώντας τους πολίτες, που εθελοντικά θα αναφέρουν τα προβλήματα που εντοπίζουν και η επίλυση των περιστατικών θα διαμοιράζεται στους διάφορους δήμους, ώστε να γίνεται πιο γρήγορη η επίλυσή τους.

2.2 Internet of Things

Με την ορολογία Internet of Things (IoT) αναφερόμαστε σε μία πληθώρα συσκευών, που είναι ενσωματωμένες με αισθητήρες, λογισμικό και άλλες τεχνολογίες, και είναι συνδεδεμένες μεταξύ τους μέσω διαδικτύου.[4][5][6] Το IoT έχει ως στόχο την σύνδεση και την ανταλλαγή δεδομένων μεταξύ των διαφόρων συσκευών και συστημάτων. Η συλλογή δεδομένων γίνεται με ένα συγκεκριμένο σκοπό, ο οποίος ποικίλει ανάλογα με το είδος της συσκευής και τι ανάγκες μας καλύπτει.

Γενικότερα, ο όρος “Things”, περιλαμβάνει από τις πιο απλές μέχρι και τις πιο περίπλοκες συσκευές, φτάνει αυτές να έχουν την δυνατότητα σύνδεσης και ανταλλαγής δεδομένων με τη

χρήση διαδικτύου. Μερικές από αυτές είναι τα κινητά τηλέφωνα, ρολόγια, ακουστικά, οχήματα, οικιακές συσκευές, λαμπτήρες και οτιδήποτε άλλο μπορούμε να φανταστούμε.

Αυτές οι συσκευές είναι συνδεδεμένες με μια πλατφόρμα, η οποία συλλέγει τα δεδομένα τους. Οι συσκευές IoT μπορούν εύκολα να εντοπίσουν ποια δεδομένα είναι χρήσιμα ή όχι και να τα αποθηκεύσουν για μελλοντική χρήση. Με αυτόν τον τρόπο, μας δίνεται η δυνατότητα να αυτοματοποιήσουμε διάφορες λειτουργίες. Δημιουργούνται αρκετές εφαρμογές οι οποίες συλλέγουν τις πληροφορίες από τις IoT συσκευές, τις αποθηκεύουν και στη συνέχεια τις αξιοποιούν για να επιλύσουν δύσκολα, χρονοβόρα ή και επικίνδυνα μερικές φορές προβλήματα. Οι εφαρμογές αυτές χρησιμοποιούνται από τους ανθρώπους, για να χειρίζονται και να παρακολουθούν τις “έξυπνες” συσκευές, αλλά και να επεξεργάζονται και να μεταφράζουν τα δεδομένα που συλλέγονται.

Πλέον το IoT έχει αποκτήσει αρκετή ζήτηση στην ψηφιακή εποχή που ζούμε και δε θα μπορούσε να λείπει από την διπλωματική εργασία μου. Στη δικιά μου περίπτωση, το σύστημα αναφοράς περιστατικών αποτελεί μία διαδικτυακή εφαρμογή, η οποία θα συλλέγει πληροφορίες από τα κινητά τηλέφωνα, τα tablets, και τους ηλεκτρονικούς υπολογιστές των χρηστών, συσκευές άμεσα συνδεδεμένες με το διαδίκτυο. Οι πληροφορίες αυτές θα αποθηκεύονται σε μία βάση δεδομένων και ακολούθως οι υπάλληλοι των δήμων θα τις βλέπουν και θα τις επεξεργάζονται ανάλογα.

2.3 Smart City

Μία έξυπνη πόλη, χρησιμοποιεί διάφορες ηλεκτρονικές μεθόδους και την σύγχρονη τεχνολογία για να παρέχει διάφορες υπηρεσίες στους πολίτες της και να επιλύει τα προβλήματα που εμφανίζονται. [7][8] Κύριος σκοπός μίας έξυπνης πόλης είναι η βελτιστοποίηση των λειτουργιών της, ώστε να διευκολύνει τη ζωή των πολιτών και των επισκεπτών, προσφέροντας τους ευημερία.

Για την καταπολέμηση των αστικών προβλημάτων, ακολουθείται μία συγκεκριμένη διαδικασία από τις έξυπνες πόλεις. Αρχικά, πρέπει να γίνει συλλογή των δεδομένων. Ακολούθως, τα δεδομένα πρέπει να αναλυθούν και αφού γίνει η ανάλυση, τα αποτελέσματά της θα κοινοποιηθούν στους υπεύθυνους λήψης αποφάσεων. Στο τέλος, αυτοί θα αναλάβουν δράση για να επιλύσουν τα προβλήματα που εμφανίστηκαν και να βελτιώσουν την ποιότητα ζωής των πολιτών.

Ξεκινώντας από την συλλογή δεδομένων, μία έξυπνη πόλη μπορεί να χρησιμοποιεί διάφορους τρόπους για να λαμβάνει πληροφορίες. Τα δεδομένα συλλέγονται από τις συσκευές των πολιτών αλλά και από διάφορους αισθητήρες, που είναι εγκατεστημένοι στην πόλη και στα κτήρια της. Χάρη στις εξαιρετικά καινοτόμες λύσεις που προσφέρει το Internet of Things μπορούμε να ενσωματώσουμε διάφορες συσκευές συνδεδεμένες στο διαδίκτυο για να έχουμε περισσότερη αποτελεσματικότητα στη συλλογή δεδομένων.

Αφού γίνει η συλλογή των δεδομένων, αυτά αποθηκεύονται μέχρι να έρθει η στιγμή να αξιοποιηθούν. Τα δεδομένα αναλύονται και επεξεργάζονται από τους αρμόδιους και στη συνέχεια γίνονται οι απαραίτητες ενέργειες για επίλυση των προβλημάτων και παροχή υπηρεσιών στους πολίτες. Η τεχνολογία αυτή επιτρέπει στους υπεύθυνους να αλληλοεπιδρούν άμεσα με την υποδομή της κοινότητας και της πόλης και να παρακολουθούν τι συμβαίνει.

Με τα δεδομένα που συλλέγονται, οι πόλεις προσπαθούν να διαχειριστούν τους πόρους, τις υπηρεσίες και τα περιουσιακά στοιχεία τους με τον καλύτερο δυνατό τρόπο. Γίνεται προσπάθεια για βελτίωση της οικονομικής ποιότητας και μείωση της σπατάλης, αλλά και για να αποκτηθεί μια πιο βιώσιμη υποδομή και να συντηρηθούν ορισμένα σημεία της πόλης που χρειάζονται επισκευή. Ακόμη, οι κοινότητες μπορούν να καταπολεμήσουν την ατμοσφαιρική ρύπανση και την κλιματική αλλαγή, κάνοντας καλύτερη κατανομή ενέργειας, βελτιώνοντας τη συλλογή απορριμμάτων και μειώνοντας την κυκλοφοριακή συμφόρηση στους δρόμους. Τέλος, οι έξυπνες πόλεις μπορούν να παρέχουν στους πολίτες το αίσθημα της ασφάλειας, αφού έχουν την δυνατότητα να παρακολουθούν επικίνδυνες περιοχές όπου συμβαίνουν κατολισθήσεις, τυφώνες, εγκληματικά περιστατικά κ.ά. και να προειδοποιούν όταν χρειάζεται. Γενικότερα, μία έξυπνη πόλη είναι πιο έτοιμη στο να ανταποκριθεί στις δύσκολες προκλήσεις της καθημερινότητας, παρά μία απλή πόλη και έτσι αυτό-βελτιώνεται και αναπτύσσεται.

Οι πολίτες επωφελούνται από τις υπηρεσίες και τις λειτουργίες που τους παρέχονται. [9] Όμως παίζουν ένα πολύ σημαντικό ρόλο και οι ίδιοι στην πόλη τους, αφού μία έξυπνη πόλη δίνει την δυνατότητα και στους πολίτες να συμμετέχουν. Για παράδειγμα, οι πολίτες μπορούν να έχουν πρόσβαση στα διάφορα συστήματα, για να ειδοποιούν τους αρμόδιους για τυχόν προβλήματα που εντοπίζουν και να παρέχουν τις απαραίτητες πληροφορίες. Με αυτές τις πληροφορίες η κυβέρνηση και οι υπάλληλοι μπορούν να δράσουν αμέσως για να λύσουν σχεδόν κάθε πρόβλημα.

Το Σύστημα Αναφοράς Περιστατικών, πάνω στο οποίο αναφέρεται η διπλωματική μου θα χρησιμοποιεί αυτή την λογική του Smart City. Δηλαδή, θα συλλέγει δεδομένα, με τη βοήθεια των IoT συσκευών και των πολιτών και αυτά τα δεδομένα θα χρησιμοποιούνται για την ανάπτυξη της πόλης, για την προώθηση πρακτικών αειφόρου ανάπτυξης και για την αντιμετώπιση των αυξανόμενων αστικών προκλήσεων. Οι πολίτες θα έχουν πρόσβαση στο σύστημα για να ειδοποιούν τους υπαλλήλους, για περιστατικά όπως λακκούβες, πεταμένα σκουπίδια, γκράφιτι κ.ά. και οι υπάλληλοι θα λαμβάνουν δράση για να τα επιλύσουν. Με αυτό τον τρόπο θα γίνεται βελτίωση της ποιότητας ζωής των πολιτών στα πλαίσια της βιώσιμης ανάπτυξης και οι επισκέπτες θα σχηματίζουν μία πολύ καλή εντύπωση για την πόλη.

2.4 Mobile Computing

Το mobile computing είναι η ορολογία που αναφέρεται σε μία πληθώρα συσκευών, οι οποίες επιτρέπουν στους χρήστες να έχουν πρόσβαση σε πληροφορίες και δεδομένα από οπουδήποτε και αν βρίσκονται. [10] Είναι ένα πεδίο της πληροφορικής, το οποίο στηρίζεται στην ασύρματη επικοινωνία και στις φορητές συσκευές (mobile devices), για να παρέχει πληροφόρηση στους τελικούς χρήστες ανεξαρτήτου της γεωγραφικής θέσης τους.

Αυτό το πεδίο της πληροφορικής υποστηρίζεται κυρίως από τα κινητά τηλέφωνα, τους φορητούς υπολογιστές και άλλες φορητές συσκευές που επιτρέπουν στους χρήστες να μεταδίδουν και να λαμβάνουν πληροφορίες από οπουδήποτε, ανά πάσα στιγμή. Οι φορητές συσκευές ποικίλουν σε μέγεθος, αφού μιλάμε για κινητά τηλέφωνα, tablets και laptops. Μπορούν να προσφέρουν τη δυνατότητα για τηλεφωνικές κλήσεις, για σύνδεση στο διαδίκτυο και πρόσβαση σε διάφορες άλλες εφαρμογές που επιλέγουν οι χρήστες. Η πρόσβαση στο διαδίκτυο είναι απαραίτητη σήμερα στις φορητές συσκευές καθώς και η ενσωματωμένη ισχύ μπαταρίας.

Για την πρόσβαση των τελικών χρηστών σε πληροφορίες, δεδομένα, πόρους και διάφορες άλλες λειτουργίες καθώς αυτοί μετακινούνται, χρησιμοποιείται ένα σύνολο τεχνολογιών, υπηρεσιών και προϊόντων πληροφορικής. [11] Καταρχήν, χρησιμοποιείται το mobile communication που αναφέρεται στην υποδομή που διασφαλίζει την αξιόπιστη επικοινωνία των συσκευών. Πολύ σημαντικό είναι και το mobile hardware και software. Το mobile computing βασίζεται στο Wi-Fi

και στην κινητή τεχνολογία. Παρέχεται ακόμα η δυνατότητα για αποθήκευση δεδομένων στην συσκευή ή στο cloud.

Το mobile computing μπορεί να προσφέρει πολλά στην σημερινή κοινωνία. Από απλή ψυχαγωγία μέχρι και μεταφορά πληροφοριών. Από ένα απλό κινητό τηλέφωνο μπορούμε να λάβουμε οδηγίες, για να οδηγηθούμε σε ένα σημείο. Ακόμη, μπορούμε να παραμείνουμε συνδεδεμένοι με όλες τις πηγές ανά πάσα στιγμή και να αλληλοεπιδρούμε με διάφορους άλλους χρήστες. Επιπρόσθετα, οι φορητές συσκευές χρησιμοποιούνται και για την εργασία των χρηστών αλλά και για τις προσωπικές τους ανάγκες. Οι εφαρμογές του mobile computing σήμερα έχουν γίνει αρκετά διαδεδομένες σε επιχειρήσεις, βιομηχανίες και στους καταναλωτές και το βασικό πλεονέκτημα που τις έχει κάνει τόσο δημοφιλείς είναι η ευκολία στη μετακίνησή τους, που επιτρέπει στους χρήστες να έχουν πρόσβαση στις πληροφορίες που θέλουν ανά πάσα στιγμή.

Σίγουρα, το mobile computing δεν θα μπορούσε να μην αποτελεί βασικό στοιχείο του συστήματος αναφοράς περιστατικών. Το σύστημα αυτό βασίζεται στις φορητές συσκευές, αφού οι χρήστες του, δηλαδή οι πολίτες των δήμων της Κύπρου, μόλις συναντήσουν ένα προβληματικό περιστατικό θα πρέπει να τους δίνεται η δυνατότητα να το αναφέρουν από οπουδήποτε και αν βρίσκονται, ανά πάσα στιγμή. Αυτό θα επιτυγχάνεται με τα mobile devices και οι χρήστες θα μπορούν να μεταδίδουν τις πληροφορίες για το περιστατικό (φωτογραφίες, τύπος περιστατικού, περιγραφή και τοποθεσία) οποιαδήποτε στιγμή ανεξαρτήτου γεωγραφικής θέσης.

2.5 Παρόμοια Συστήματα

Για την ανάπτυξη του συστήματος αναφοράς περιστατικών έχω πάρει αρκετές ιδέες από προϋπάρχοντα συστήματα. Αυτά τα συστήματα αποτελούνται από παρόμοια στοιχεία και έχουν σαν σκοπό να βελτιώσουν την πόλη για την οποία δημιουργήθηκαν. Συμβάλλουν όλα στην ανάπτυξη των πόλεων και επιδιώκουν να δημιουργήσουν “Smart” – “Εξυπνες” πόλεις.

2.5.1 Snap Send Solve – Αυστραλία & Νέα Ζηλανδία [12]

Το κυριότερο σύστημα το οποίο έχω μελετήσει και έχω στηριχτεί για να πάρω ιδέες και να αναπτύξω το δικό μου σύστημα είναι το “Snap Send Solve”. Το συγκεκριμένο σύστημα

απευθύνεται τους πολίτες της Αυστραλίας και της Νέας Ζηλανδίας και στοχεύει στη βελτίωση της ποιότητας ζωής των ανθρώπων που ζουν εκεί.

Οι χρήστες μπορούν να χρησιμοποιήσουν την ιστοσελίδα του “Snap Send Solve” ή την εφαρμογή σε Android ή iOS. Το σύστημα αυτό προσφέρει στους χρήστες ένα πολύ εύκολο τρόπο για να αναφέρουν περιστατικά που εντοπίζουν στην κοινότητά του όπως γεμάτους κάδους, παράνομο παρκάρισμα, γκράφιτι κ.α. Οι χρήστες αφού εντοπίσουν ένα πρόβλημα, μπορούν να ενημερώσουν την αρμόδια αρχή με την χρήση αυτής της εφαρμογής με κάποια πολύ απλά βήματα. Αρχικά θα αναρτήσουν μερικές φωτογραφίες του περιστατικού, στη συνέχεια θα επιλέξουν τη γεωγραφική θέση του περιστατικού στο χάρτη και τέλος θα επιλέξουν τον τύπο του περιστατικού. Αυτά τα βήματα έχω ακολουθήσει και εγώ με διαφορετική σειρά στο σύστημά μου.

Το συγκεκριμένο σύστημα στέλνει την κάθε αναφορά περιστατικού στην αρμόδια αρχή, που αναλαμβάνει να λύσει μία κατηγορία προβλημάτων. Οι πολίτες προτιμούν να έχουν ένα σύστημα στο οποίο θα αναφέρουν όλες τις κατηγορίες προβλημάτων και όχι ένα σύστημα για κάθε κατηγορία. Οπότε, το σύστημα αναλαμβάνει να διανέμει τις αναφορές στην αρμόδια αρχή κάθε φορά. Για παράδειγμα, αν το πρόβλημα αφορά ανεπαρκή φωτισμό σε κάποια περιοχή, το πρόβλημα θα μεταφερθεί στην Αρχή Ηλεκτρισμού της χώρας. Έτσι, τα προβλήματα μοιράζονται και λύνονται πιο γρήγορα. Η συγκεκριμένη ιδέα ακολουθήθηκε στο σύστημά μου για τους διάφορους δήμους. Σίγουρα, αν τα προβλήματα διανέμονται στον κάθε δήμο θα μοιράζεται ο φόρτος εργασίας και θα έχουμε μεγαλύτερη αποτελεσματικότητα και γρήγορη επίλυσή τους.

2.5.2 Improve My City – Δήμος Θέρμης (Ελλάδα) [13]

Το δεύτερο σύστημα απ’ το οποίο έχω εμπνευστεί για να δημιουργήσω το σύστημα αναφοράς περιστατικών είναι το αντίστοιχο σύστημα που έχει αναπτύξει ο Δήμος Θέρμης στην Ελλάδα για τους δημότες του. Και αυτό το σύστημα έχει ως στόχο να προσφέρει ευημερία στους πολίτες και να αναπτύξει τον Δήμο Θέρμης.

Όπως και ο “Snap Send Solve”, το σύστημα αναφοράς περιστατικών του Δήμου Θέρμης διατίθεται στους πολίτες σαν ιστοσελίδα και σαν εφαρμογή σε Android και iOS. Η εφαρμογή επιτρέπει στους πολίτες αλλά και στους επισκέπτες του Δήμου να αναφέρουν περιστατικά, όπως:

σπασμένα πλακάκια στους δρόμους, καμένες λάμπες ηλεκτροδότησης, πεταμένα σκουπίδια κ.ά., αλλά και να κάνουν προτάσεις για να βελτιωθεί η πόλη. Ακόμη, μπορούν να σχολιάζουν και να δίνουν θετικές ψήφους σε καταχωρήσεις άλλων χρηστών.

Η διαδικασία που ακολουθούν οι χρήστες σε αυτήν την εφαρμογή απαιτεί από τον χρήστη να δημιουργήσει λογαριασμό στη σελίδα και να συνδεθεί πριν αναφέρει κάποιο πρόβλημα. Έχω ακολουθήσει και εγώ αυτήν τη διαδικασία στο σύστημά μου, αφού θεώρησα σημαντικό οι χρήστες να έχουν ένα λογαριασμό στο σύστημα, ώστε να επικοινωνούν μαζί τους οι αρμόδιοι σε περίπτωση που δεν εντοπίζουν το πρόβλημα που έχουν αναφέρει. Επίσης, ένας χρήστης σίγουρα θα θέλει να έχει πρόσβαση στο ιστορικό των αναφορών που έκανε και να βλέπει κάποια επιπλέον στοιχεία για το λογαριασμό του αλλά και για τις αναφορές του. Οπότε, ήταν απαραίτητη η δημιουργία λογαριασμού των χρηστών, όπως ακριβώς και στην εφαρμογή “Improve My City” του Δήμου Θέρμης.

Συνεχίζοντας με την διαδικασία που ακολουθεί η εφαρμογή “Improve My City”, αφού συνδεθεί ο χρήστης στο λογαριασμό τότε θα μπορεί να αναφέρει ένα περιστατικό συμπληρώνοντας ένα μικρό τίτλο και μία περιγραφή, επιλέγοντας κατηγορία περιστατικού και τοποθεσία και αναρτώντας μία φωτογραφία. Παρόμοια διαδικασία ακολουθείται και από το Snap Send Solve αλλά και από το σύστημα που έχω αναπτύξει.

Έπειτα, το περιστατικό περνά από διάφορες φάσεις μέχρι να επιλυθεί. Οι φάσεις αυτές παρουσιάζονται στο χρήστη που έχει υποβάλει το περιστατικό, έτσι ώστε να γνωρίζει την κατάσταση της αναφοράς του. Αρχικά βρίσκεται στην κατάσταση “Υποβλήθηκε”, μετά μόλις προωθηθεί στην αρμόδια υπηρεσία αλλάζει στην κατάσταση “Σε εξέλιξη” και τέλος, μόλις το πρόβλημα αντιμετωπιστεί αλλάζει σε “Διεκπεραιώθηκε”. Την ιδέα αυτή την πρόσθεσα και στο δικό μου σύστημα, γιατί είναι σημαντικό να παρέχεται ανατροφοδότηση στους χρήστες. Παράλληλα, οι υπάλληλοι των δήμων θα έχουν μία καλύτερη εικόνα και θα ομαδοποιούν πιο σωστά τα προβλήματα για να μπορούν να τα διεκπεραιώσουν γρηγορότερα. Οι διάφορες καταστάσεις που έχωβάλει στο σύστημα μου είναι “Unseen”, “Seen”, “In Progress” και “Solved” και στοχεύουν στο να δίνουν feedback στο χρήστη αλλά και στο να λύνονται αποδοτικά τα προβλήματα.

Το τελευταίο στοιχείο στο οποίο θα γίνει αναφορά για την συγκεκριμένη εφαρμογή, το οποίο έχω υιοθετήσει και εγώ στο δικό μου σύστημα, είναι ένας χάρτης με μερικά από τα περιστατικά που έχουν αναφέρει οι δημότες. Στο χάρτη αυτό φαίνεται ο δήμος Θέρμης και τα περιστατικά που ανέφεραν οι πολίτες του. Επιπλέον, παρουσιάζονται και οι βασικές πληροφορίες για κάθε περιστατικό. Με αυτόν τον τρόπο οι πολίτες βλέπουν και τα υπόλοιπα περιστατικά που αναφέρουν οι συμπολίτες τους και ωθούνται στο να αναφέρουν και αυτοί οτιδήποτε εντοπίζουν για να κάνουν τον δήμο τους ένα καλύτερο και πιο βιώσιμο μέρος. Στο σύστημά μου έχω προσθέσει ένα χάρτη τον οποίο οι χρήστες βλέπουν μόνο αν έχουν συνδεθεί στο λογαριασμό τους. Σε αυτόν το χάρτη φαίνονται τα τελευταία 20 “Solved” περιστατικά του δήμου στον οποίο ανήκει ο χρήστης, καθώς και ο τύπος αλλά και η μικρή περιγραφή του περιστατικού. Έτσι, ο χρήστης επηρεάζεται και ωθείται να βελτιώσει τον δήμο του αναφέροντας και αυτός κάποιο πρόβλημα το οποίο έχει εντοπίσει.

Κεφάλαιο 3

Προδιαγραφές Συστήματος

3.1 Απαιτήσεις Συστήματος	14
3.2 Αρχιτεκτονική Συστήματος	16
3.3 Απαιτούμενες Τεχνολογίες	17
3.1.1 Laravel	17
3.1.2 Composer	19
3.1.3 PHP	19
3.1.4 HTML	20
3.1.5 CSS	21
3.1.6 JavaScript	21
3.1.7 MySQL	22
3.2 Σχεδιασμός Βάσης Δεδομένων	23

3.1 Απαιτήσεις Συστήματος

Στο συγκεκριμένο υποκεφάλαιο θα παρουσιαστούν συνοπτικά οι απαιτήσεις του συστήματος.

- 1) Contact Us: Δίνεται η δυνατότητα στους χρήστες που επισκέπτονται το σύστημα να επικοινωνήσουν με τους υπεύθυνους αποστέλλοντας email συμπληρώνοντας τη φόρμα με ένα όνομα, email, θέμα και το μήνυμα που θέλουν να αποστείλουν.
- 2) Εγγραφή Χρήστη: Ο χρήστης μπορεί να δημιουργήσει λογαριασμό συμπληρώνοντας τα απαραίτητα στοιχεία του (ονοματεπώνυμο, τηλέφωνο, email, πόλη, δήμο, username και password)
- 3) Επιβεβαίωση Ηλεκτρονικής Διεύθυνσης: Μετά την εγγραφή του χρήστη αποστέλλεται ένα verification email στον χρήστη με ένα τυχαίο token και την σελίδα όπου πρέπει να

συμπληρώσει το username, το password και το token για να ολοκληρώσει την διαδικασία του signup.

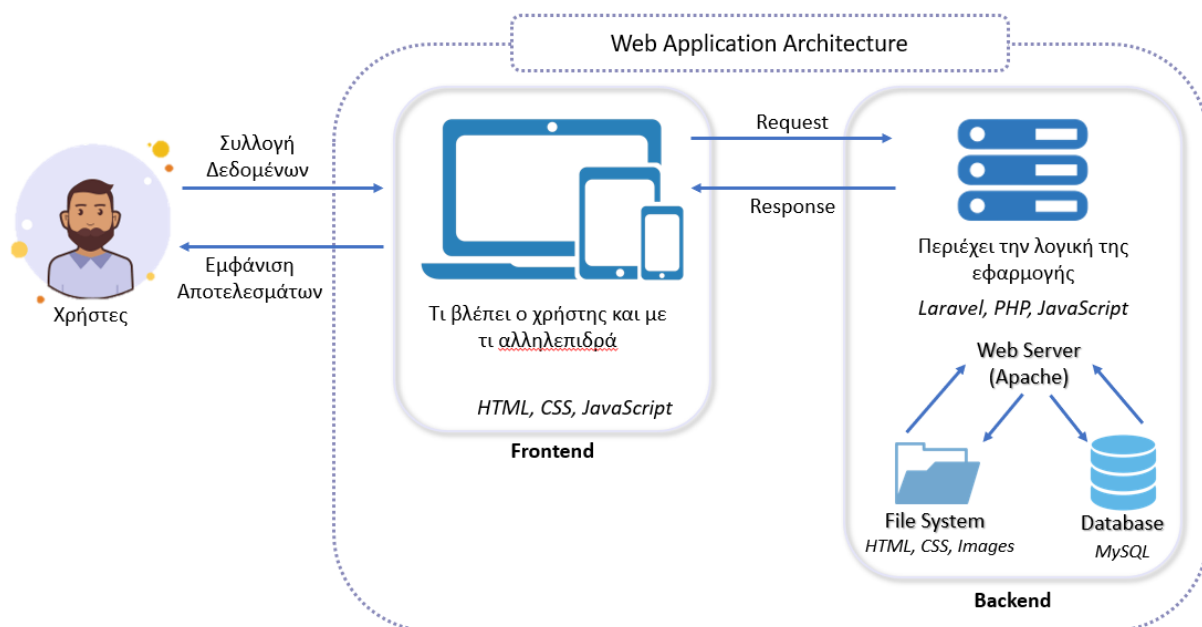
- 4) Login: Ο χρήστης μπορεί να συνδεθεί στο λογαριασμό του συμπληρώνοντας username και password.
- 5) Forgot Password: Ο χρήστης συμπληρώνει το username του και του αποστέλλεται ένα μοναδικό link στο email του όπου μπορεί να συμπληρώσει το username του και το νέο του password. Η διαδικασία πρέπει να γίνει εντός μίας ώρας.
- 6) Forgot Username: Ο χρήστης συμπληρώνει το email του και του αποστέλλεται ένα μοναδικό token και ένα link όπου θα συμπληρώσει το email του, το μοναδικό token και το νέο του username. Η διαδικασία πρέπει να γίνει εντός μίας ώρας.
- 7) Αναφορά Περιστατικού: Το σύστημα δίνει την δυνατότητα στο χρήστη αφού κάνει login, εάν συναντήσει ένα προβληματικό περιστατικό σε οποιοδήποτε δήμο, να το αναφέρει, συμπληρώνοντας την απαραίτητη φόρμα με τον τύπο του περιστατικού, μία μικρή περιγραφή, μέχρι 3 φωτογραφίες του περιστατικού και την τοποθεσία.
- 8) Προβολή Πληροφοριών του Χρήστη: Ο χρήστης μπορεί να δει τις πληροφορίες του λογαριασμού του.
- 9) Αλλαγή Πληροφοριών Χρήστη: Ο χρήστης μπορεί να αλλάξει ορισμένες από τις πληροφορίες του (email, τηλέφωνο, πόλη και δήμο).
- 10) Διαγραφή Χρήστη: Ο κάθε χρήστης μπορεί να διαγράψει τον δικό του λογαριασμό.
- 11) Report History: Ο κάθε χρήστης μπορεί να δει όλες τις αναφορές περιστατικών που έχει κάνει.
- 12) Διαχείριση Χρηστών: Οι υπάλληλοι των δήμων μπορούν να δουν τους χρήστες του δήμου τους και να αλλάξουν τα στοιχεία τους (ονοματεπώνυμο, email, τηλέφωνο, πόλη, δήμο, username, password, τύπο χρήστη και αριθμό φορών που έχουν γίνει blocked αναφορές του χρήστη). Να σημειωθεί ότι δεν μπορούν να αλλάξουν πληροφορίες άλλων υπαλλήλων του δήμου ή διαχειριστών.
Οι διαχειριστές μπορούν να δουν όλους τους χρήστες και να αλλάξουν τα στοιχεία όλων των χρηστών ανεξαρτήτως τύπου.
- 13) Διαχείριση Αναφορών: Οι υπάλληλοι των δήμων μπορούν να δουν τις αναφορές των περιστατικών που αφορούν τον δικό τους δήμο και να αλλάξουν την κατάσταση σε μία

από τις “Unseen”, “Seen”, “In Progress”, “Solved” και “Blocked”. Οι διαχειριστές μπορούν να διαχειριστούν όλες τις αναφορές.

3.2 Αρχιτεκτονική Συστήματος

Η Αρχιτεκτονική του Συστήματος αποτελείται από το Frontend και το Backend όπως συμβαίνει σε κάθε web application. Με την βοήθεια του Frontend (ή αλλιώς client-site), ο χρήστης μπορεί να βλέπει την διαδικτυακή εφαρμογή και να αλληλεπιδρά μαζί της. Για το Frontend έχω χρησιμοποιήσει τις γλώσσες προγραμματισμού HTML, CSS και JavaScript.

Όσον αφορά το Backend (ή αλλιώς server-site), είναι το σημείο όπου γίνεται η επεξεργασία των HTTP αιτημάτων. Ο χρήστης δεν έχει πρόσβαση στο backend, όμως όταν ένα request γίνεται, τότε το backend είναι υπεύθυνο στο να το διαχειριστεί και να απαντήσει στον χρήστη. Εδώ, χρησιμοποίησα τις γλώσσες PHP, JavaScript, αλλά και το framework Laravel. Κατά την διαδικασία που γίνεται στο backend, ο web server (στην περίπτωση μου χρησιμοποιώ τον Apache), συνομιλεί με το File System (HTML, CSS, JavaScript) και με την βάση δεδομένων (MySQL) για να αποθηκεύει δεδομένα αλλά και για να ανακτά πληροφορίες. Η αρχιτεκτονική του συστήματος μου φαίνεται στο σχήμα της εικόνας 3.2.



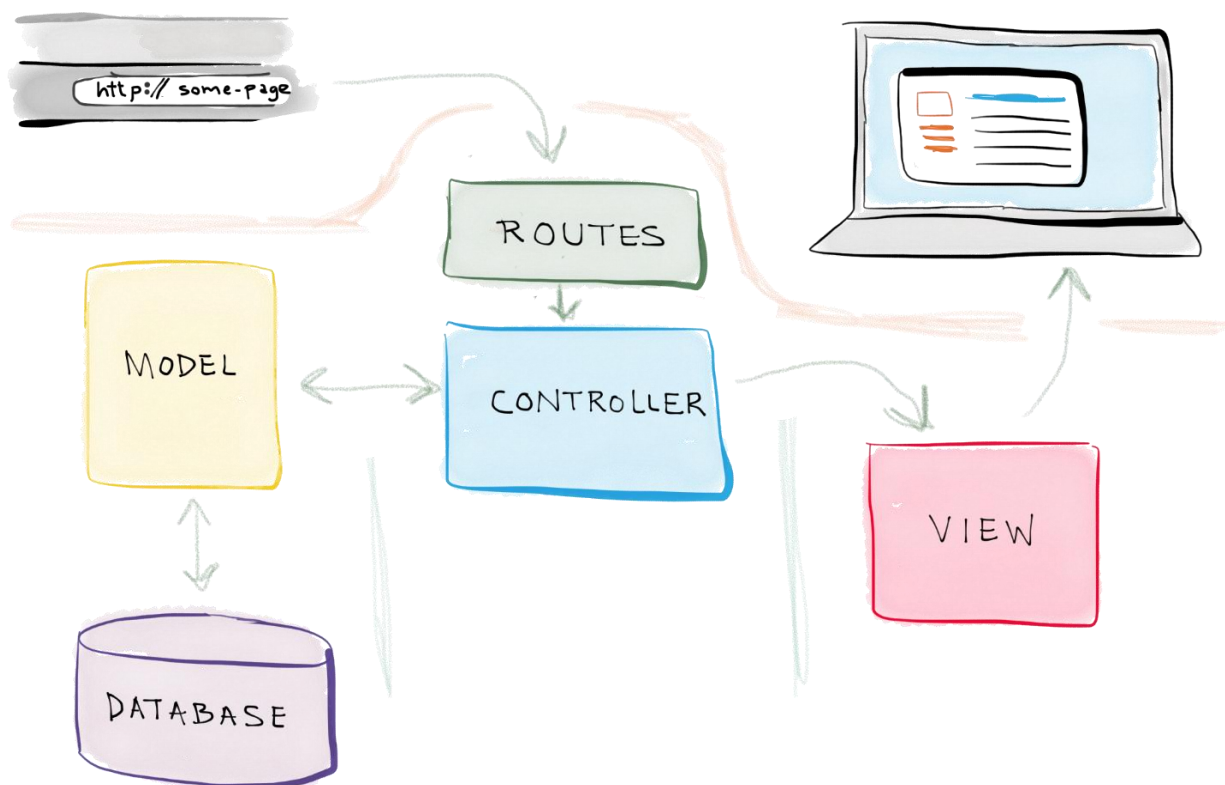
Εικόνα 3.2 : Αρχιτεκτονική Συστήματος

3.3 Απαιτούμενες Τεχνολογίες

Για την ανάπτυξη του συστήματος αναφοράς περιστατικών στους δήμους επέλεξα να χρησιμοποιήσω κάποιες συγκεκριμένες τεχνολογίες. Μελετήθηκε και χρησιμοποιήθηκε το framework Laravel, το οποίο χρησιμοποιεί τον Composer και τη γλώσσα προγραμματισμού PHP, η γλώσσα σήμανσης HTML και CSS αλλά και η γλώσσα JavaScript. Τέλος, για τη βάση δεδομένων μου επέλεξα την MySQL. Αυτές οι τεχνολογίες θα αναλυθούν και θα επεξηγηθούν πιο κάτω.

3.1.1 Laravel

Η Laravel είναι ένα Model-View-Controller framework το οποίο χρησιμοποιεί την γλώσσα προγραμματισμού PHP. [14] Αποτελεί ένα web application framework το οποίο έχει μία ρεαλιστική και δομημένη μορφή. Χρησιμοποιείται για τη δημιουργία διαδικτυακών εφαρμογών και έχει αρχίσει να χρησιμοποιείται από πολλές εταιρείες και προγραμματιστές.



Εικόνα 3.1.1 : MVC - Laravel

Όπως αναφέρθηκε και πριν η Laravel ακολουθεί το μοντέλο σχεδιασμού Model – View – Controller (MVC). [15] Αυτό που συμβαίνει όταν κάποιος χρήστης επισκέπτεται ένα συγκεκριμένο link είναι η διαδικασία που φαίνεται και στο σχήμα της εικόνας 3.1.5.2. Η Laravel αντιστοιχεί το link που έχει επισκεφθεί ο χρήστης με τις καθορισμένες ενέργειες στους controllers, χρησιμοποιώντας τα routes. Έτσι, όταν ένα αίτημα υποβάλλεται, δηλαδή όταν ο χρήστης εισάγει ένα URL, το οποίο βέβαια σχετίζεται με την εφαρμογή που δημιουργούμε με τη βοήθεια της Laravel, τότε ένα route αντιστοιχείται με μία ενέργεια του ελεγκτή (controller). Ο controller αξιοποιεί τα διάφορα μοντέλα για να ανακτά πληροφορίες από την βάση δεδομένων και ακολούθως προβάλλει αυτά τα δεδομένα στις οθόνες μας με τη χρήση των Views. Πιο αναλυτικά, το Model αναφέρεται σε ένα αντικείμενο ή αλλιώς σε ένα πόρο που θα διαχειρίζεται η εφαρμογή μας. Όσον αφορά τον Controller, είναι αυτός που θα διαχειρίζεται ένα σύνολο αντικειμένων. Τέλος, τα Views είναι τα διάφορα .blade files τα οποία περιέχουν τον κώδικα html και αντιπροσωπεύει τις διάφορες οθόνες που στο τέλος θα δει ο χρήστης.

Η σύνταξη της Laravel είναι αρκετά κομψή και εκφραστική. Προσπαθεί να αφαιρέσει τον κόπο από τους developers και να κάνει τη δημιουργία των web applications μία πιο εύκολη διαδικασία. Προσφέρει γρήγορη ανάπτυξη, ασφάλεια, πολύ καλή τεκμηρίωση και είναι αρκετά γρήγορη στην εκτέλεση. Επιπρόσθετα, είναι εύκολη στην κατανόηση και ένας νέος προγραμματιστής στο χώρο της Laravel μπορεί να μάθει πολύ γρήγορα να την χρησιμοποιεί και να αναπτύσσει συστήματα. Επίσης, διευκολύνει ορισμένες κοινές εργασίες που χρησιμοποιούνται στις πλείστες διαδικτυακές εφαρμογές, όπως το routing, το authorization, το authentication, το caching κ.α. Εκτός από αυτές τις διευκολύνσεις προσφέρει και αρκετά πακέτα για υποστήριξη διαφόρων λειτουργιών, τα οποία έχουν τη δυνατότητα να εξελιχθούν από τους προγραμματιστές.

Τα πλεονεκτήματα που αναφέρθηκαν πιο πάνω ήταν αρκετά για να με κάνουν να επιλέξω το συγκεκριμένο framework για την ανάπτυξη του συστήματός μου. Επιπλέον, η Laravel προσφέρει πολύ καλό documentation σε όσους θέλουν να ασχοληθούν με αυτήν. [16] Επίσης, το Laracasts, ένας ιστότοπος με πολλά tutorials για PHP, Laravel και διάφορες τεχνολογίες frontend, ήταν απίστευτα χρήσιμος στην πορεία μου να μάθω να χρησιμοποιώ το framework. [17] Ολοκληρώνοντας, η επιλογή μου να χρησιμοποιήσω το συγκεκριμένο framework δεν στάθηκε εμπόδιο καθ' όλη τη διάρκεια της ανάπτυξης του συστήματος. Αντιθέτως, με βοήθησε ιδιαίτερα και στο τέλος κατέληξα με μία επιπλέον γνώση, που σίγουρα θα με βοηθήσει στη μετέπειτα

επαγγελματική μου πορεία, αφού η Laravel έχει αρχίσει να διαδίδεται και να χρησιμοποιείται σε πολλές επιχειρήσεις.

Το Version της Laravel που χρησιμοποιώ είναι το 7.30.1, αν και το τελευταίο version που έχει βγει μέχρι αυτή τη στιγμή είναι το 8.16.1. Για να μπορέσω να κατεβάσω την Laravel στο virtual machine που μου δόθηκε και να δουλέψει σωστά έπρεπε πρώτα να κατεβάσω την PHP που θα αναλυθεί στο υποκεφάλαιο 3.5.2 αλλά και τον Composer που θα αναλυθεί στο υποκεφάλαιο 3.5.3. Τέλος, ως εργαλείο ανάπτυξης του κώδικα χρησιμοποίησα το text editor “Sublime Text 3”. [18]

3.1.2 Composer [19]

Ο Composer είναι ένα application-level package manager για την γλώσσα PHP και παρέχει μία καθορισμένη δομή για την διαχείριση εξαρτήσεων και βιβλιοθηκών της γλώσσας. [20] Μας επιτρέπει να προσθέτουμε, να αφαιρούμε αλλά και να διαχειριζόμαστε στα προγραμματιστικά έργα μας πακέτα, βιβλιοθήκες και εφαρμογές PHP που είναι διαθέσιμες στο “Packagist”, το κύριο repository που περιέχει όλα τα διαθέσιμα πακέτα. Ακόμη, παρέχει δυνατότητες αυτόματης φόρτωσης για ορισμένες βιβλιοθήκες και διευκολύνει τη χρήση κώδικα που έχει αναπτυχθεί από άλλους προγραμματιστές.

Ο Composer εκτελείτε από την γραμμή εντολών και είναι απαραίτητο εργαλείο για το framework Laravel. Η τελευταία έκδοση του Composer είναι η 2.0.13, όμως εγώ χρησιμοποιώ την έκδοση 1.10.13.

3.1.3 PHP

Η γλώσσα PHP είναι μία server-site scripting language. Παρόλο που είναι μία γενικής χρήσης γλώσσα προγραμματισμού χρησιμοποιείται κυρίως για την ανάπτυξη δυναμικών και διαδραστικών ιστοσελίδων. [21] Τα αρχικά PHP στην αρχή ερμηνεύονταν ως “Personal Home Page”, όμως πλέον ερμηνεύονται ως “Hypertext Preprocessor”.

Τα PHP αρχεία μπορούν να περιέχουν, εκτός από κώδικα PHP, και κώδικα HTML, CSS και JavaScript. Ο Server εκτελεί τον κώδικα και μας επιστρέφει το πρόγραμμα στον παγκόσμιο ιστό ως ένα απλό HTML file. Η PHP συμβάλλει στη δημιουργία δυναμικών σελίδων, αφού μπορεί να

δημιουργήσει, να ανοίξει, να διαβάσει, να γράψει, να διαγράψει και να κλείσει αρχεία στον server, κάτι που ένα απλό HTML file δεν μπορεί να κάνει. Παράλληλα, μας δίνει την δυνατότητα να συλλέγουμε δεδομένα από φόρμες και να τα αποθηκεύουμε στην βάση δεδομένων μας και να τα επεξεργαζόμαστε ή και ακόμα να κρυπτογραφεί δεδομένα, ώστε να προσφέρει στους χρήστες ασφάλεια. Επιπρόσθετα, μπορεί να διαχειρίζεται τα cookies, αλλά και να προσφέρει λειτουργίες όπως authentication και authorization για τους χρήστες.

Η PHP είναι ευρέως χρησιμοποιούμενη γλώσσα ανοιχτού κώδικα και αρκετά δημοφιλής. Είναι αρκετά εύκολη στην εκμάθηση και συστήνεται σαν πρώτη server-side γλώσσα για αρχάριους. Εκτελείται σε διάφορες πλατφόρμες (Windows, MacOS, Linux, Unix κ.α.) και είναι συμβατή σχεδόν με όλους τους servers που χρησιμοποιούνται σήμερα (Apache, IIS κ.α.). Τέλος, υποστηρίζει αρκετές βάσεις δεδομένων και παρέχει πολλές δυνατότητες στον προγραμματιστή. Χρησιμοποιείται από το Facebook, από το WordPress αλλά και από την Laravel.

Το τελευταίο version που εμφανίστηκε στις 29 του Απρίλη 2021 είναι το 8.0.5 και εγώ χρησιμοποιώ το version 7.2.24.

3.1.4 HTML

Η HTML είναι η μία γλώσσα σήμανσης και είναι η τυπική γλώσσα για ανάπτυξη ιστοσελίδων. [22] Τα αρχικά HTML ερμηνεύονται ως “HyperText Markup Language”. [23] [24] Με τον όρο Markup (σήμανση) αναφερόμαστε στις διάφορες ετικέτες (tags) που αντιστοιχούν στο περιεχόμενο της σελίδας. Μερικές από αυτές είναι <p>, για παραγράφους, , για εικόνες, <audio>, για ήχο, <video>, για βίντεο και πολλές άλλες. Με αυτόν τον τρόπο καθορίζεται το περιεχόμενο των ιστοσελίδων αφού τα διάφορα προγράμματα περιήγησης ερμηνεύουν τις διάφορες ετικέτες και εμφανίζουν την κατάλληλη πληροφορία στην οθόνη του χρήστη.

Το τελευταίο version που χρησιμοποιώ και εγώ είναι το HTML 5. Μέσω της Laravel δημιουργούνται τα blade files τα οποία αντιστοιχούν στα διάφορα views, που αναφέρθηκαν και πριν στο υποκεφάλαιο 3.5.1. Τα blade files είναι ουσιαστικά php files τα οποία περιέχουν κώδικα σε HTML. Με την χρήση των blade files η Laravel μας επιτρέπει να επαναχρησιμοποιούμε κώδικα

και να μην επαναλαμβάνουμε σημεία που φαίνονται σε κάθε σελίδα όπως το header, το navigation bar και το footer.

3.1.5 CSS

Το CSS είναι μία style sheet γλώσσα που μαζί με την HTML αποτελούν τη βάση για τη δημιουργία ιστοσελίδων. [25] Ενώ η γλώσσα HTML καθορίζει το περιεχόμενο των ιστοσελίδων η γλώσσα CSS καθορίζει πως αυτό το περιεχόμενο θα παρουσιαστεί. Τα αρχικά CSS σημαίνουν Cascading Style Sheets.

Η μορφοποίηση του περιεχομένου συμπεριλαμβάνει τη διάταξη του κειμένου, των φωτογραφιών και των βίντεο, τις διάφορες γραμματοσειρές που θα επιλεγθούν για τα γράμματα αλλά και τα χρώματα με τα οποία θα αποδοθούν. Γενικότερα, το CSS αποδίδει το στυλ ή αλλιώς την εμφάνιση του περιεχομένου των ιστοσελίδων το οποίο διαχωρίζεται από τα αρχεία HTML συνήθως σε ξεχωριστά files. Ο διαχωρισμός αυτός γίνεται για να μην δημιουργείται σύγχυση όταν τροποποιούνται είτε το στυλ είτε το περιεχόμενο αλλά και για να δίνεται η δυνατότητα επαναχρησιμοποίησης του CSS κώδικα από πολλά HTML files.

Η τελευταία έκδοση του CSS είναι το CSS 3 το οποίο και χρησιμοποιώ. Στην ιεραρχία της Laravel τα CSS files τοποθετούνται στο φάκελο “public”, για να μπορεί η Laravel να τα ανακτά εφόσον σε αυτόν το φάκελο όλα τα αρχεία είναι publicly available.

3.1.6 JavaScript

Η JavaScript(JS) είναι και αυτή μία από τις βασικές γλώσσες προγραμματισμού για ανάπτυξη ιστοσελίδων μαζί με την HTML, CSS και PHP. Σε αντίθεση με την PHP, η JavaScript είναι μία client-site scripting γλώσσα προγραμματισμού και μας δίνει τη δυνατότητα να δημιουργήσουμε μία δυναμική και διαδραστική σελίδα από τη μεριά του πελάτη. [26]

Μας επιτρέπει να εφαρμόζουμε πιο σύνθετες λειτουργίες και να κάνει τις ιστοσελίδες πιο ενδιαφέρουσες, αφού δεν θα μας δείχνουν μόνο απλές στατικές πληροφορίες. Μερικά παραδείγματα είναι η ενημέρωση του περιεχομένου της σελίδας, η εμφάνιση διαδραστικών χαρτών, κινούμενων εικόνων και βίντεο και πολλές άλλες λειτουργίες που καθιστούν τη σελίδα

δυναμική και όχι στατική. [27] Παρόλο που ο κώδικας JavaScript μπορεί να βρίσκεται μαζί με τον HTML κώδικα, συνήθως διαχωρίζονται, όπως ακριβώς και με το CSS, για επαναχρησιμοποίηση του JS κώδικα και για να μην υπάρχει σύγχυση.

Η τελευταία έκδοση του JavaScript είναι η ECMAScript 2020. Όπως και με τα αρχεία CSS, Έτσι και τα αρχεία JavaScript τοποθετούνται στον φάκελο “public” στην ιεραρχία της Laravel για να είναι publicly available.

3.1.7 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα. [28] [29]. Μία βάση δεδομένων είναι υπεύθυνη για να αποθηκεύει τα δεδομένα, να τα οργανώνει σε πίνακες, να τα ταξινομεί αλλά και να τα επιστρέφει στο χρήστη μετά από συγκεκριμένη αναζήτηση. Το όνομα MySQL είναι συνδυασμός των “My”, το όνομα της κόρης του συνιδρυτή Michael Widenius, και “SQL”, η συντομογραφία για το Structured Query Language.

Με την χρήση της γλώσσας SQL οι προγραμματιστές μπορούν πολύ εύκολα να τροποποιήσουν την βάση εισάγοντας, διαγράφοντας ή αλλάζοντας δεδομένα μέσα. Παράλληλα μπορούν να ανακτήσουν πληροφορίες από τη βάση και να τις χρησιμοποιήσουν για όποιο σκοπό χρειάζονται. Η MySQL είναι ιδανική τόσο και για μεγάλα όσο και για μικρά συστήματα και λειτουργεί από την μεριά του server. Είναι αρκετά γρήγορη, αξιόπιστη και εύχρηστη και για αυτό την επέλεξα για να την χρησιμοποιήσω.

Το τελευταίο version της MySQL που υπάρχει είναι το MySQL 8.0, το οποίο και χρησιμοποιώ. Το συγκεκριμένο version που επέλεξα έχει μεγάλη σημασία, γιατί περιλαμβάνει ορισμένα queries, τα οποία είναι απαραίτητα και δεν υπήρχαν σε προηγούμενα versions. Π.χ. υπάρχει το “ST_Contains”, το οποίο μας επιτρέπει να ελέγχουμε αν ένα σημείο βρίσκεται μέσα σε ένα πολύγωνο. Το συγκεκριμένο query ήταν αρκετά σημαντικό για το σύστημα που αναπτύχθηκε, έτσι ώστε να βρίσκει σε ποιο δήμο ανήκει το κάθε report που αναφέρει κάποιος χρήστης, με βάση τα γεωγραφικά όρια του δήμου και το γεωγραφικό σημείο του περιστατικού.

Η Laravel, υποστηρίζει τέσσερις βάσεις δεδομένων, την MySQL 5.7+, την PostgreSQL 9.6+, την SQLite 3.8.8+ και τον SQL Server 2017+. Με τη Laravel η αλληλεπίδραση με τη βάση δεδομένων

καθίσταται αρκετά απλή αφού χρησιμοποιεί ακατέργαστη την γλώσσα SQL, ένα εργαλείο δημιουργίας query και το Eloquent ORM [30], ένα πολύ ισχυρό και εκφραστικό εργαλείο της PHP, που μας επιτρέπει να αλληλοεπιδρούμε με τα διάφορα αντικείμενα και τις σχέσεις των βάσεων δεδομένων μας, με μεγάλη ευκολία [31]. Περισσότερα για τη σχεδίαση της βάσης δεδομένων θα αναλυθούν στο υποκεφάλαιο 3.2 που ακολουθεί.

3.4 Σχεδιασμός Βάσης Δεδομένων

Η βάση δεδομένων μου αποτελείται από τους πιο κάτω πίνακες:

Πίνακας cities: Αποθηκεύει τις 5 επαρχίες της Κύπρου. Αποτελείται από ένα μοναδικό id και το όνομα(name) της πόλης. Συνδέεται με τον πίνακα municipalities με τη σχέση ένα προς πολλά, αφού μία πόλη μπορεί να περιέχει πολλούς δήμους, όμως ένας δήμος μπορεί να ανήκει σε μία πόλη.

Πίνακας municipalities: Αποθηκεύει τους δήμους της Κύπρου. Αποτελείται από ένα μοναδικό id και το όνομα(name) του δήμου, τα γεωγραφικά όρια του δήμου (coordinates) που έχουν τύπο “Multipolygon” και ουσιαστικά αποτελούνται από διάφορα γεωγραφικά σημεία και τέλος, το foreign key city_id. Συνδέεται με τον πίνακα cities με την σχέση ένα προς πολλά αφού μία πόλη μπορεί να περιέχει πολλούς δήμους, όμως ένας δήμος μπορεί να ανήκει σε μία πόλη. Επίσης, συνδέεται με τον πίνακα users με την σχέση ένα προς πολλά, αφού ένας δήμος μπορεί να περιέχει πολλούς χρήστες, όμως ένας χρήστης ανήκει σε μόνο ένα δήμο. Συνδέεται ακόμα και με τον πίνακα reports με την σχέση ένα προς πολλά, αφού ένας δήμος περιέχει πολλά reports, όμως ένα report ανήκει σε ένα μόνο δήμο.

Πίνακας reports: Αποθηκεύει τις αναφορές που κάνουν οι χρήστες στο σύστημα. Αποτελείται από ένα μοναδικό id και τα στοιχεία του report που είναι τα εξής: τύπος, περιγραφή, status (unseen, seen, in progress, solved, blocked), γεωγραφικό μήκος και πλάτος, τα paths των φωτογραφιών που αποθηκεύτηκαν στο server, ώστε να μπορούμε να τα ανακτούμε (photopath1, photopath2, photopath3), τότε δημιουργήθηκε και ενημερώθηκε η αναφορά, τα foreign keys user_id και municipality_id, αλλά και το score, έναν αριθμό από 0 μέχρι 100, που καθορίζει πόσο έγκυρη ήταν η αναφορά. Συνδέεται με τον πίνακα municipalities με την σχέση ένα προς πολλά αφού μία

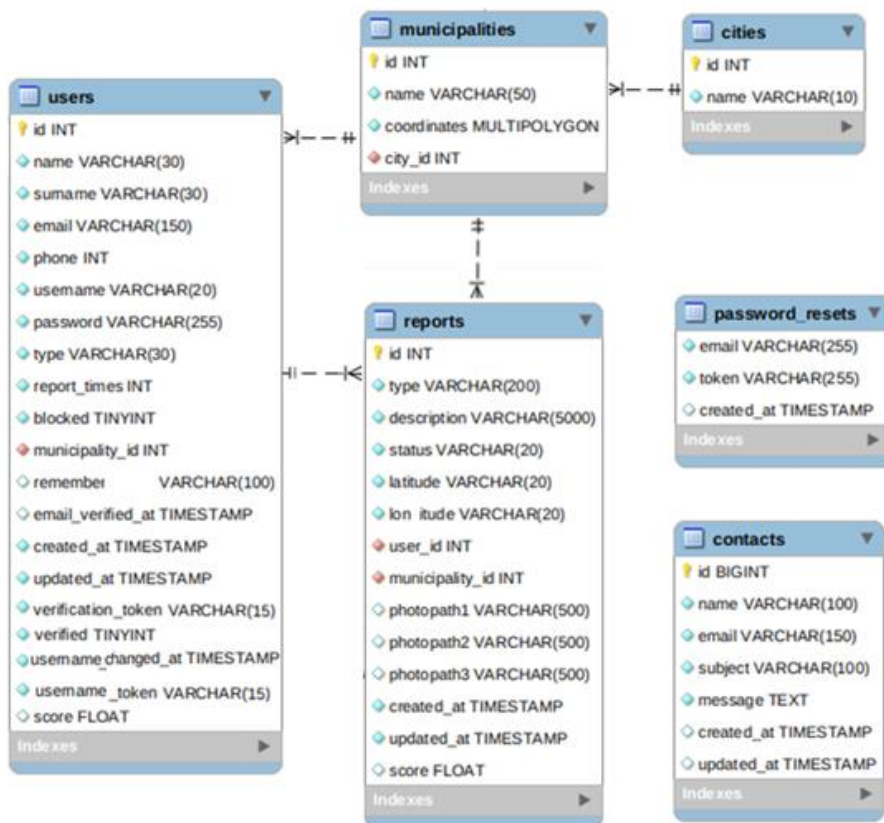
αναφορά μπορεί να ανήκει σε μόνο ένα δήμο, αλλά ένας δήμος μπορεί να περιέχει πολλές αναφορές. Συνδέεται και με τον πίνακα users με την σχέση ένα προς πολλά, αφού ένας χρήστης μπορεί να αναφέρει πολλά reports και ένα report ανήκει σε ένα μόνο χρήστη.

Πίνακας users: Αποθηκεύει τα στοιχεία των χρηστών. Αποτελείται από ένα μοναδικό id και το όνομα και επίθετο του χρήστη, το email και το τηλέφωνό του, το username και το password του κρυπτογραφημένο, τον τύπο του (simple, operator, administrator), τότε δημιουργήθηκε και ανανεώθηκε ο λογαριασμός του χρήστη, το remember_token για ανανέωση του password και το foreign key municipality_id. Επιπρόσθετα σαν στοιχεία αποθηκεύονται και το πόσες φορές έχει γίνει ένα report του χρήστη blocked από τους υπεύθυνους, αφού θεώρησαν πως το report περιείχε ακατάλληλο περιεχόμενο(report_times) και αν ο χρήστης έχει γίνει blocked, το οποίο προκύπτει αν τα report_times γίνουν 3(blocked). Τέλος, αποθηκεύεται και το score το οποίο προκύπτει από τον μέσο όρο των scores των solved reports που έχει κάνει ο χρήστης. Συνδέεται με τον πίνακα municipalities με τη σχέση ένα προς πολλά, αφού ένας δήμος μπορεί να περιέχει πολλούς χρήστες, όμως ένας χρήστης ανήκει σε μόνο ένα δήμο. Σύνδεση ένα προς πολλά υπάρχει και με τον πίνακα reports, αφού ένας χρήστης μπορεί να αναφέρει πολλά reports και ένα report ανήκει σε ένα μόνο χρήστη.

Πίνακας password_resets: Αποθηκεύει τις απαραίτητες πληροφορίες για να γίνει αλλαγή του password των χρηστών. Αποτελείται από το field email, που όμως μέσα αποθηκεύει το μοναδικό username του χρήστη. Η αλλαγή αυτή προέκυψε λόγω του ότι η Laravel προσφέρει μία έτοιμη λειτουργία για authentication και authorization την οποία και χρησιμοποίησα, όμως στη δικιά μου περίπτωση χρησιμοποιώ το username και όχι το email που προσφέρει η Laravel σαν μοναδικό πεδίο. Ακόμα, αποθηκεύεται ένα token που δημιουργείται τυχαία και το πότε αυτή η αίτηση για αλλαγή κωδικού έχει καταχωρηθεί, ώστε μετά από 60 λεπτά να λήξει η διορία που έχει ο χρήστης για να αλλάξει κωδικό. Δεν συνδέεται με κάποιο πίνακα.

Πίνακας contacts: Αποθηκεύει τις πληροφορίες που προέρχονται από την φόρμα Contact Us του συστήματος. Οι πληροφορίες αυτές είναι ένα μοναδικό id, το όνομα αυτού που έστειλε το μήνυμα και το email του, το θέμα, το μήνυμα και πότε στάλθηκε ή ανανεώθηκε. Δεν συνδέεται με κάποιο πίνακα.

Πιο κάτω στην εικόνα 3.2 βλέπουμε το ER Diagram της βάσης δεδομένων, που παρουσιάζει όλα όσα αναλύθηκαν σε αυτό το υποκεφάλαιο 3.2 για το σχεδιασμό της βάσης δεδομένων.



Εικόνα 3.2 : ER Diagram

Η δημιουργία των πινάκων στη βάση δεδομένων έγιναν με τη βοήθεια των migrations, που προσφέρει η Laravel. Τα migrations είναι ένας τρόπος με τον οποίο μπορούμε να δημιουργήσουμε έναν πίνακα στη βάση δεδομένων, χωρίς να μεταβούμε στην πραγματικότητα στη διαχείριση της βάσης. Αυτό μας δίνει τη δυνατότητα να επιστρέψουμε σε παλιές εκδόσεις που είχαμε στη βάση και επίσης διευκολύνει την ομαδική εργασία. Ακόμη τα migrations μας επιτρέπουν να αφαιρέσουμε ή να προσθέσουμε στη βάση πεδία χωρίς να πειραχτούν οι ήδη υπάρχουσες εγγραφές

πάνω. Στο Παράρτημα Α φαίνονται τα migrations που δημιούργησα για τους πίνακες στη βάση δεδομένων μου.

Κεφάλαιο 4

Υλοποίηση Συστήματος

4.1 Guest – Επισκέπτης	27
4.1.1 Αρχική Σελίδα	28
4.1.2 Contact Us	31
4.1.3 Sign Up	32
4.1.4 Login	33
4.2 Simple User - Πολίτης	37
4.2.1 Αρχική Σελίδα	37
4.2.2 Report A Problem	38
4.2.3 User's Details	40
4.2.4 Report History	42
4.3 Operator – Υπάλληλος Δήμου	43
4.3.1 Management	43
4.3.1.1 Reports	44
4.3.1.2 Users	45
4.4 Administrator – Διαχειριστής Συστήματος	47

4.1 Guest - Επισκέπτης

Ο απλός επισκέπτης της ιστοσελίδας δεν έχει πολλές δυνατότητες. Μπορεί να διαβάσει πληροφορίες για το σύστημα, να κάνει sign up αν δεν έχει λογαριασμό ή login αν έχει δημιουργήσει λογαριασμό και να στείλει email στους υπεύθυνους της σελίδας, για τυχόν απορίες ή βελτιώσεις που έχει να προτείνει. Δεν μπορεί να αναφέρει κάποιο πρόβλημα εάν δεν έχει λογαριασμό. Πιο κάτω θα παρουσιαστούν αναλυτικά όλες οι λειτουργίες που μπορεί να κάνει ο επισκέπτης και θα επεξηγηθούν τα βήματα που θα χρειαστεί να ακολουθήσει.

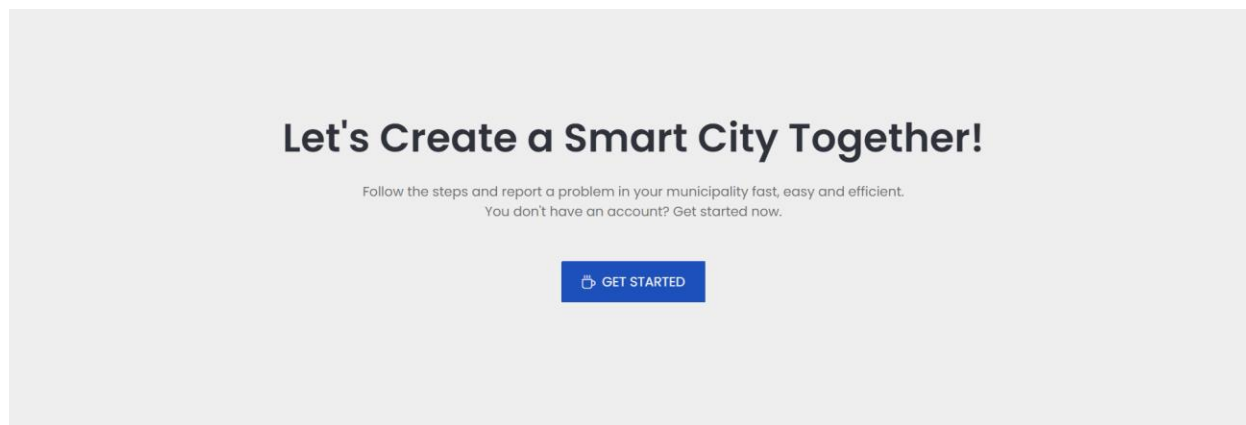
4.1.1 Αρχική Σελίδα

Στην αρχική σελίδα όπως και σε όλες τις υπόλοιπες σελίδες συναντούμε το navigation bar, το οποίο παραμένει στο πάνω μέρος της οθόνης ακόμα και αν μετακινηθούμε πιο κάτω για να δούμε και την υπόλοιπη σελίδα, και περιλαμβάνει το logo και το menu. Στο menu υπάρχουν όλες οι επιλογές που μπορεί να κάνει ένας απλός επισκέπτης πριν δημιουργήσει λογαριασμό και συνδεθεί. Το navigation bar και το 1^ο μέρος της αρχικής σελίδας φαίνονται στην εικόνα 4.1.1.1.



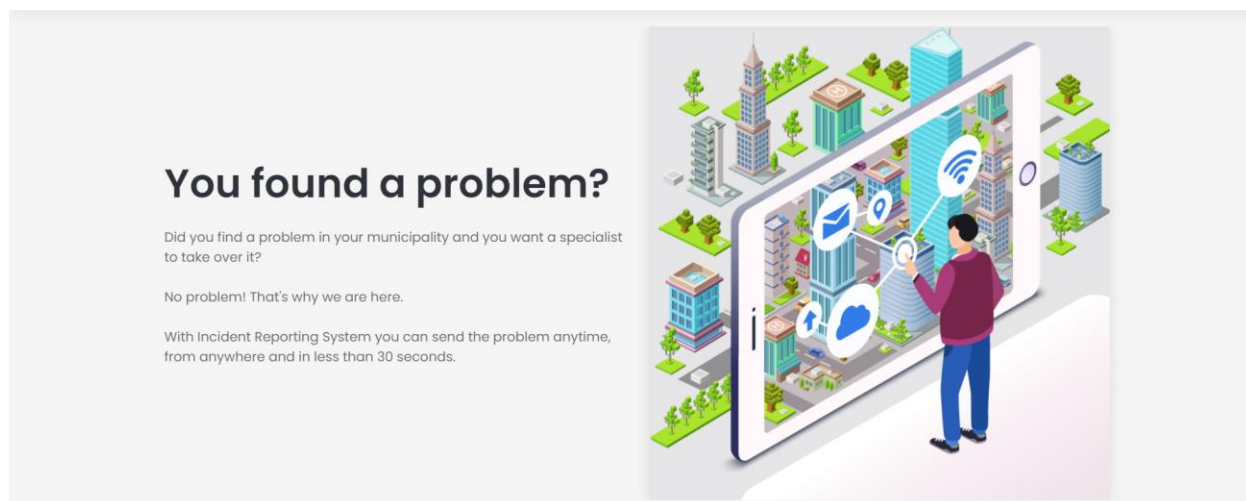
Εικόνα 4.1.1.1 : Αρχική Σελίδα (1)

Προχωρώντας πιο κάτω το σύστημα προσπαθεί να ενθαρρύνει τον επισκέπτη να δημιουργήσει λογαριασμό στη σελίδα, για να μπορέσει και αυτός να συμβάλει θετικά στη δημιουργία μίας έξυπνης πόλης. Το κουμπί “Get Started” τον οδηγεί στην φόρμα του Sign Up για να δημιουργήσει λογαριασμό στο σύστημα. Το συγκεκριμένο τμήμα της αρχικής σελίδας φαίνεται στην εικόνα 4.1.1.2.



Εικόνα 4.1.1.2 : Αρχική Σελίδα (2)

Ακολουθώντας, όπως φαίνεται και στην εικόνα 4.1.1.3, η σελίδα μας δίνει μερικές πληροφορίες για το σύστημα. Μας εξηγεί το σκοπό του συστήματος και αναφέρει μερικά στοιχεία, όπως ότι μπορεί να χρησιμοποιηθεί από οπουδήποτε, οποιαδήποτε ώρα και η αναφορά ενός προβλήματος μπορεί να διαρκέσει μερικά δευτερόλεπτα.



Εικόνα 4.1.1.3 : Αρχική Σελίδα (3)

Ακόμη, η εικόνα 4.1.1.4 μας δείχνει το κομμάτι της αρχικής σελίδας που εξηγεί στο χρήστη τα βήματα που πρέπει να ακολουθήσει για να αναφέρει ένα πρόβλημα που έχει εντοπίσει.

How to report a problem?



1. Create an account

First, you have to create an account. Sign up, fill in your data and then you can login to report a problem.



2. Report a Problem

Take a picture of the problem, write a small description and choose the location where the incident is taking place.

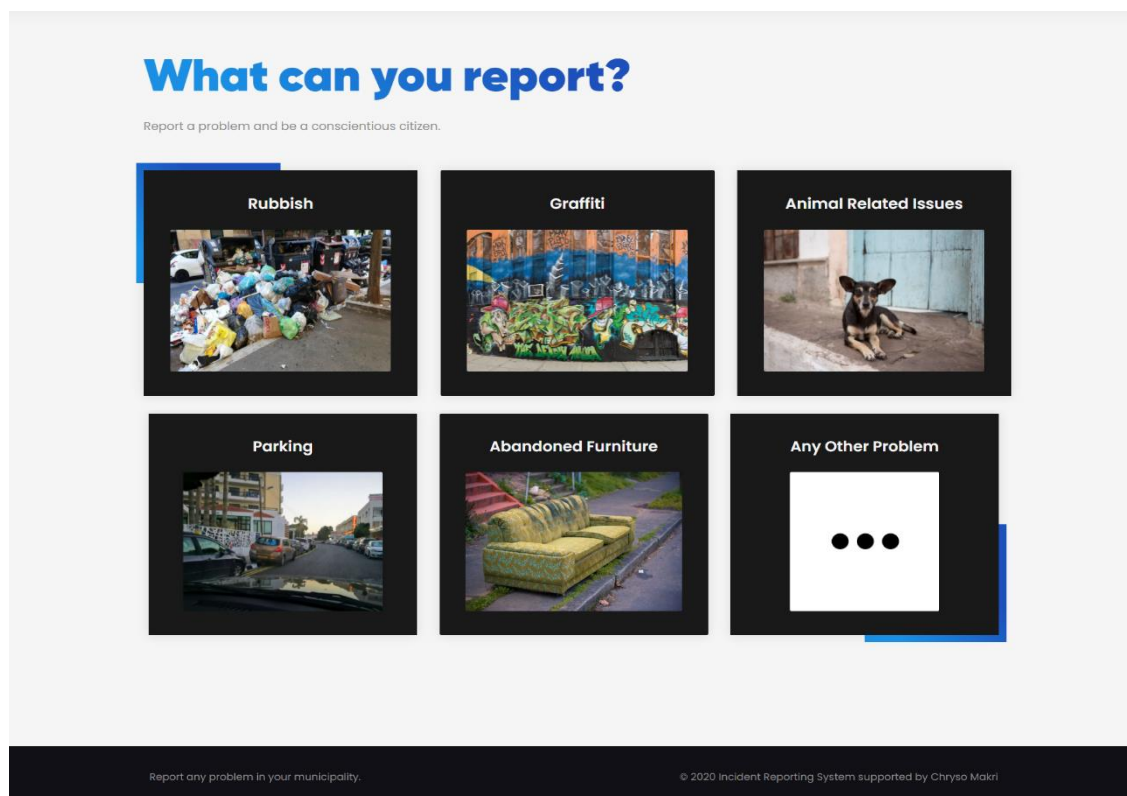


3. We solve it

The incident is stored in our database and it will be handled as soon as possible.

Εικόνα 4.1.1.4 : Αρχική Σελίδα (4)

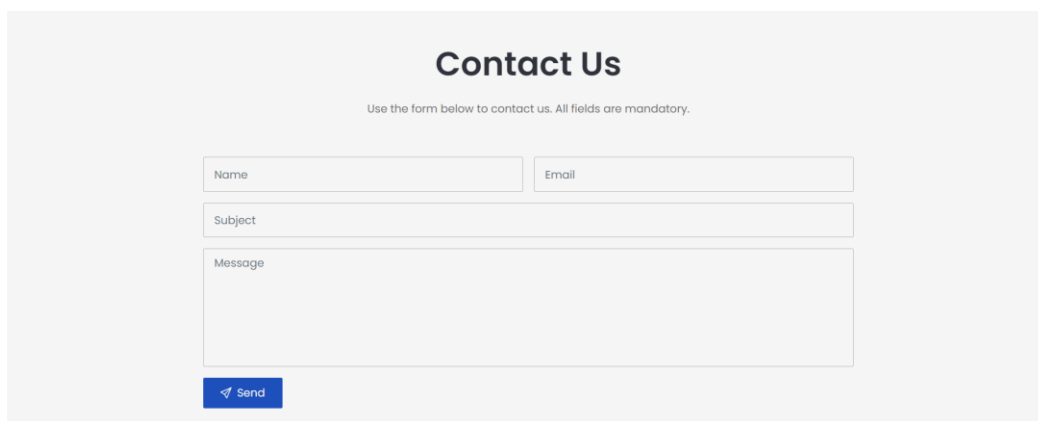
Τέλος, όπως βλέπουμε στην εικόνα 4.1.1.5, το σύστημα μας αναφέρει ενημερωτικά μερικά από τα βασικά προβλήματα που μπορεί ένας χρήστης να αναφέρει. Επίσης, στο κάτω μέρος της οθόνης φαίνεται και το footer της σελίδας, το οποίο παραμένει σταθερό σε κάθε σελίδα.



Εικόνα 4.1.1.5 : Αρχική Σελίδα (5)

4.1.2 Contact Us

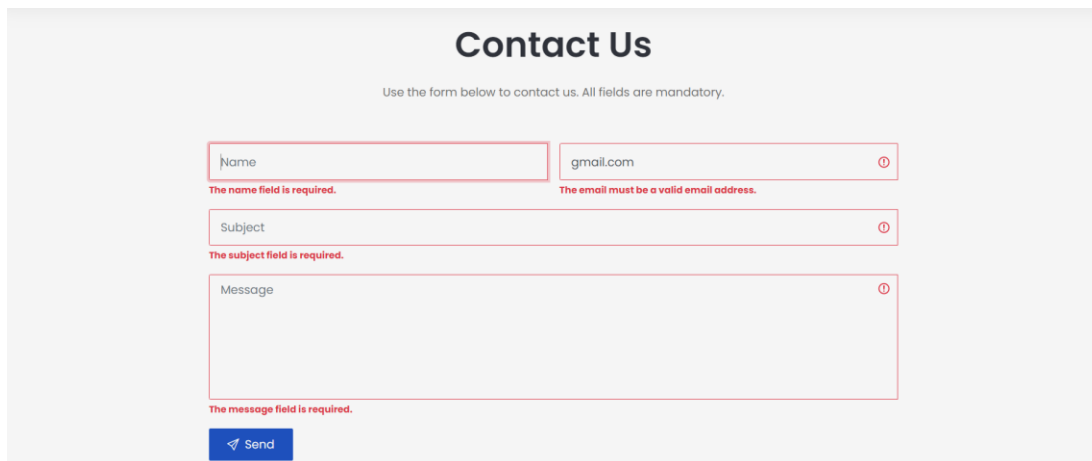
Στην σελίδα Contact Us ο χρήστης έχει την ευκαιρία να στείλει ένα email στους υπεύθυνους του συστήματος είτε για να λυθούν κάποιες απορίες, είτε για να ενημερώσει ότι εντόπισε κάποιο λάθος, είτε για να προτείνει ιδέες για βελτίωση του συστήματος. Μπορεί πολύ εύκολα να συμπληρώσει τα απαραίτητα στοιχεία της φόρμας contact us που φαίνεται και στην εικόνα 4.1.2 και να πατήσει το “Sent”.



The screenshot shows a 'Contact Us' form with the title 'Contact Us' and a subtitle 'Use the form below to contact us. All fields are mandatory.' The form contains four input fields: 'Name', 'Email', 'Subject', and 'Message'. The 'Message' field is a larger text area. At the bottom left of the form is a blue button with a white paper plane icon and the text 'Send'.

Εικόνα 4.1.2.1 : Contact Us (1)

Ο χρήστης πρέπει να συμπληρώσει σωστά τα στοιχεία του και αυτό περιλαμβάνει τη συμπλήρωση όλων των στοιχείων και την εισαγωγή μίας σωστής ηλεκτρονικής διεύθυνσης. Αν ο χρήστης κάνει κάποιο λάθος εμφανίζονται τα ανάλογα μηνύματα λάθος, όπως εμφανίζονται στην εικόνα 4.1.2.2, για να ειδοποιήσουν τον χρήστη.

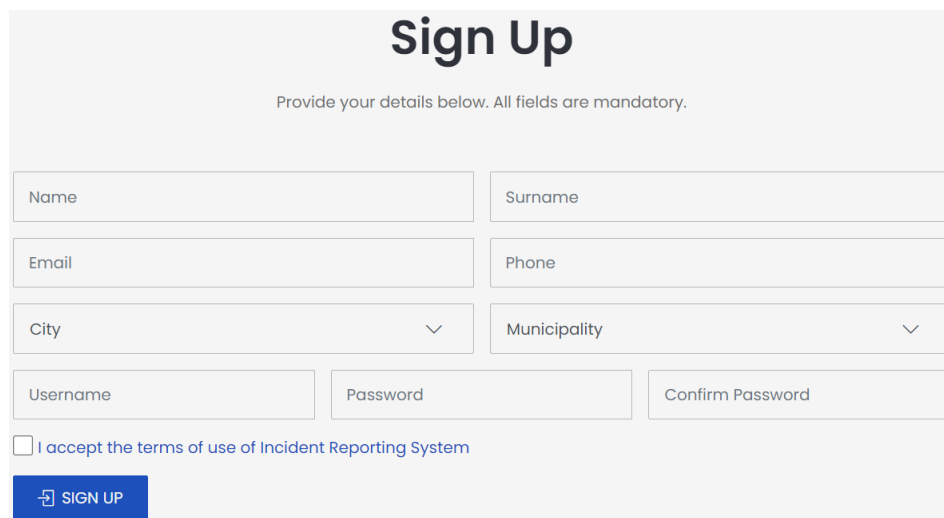


The screenshot shows the same 'Contact Us' form as in the previous image, but with validation error messages. The 'Name' field has a red border and the message 'The name field is required.' below it. The 'Email' field has a red border, contains the text 'gmail.com', and has the message 'The email must be a valid email address.' below it. The 'Subject' field has a red border and the message 'The subject field is required.' below it. The 'Message' field has a red border and the message 'The message field is required.' below it. The 'Send' button is still present at the bottom left.

Εικόνα 4.1.2.2 : Contact Us (2)

4.1.3 Sign Up

Στην σελίδα Sign Up ο χρήστης μπορεί να δημιουργήσει λογαριασμό συμπληρώνοντας ορθά όλα τα στοιχεία που φαίνονται στην εικόνα 4.1.3.1.



The image shows a 'Sign Up' form with the following fields: Name, Surname, Email, Phone, City (dropdown), Municipality (dropdown), Username, Password, and Confirm Password. Below the fields is a checkbox for 'I accept the terms of use of Incident Reporting System' and a blue 'SIGN UP' button.

Εικόνα 4.1.3.1 : Sign Up (1)

Πρέπει να ικανοποιηθούν όλοι οι περιορισμοί, που είναι οι εξής:

- Όλα τα πεδία πρέπει να συμπληρωθούν.
- Το όνομα και το επίθετο πρέπει να αποτελούνται μόνο από γράμματα.
- Το email πρέπει να ακολουθεί την σωστή μορφή των emails.
- Το τηλέφωνο πρέπει να αποτελείται από 8 αριθμούς.
- Το username πρέπει να μην χρησιμοποιείται από άλλο άτομο.
- Ο κωδικός πρέπει να αποτελείται από τουλάχιστον 8 χαρακτήρες και η επαλήθευση κωδικού να είναι η ίδια με τον κωδικό.
- Οι όροι του συστήματος πρέπει να γίνουν αποδεκτοί.

Όπως φαίνεται και στην εικόνα 4.1.3.1, σε περίπτωση που ο χρήστης εισάγει κάτι λανθασμένα ή δεν συμπλήρωσε ένα πεδίο, το σύστημα του εμφανίζει μηνύματα λάθους όταν πατήσει “Sign Up”, για να τα διορθώσει.

The screenshot shows a 'Sign Up' form with the following fields and errors:

- Name:** Input field contains '123'. Error: 'The name may only contain letters.'
- Surname:** Input field is empty. Error: 'The surname field is required.'
- Email:** Input field contains 'gmail.com'. Error: 'The email must be a valid email address.'
- Phone:** Input field contains '1'. Error: 'The phone must be 8 digits.'
- City:** Input field is empty. Error: 'The city field is required.'
- Municipality:** Input field is empty. Error: 'The municipality field is required.'
- Username:** Input field contains 'cmakri02'. Error: 'The username has already been taken.'
- Password:** Input field contains '***'. Error: 'The password must be at least 8 characters.'
- Repassword:** Input field contains '....'. Error: 'The repassword and password must match.'

At the bottom, there is a checkbox for 'I accept the terms of use of Incident Reporting System' with the error 'The gdpr must be accepted.' and a blue 'SIGN UP' button.

Εικόνα 4.1.3.2 : Sign Up (2)

Αφού γίνει σωστά η διαδικασία εγγραφής, ένα verification email θα έρθει στο email account που έχουμε δηλώσει. Το email περιλαμβάνει ένα μοναδικό token, καθώς και το σύνδεσμο για την σελίδα όπου θα γίνει το verification. Η σελίδα αυτή φαίνεται στην εικόνα 4.1.3.3. Ο χρήστης θα συμπληρώσει το username και το password του, αλλά και το μοναδικό token που έχει λάβει στο email του. Αφού πατήσει submit και εάν όλα τα στοιχεία συμπληρωθούν σωστά, τότε η διαδικασία του sign up θα ολοκληρωθεί με επιτυχία.

4.1.4 Login

Αν ο χρήστης έχει πλέον δημιουργήσει λογαριασμό στο σύστημα μπορεί να κάνει login συμπληρώνοντας το username και το password του. Επίσης, δίνεται η δυνατότητα για να ανανεώσει τον κωδικό του, όπως επίσης και το link για μετάβαση στη σελίδα του sign up, για να δημιουργήσει λογαριασμό εάν δεν έχει. Η σελίδα Sign Up φαίνεται στη εικόνα 4.1.4.1.

Login

Provide your details below. All fields are mandatory.

Username Password

Login

[Forgot your username?](#)

[Forgot your password?](#)

[You don't have an account?](#)

Εικόνα 4.1.4.1 : Login (1)

Όπως φαίνεται και στην εικόνα 4.1.4.2, το σύστημα ειδοποιεί τον χρήστη να συμπληρώσει τα στοιχεία του εάν πατήσει το “Login” πριν εισάγει αυτά που χρειάζονται.

Login

Provide your details below. All fields are mandatory.

Username Password

Login

Please fill in this field.

Εικόνα 4.1.4.2 : Login (2)

Επίσης, όπως φαίνεται και στην εικόνα 4.1.4.3, εάν ο χρήστης συμπληρώσει λάθος τα στοιχεία του ή δεν συμπληρώσει κάποιο από αυτά το σύστημα εμφανίζει τα απαραίτητα μηνύματα λάθους.

Login

Provide your details below. All fields are mandatory.

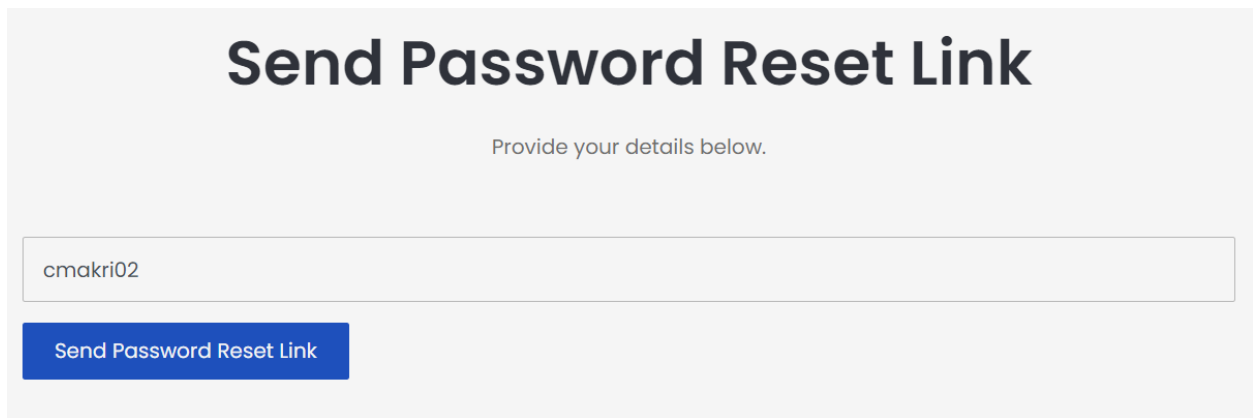
cmakri02 Password

These credentials do not match our records.

Login

Εικόνα 4.1.4.3 : Login (3)

Όσον αφορά τη διαδικασία του forgot password, όταν πατήσουμε στο link από την σελίδα login εμφανίζεται η σελίδα που βλέπουμε στην εικόνα 4.1.4.4.



Send Password Reset Link

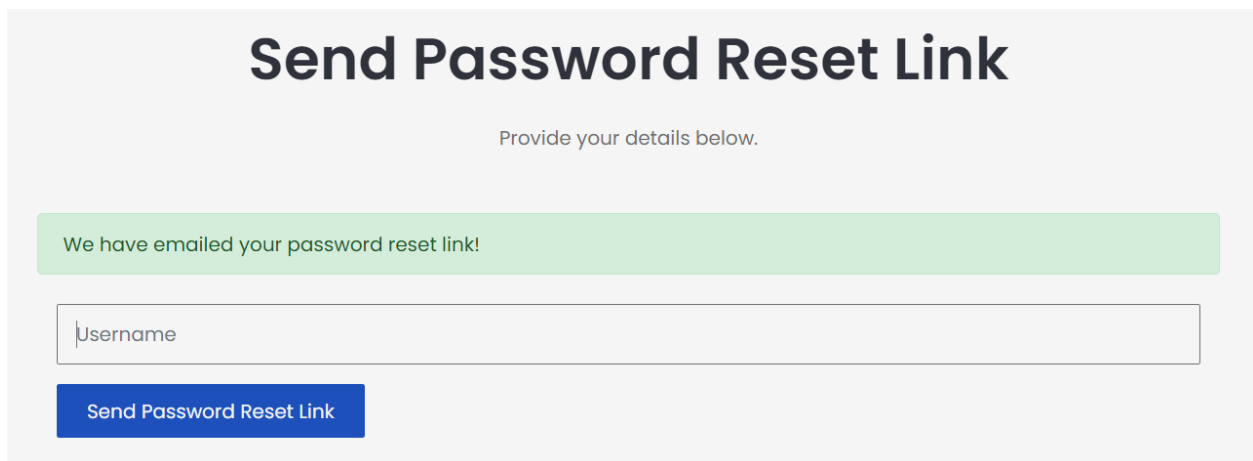
Provide your details below.

cmakri02

Send Password Reset Link

Εικόνα 4.1.4.4 : Password Reset (1)

Αφού συμπληρώσουμε το username μας και πατήσουμε “Send” το σύστημα μας ενημερώνει πως μας έστειλε ένα link στο email μας για να ανανεώσουμε τον κωδικό μας. Οι αντίστοιχες εικόνες είναι οι 4.1.4.5 και 4.1.4.6.



Send Password Reset Link

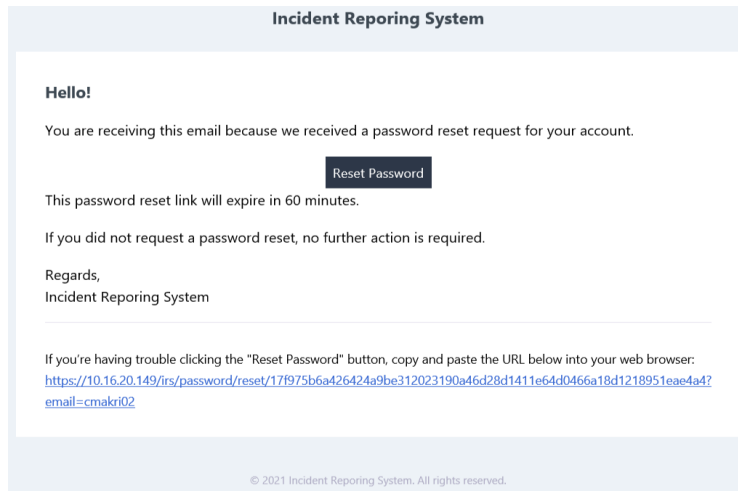
Provide your details below.

We have emailed your password reset link!

Username

Send Password Reset Link

Εικόνα 4.1.4.5 : Password Reset (2)



Εικόνα 4.1.4.6 : Password Reset (3)

Τέλος, αφού πατήσουμε το link που έχει σταλθεί στο email μας, οδηγούμαστε στη σελίδα που φαίνεται στην εικόνα 4.1.4.7 και συμπληρώνουμε τα απαραίτητα στοιχεία σωστά για να αλλάξουμε τον κωδικό μας.

The image shows a 'Reset Password' form. The title is 'Reset Password' with the subtitle 'Provide your details below. All fields are mandatory.' There are three input fields: 'Username', 'Password', and 'Confirm Password'. Below the 'Password' field is a blue button labeled 'Reset Password'.

Εικόνα 4.1.4.7 : Password Reset (4)

Όσον αφορά τη διαδικασία του forgot username, όταν πατήσουμε στο link από την σελίδα login εμφανίζεται η σελίδα που βλέπουμε στην εικόνα 4.1.4.8.

The image shows a 'Send Username Reset Email' form. The title is 'Send Username Reset Email' with the subtitle 'Provide your email below.' There is one input field labeled 'Email'. Below it is a blue button with a checkmark icon and the text 'Send Username Reset Email'.

Εικόνα 4.1.4.8 : Username Reset (1)

Αφού συμπληρώσουμε το email μας και πατήσουμε “Send Username Reset Email” το σύστημα μας ενημερώνει πως μας έστειλε ένα link στο email μας για να ανανεώσουμε τον username μας. Στην εικόνα 4.1.4.9 βλέπουμε το email που ερχεται στο λογαριασμό μας. Επίσης στην εικόνα 4.1.4.10 βλέπουμε την σελίδα που αν συμπληρώσουμε τα απαραίτητα στοιχεία ανανεώνεται το username μας.

Incident Reporting System

Hello User,

To complete change your username, please follow these 2 simple steps:

1. Copy the unique token below:

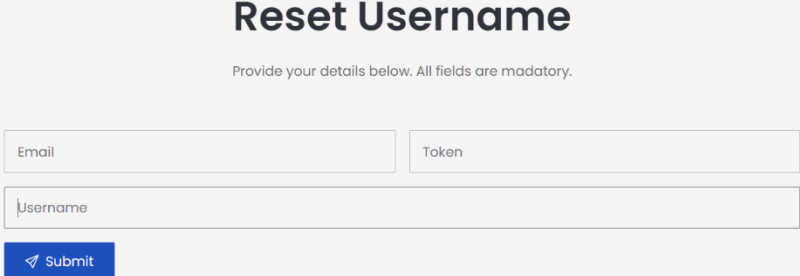
Token: htN0GvDMRd8ioY3

This token will expire in 60 minutes.

2. Visit the url by clicking on the link below and fill the necessary fields.

<https://10.16.20.149/irs/changeUsername>

Sent from the site: Incident Reporting System



Εικόνα 4.1.4.9 : Username Reset (2)

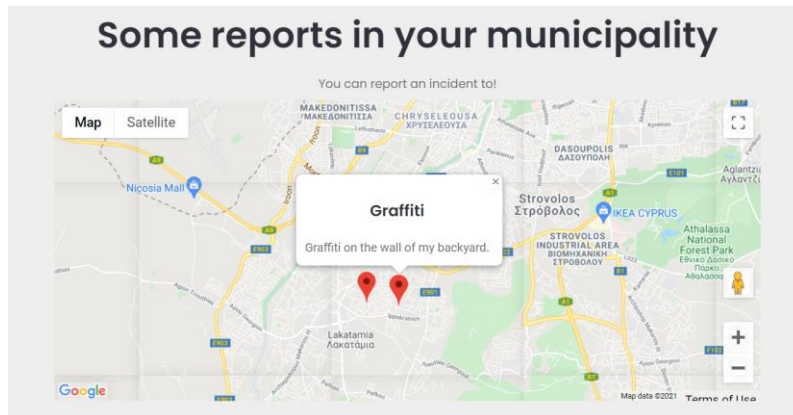
Εικόνα 4.1.4.10 : Username Reset (3)

4.2 Simple User - Πολίτης

Μόλις ο επισκέπτης δημιουργήσει λογαριασμό και κάνει login, πλέον θεωρείται σαν Simple User που έχει κάποιες επιπλέον δυνατότητες. Οι λειτουργίες που περιεγράφηκαν στο υποκεφάλαιο 4.1 για τον επισκέπτη της σελίδας ισχύουν και για τον Simple User. Οι επιπρόσθετες λειτουργίες παρουσιαστούν στα επόμενα υποκεφάλαια που ακολουθούν.

4.2.1 Αρχική Σελίδα

Η αρχική σελίδα παραμένει όπως και στον επισκέπτη, όμως έχει σαν επιπλέον στοιχείο τον χάρτη που φαίνεται στην εικόνα 4.2.1. Ο χάρτης αυτό παρουσιάζει 20 τυχαία solved reports (ή λιγότερα εάν δεν υπάρχουν ακόμα 20 reports) στο δήμο του χρήστη. Μόλις πατηθεί ένα από τα σημεία στο χάρτη φαίνονται οι πληροφορίες του συγκεκριμένου report: ο τύπος και η μικρή περιγραφή που έδωσε ο χρήστης. Με αυτόν τον τρόπο οι χρήστες βλέπουν τα reports των συμπολιτών τους και ωθούνται στο να αναφέρουν και αυτοί κάποιο πρόβλημα που έχουν εντοπίσει.



Εικόνα 4.2.1.1 : Χάρτης Αρχικής Σελίδας

Επιπλέον, στην αρχική σελίδα το menu έχει αλλάξει. όπως φαίνεται στην εικόνα 4.2.1.2 και έχουν προστεθεί οι επιλογές για Report A Problem αλλά και το εικονίδιο του User για να μπορέσει ο χρήστης να δει και να αλλάξει τα στοιχεία του.



Εικόνα 4.2.1.2 : Simple User Menu

4.2.2 Report A Problem

Για να αναφέρει ο χρήστης ένα πρόβλημα θα ακολουθήσει τα τρία βήματα της φόρμας αναφοράς περιστατικών. Αρχικά, επιλέγει τον τύπο του περιστατικού και γράφει μία μικρή περιγραφή. Ο τύπος μπορεί να είναι ένα από τα παρακάτω:

- Abandoned cars or vehicles
- Abandoned trolleys
- Animal related issues
- Communication equipment pits and lids
- Dumped rubbish and litter
- Graffiti
- Noise
- Parking
- Pavements and roads

- Playground equipment
- Street lights
- Street cleaning
- Trees
- Water and sewer issues
- Other

Το 1^ο βήμα για αναφορά περιστατικών φαίνεται στην εικόνα 4.2.2.1. Αν ο χρήστης δεν συμπληρώσει τα στοιχεία αυτά τα απαραίτητα μηνύματα λάθους εμφανίζονται και το σύστημα δεν επιτρέπει τη συνέχεια στο επόμενο βήμα

The screenshot shows a form titled "Provide the details" with a subtitle "Provide the details of the incident below. All fields are mandatory." Below the title is a progress bar with three steps: 1 (Provide the details), 2 (Upload photos), and 3 (Choose a location). Step 1 is currently active. The form contains two input fields: "Type of Incident" and "Write a small description about the incident." A "NEXT" button is located at the bottom right of the form.

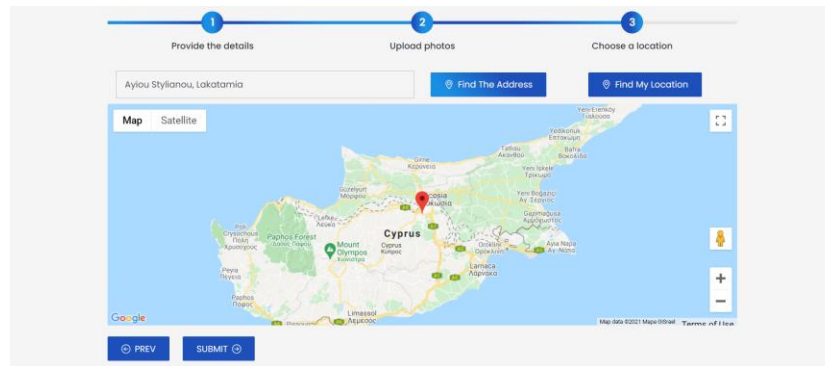
Εικόνα 4.2.2.1 : Report A Problem (1)

Προχωρώντας στο επόμενο βήμα, που φαίνεται στη εικόνα 4.2.2.2., ο χρήστης καλείται να αναρτήσει από μία μέχρι τρεις φωτογραφίες του περιστατικού. Εάν δεν τοποθετήσει καμία φωτογραφία, το σύστημα δεν επιτρέπει τη συνέχεια στο τελευταίο βήμα και τον ενημερώνει κατάλληλα.

The screenshot shows a form titled "Upload photos" with a subtitle "You can upload up to 3 pictures of the same incident." Below the title is a progress bar with three steps: 1 (Provide the details), 2 (Upload photos), and 3 (Choose a location). Step 2 is currently active. The form contains three photo upload slots. The first slot has a photo of a graffiti wall. The second and third slots are empty. A "PREV" button is located at the bottom left, and a "NEXT" button is located at the bottom right.

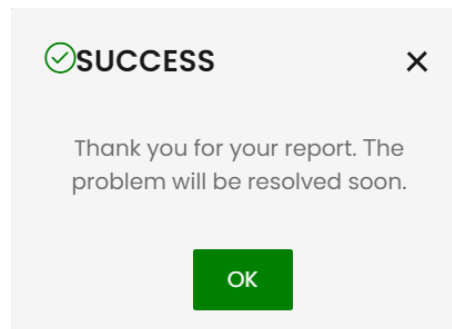
Εικόνα 4.2.2.2 : Report A Problem (2)

Τέλος, ο χρήστης πρέπει να επιλέξει το σημείο στο χάρτη, όπου έχει γίνει το περιστατικό. Δίνονται οι επιλογές να πατήσει το κουμπί “Find My Location” όπου ο χρήστης επιλέγει εάν βρίσκεται στο σημείο του περιστατικού εκείνη την στιγμή και επίσης η επιλογή για να γράψουν τη διεύθυνση και να πατήσουν “Find The Address”, εάν έχουν απομακρυνθεί από το σημείο το περιστατικού. Παράλληλα, μπορούν να ελέγξουν το σημείο που έχει τοποθετηθεί στο χάρτη αλλά και να το μετακινήσουν εάν δεν έχει τοποθετηθεί σωστά.



Εικόνα 4.2.2.3 : Report A Problem (3)

Εάν ο χρήστης τοποθετήσει ένα σημείο το οποίο είναι εκτός των γεωγραφικών ορίων των δήμων, τότε η φόρμα δεν υποβάλλεται και ενημερώνεται ο χρήστης. Εάν όλα έχουν συμπληρωθεί ορθά εμφανίζεται ένα μήνυμα επιτυχίας, όπως φαίνεται και στην εικόνα 4.2.2.4.



Εικόνα 4.2.2.4 : Report A Problem (4)

4.2.3 User's Details

Ο χρήστης για να δει τα στοιχεία του επιλέγει από το menu το εικονίδιο με τον User.  Ακολούθως, ο χρήστης οδηγείται στη σελίδα που φαίνεται στην εικόνα 4.2.3 και μπορεί να δει τα

στοιχεία του στην κάρτα αριστερά αλλά και να αλλάξει αυτά που προσφέρονται στη φόρμα δεξιά. Υπάρχει ακόμα και το link για αλλαγή κωδικού. Επιπλέον, του δίνεται η δυνατότητα να δει το Report History του πατώντας το κουμπί “Report History” στην κάρτα αλλά και η δυνατότητα να διαγράψει το λογαριασμό του και μαζί όλα τα reports που έχει αναφέρει.

User Details

This is your data. You can change your details below.

Chryso Makri
cmakri02

Credit Score: 90%

Id: 1
Email: chrysomakri13@gmail.com
Phone: 97752375
Municipality: Λακατάμεια
Type: Simple
Reported times: 0
Blocked: No
Signed up at: 2021-02-12 16:45:27
Updated at: 2021-04-05 01:43:22

[Report History](#)

Edit Details

Email and phone fields are mandatory. Change them or leave them as it is.

If you want to change your municipality select city & municipality, otherwise, leave it empty or choose your current municipality.

chrysomakri13@gmail.com
97752375
City ▼
Municipality ▼

[EDIT DETAILS](#)

[Change your password.](#)

[DELETE ACCOUNT](#)

Εικόνα 4.2.3 : User's Details

Μία επιπλέον πληροφορία για τα στοιχεία του χρήστη και συγκεκριμένα για το credit score, είναι ότι το credit score δημιουργείται από το μέσο όρο των scores των solved reports. Κάθε solved report παίρνει ένα score από το ένα μέχρι το πέντε, από τους υπεύθυνους του δήμου και ο μέσος όρος αυτών των reports του χρήστη αποτελεί το credit score. Εάν υπάρχουν solved reports ακόμα για τον χρήστη τότε το score δεν εμφανίζεται.

4.2.4 Report History

Το Report History του χρήστη παρουσιάζεται με τη μορφή πίνακα, όπως φαίνεται και στην εικόνα 4.2.4.1. Ο χρήστης μπορεί να επιλέξει ποιες αναφορές θα δει ανάλογα με το status τους (All/Unseen/Seen...). Μπορεί να δει μερικές βασικές πληροφορίες όπως το id, το date & time, τον

τύπο, το status, αλλά και να πατήσει πάνω στο κουμπί με το τόξο στη στήλη more για να δει όλες τις πληροφορίες ενός report του.

Report History				
All Unseen Seen In Progress Solved Blocked				
Id	Date & Time	Type	Status	More
42	2021-05-07 00:32:40	Street cleaning	Unseen	⊖
40	2021-05-05 20:39:59	Graffiti	Unseen	⊖
31	2021-04-05 15:07:09	Abandoned trolleys	Solved	⊖
16	2021-03-12 11:17:14	Trees	Unseen	⊖
10	2021-02-19 16:38:39	Graffiti	Solved	⊖

Εικόνα 4.2.4.1 : Report History

Όταν ο χρήστης βλέπει ένα συγκεκριμένο report μπορεί να δει όλες τις πληροφορίες, όπως αυτές φαίνοντα στις εικόνες 4.2.4.2, 4.2.4.3 και 4.2.4.4. Μία επιπλέον πληροφορία που εμφανίζεται εάν το report έχει γίνει solved είναι το score που φαίνεται με τα πέντε αστέρια. Ο υπεύθυνος που έχει δει το report, αφού έχει επιλυθεί, δίνει ένα score από το ένα μέχρι το πέντε, που αντιστοιχεί στο πόσο έγκυρη ήταν μία αναφορά και αν άξιζε να αναφερθεί. Με αυτόν τον τρόπο οι πολίτες ωθούνται να αναρτούν άξια προς αναφορά προβλήματα και τις σωστές πληροφορίες για αναφορά.

Report 10

Description:
Graffiti on the wall of my backyard.

Status:
Solved

By Chryso Makri

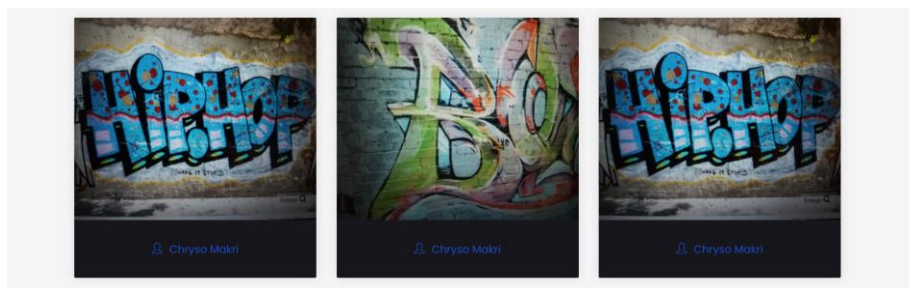
Posted on 2021-02-19 16:38:39
Updated on 2021-04-14 18:54:22

Λακατάμεια

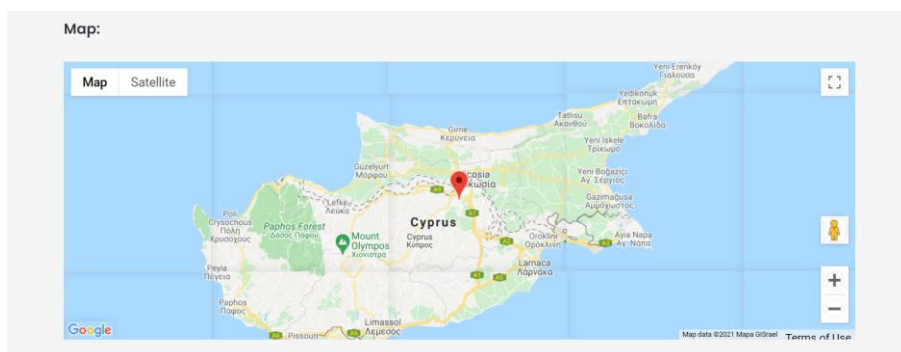
35.1149013, 33.326156100000006

★★★★★

Εικόνα 4.2.4.2 : Single Report (1)



Εικόνα 4.2.4.3 : Single Report (2)



Εικόνα 4.2.4.4 : Single Report (3s)

4.3 Operator – Υπάλληλος Δήμου

Ο Operator είναι ο Υπάλληλος του Δήμου που θα χρησιμοποιεί το σύστημα και σαν απλός χρήστης, αφού έχει όλες τις προηγούμενες επιλογές που αναφέρθηκαν πριν, αλλά και σαν ο υπεύθυνος που θα διαχειρίζεται τους χρήστες και τα reports μόνο του δικού του δήμου. Οι διάφορες αυτές λειτουργίες με τις οποίες γίνεται η διαχείριση θα αναλυθούν σε αυτό το υποκεφάλαιο.

4.3.1 Management

Μόλις ένας Operator κάνει login στο σύστημα εμφανίζεται στο menu η επιλογή του Management όπως φαίνεται και στην εικόνα 4.3.1.1. Η επιλογή αυτή αποτελείται από 2 links, ένα για τα reports και ένα για για τους Users. Το συγκεκριμένο menu είναι το ίδιο και για τους διαχειριστές της σελίδας.

Εικόνα 4.3.1 : Operator & Administrator Menu

4.3.1.1 Reports

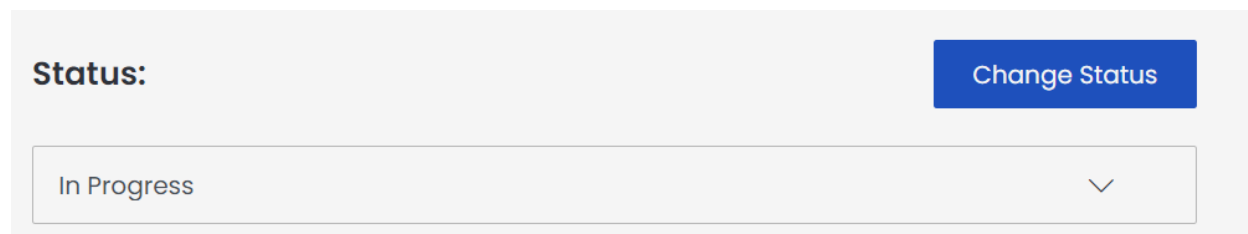
Οι υπεύθυνοι του δήμου είναι οι αρμόδιοι για να επιλύουν τα διάφορα προβλήματα που αναρτούν οι χρήστες στο σύστημα. Μπορούν να δουν τα διάφορα reports που αφορούν μόνο τον δικό τους δήμο, όπως φαίνεται στην εικόνα 4.3.1.1.1 και να επιλέξουν στο status ποιες αναφορές θα δουν (All, Unseen, Seen, ...). Επίσης, εκτός από τις βασικές πληροφορίες που φαίνονται στον πίνακα των reports, μπορούν να πατήσουν το κουμπί στην στήλη more, για να δουν όλα τα δεδομένα που αφορούν ένα συγκεκριμένο report.

Reports				
All Unseen Seen In Progress Solved Blocked				
Id	Date & Time	Type	Status	More
42	2021-05-07 00:32:40	Street cleaning	Unseen	⌵
37	2021-04-23 15:02:27	Dumped rubbish and litter	Unseen	⌵
31	2021-04-05 15:07:09	Abandoned trolleys	Solved	⌵
10	2021-02-19 16:38:39	Graffiti	Solved	⌵

Εικόνα 4.3.1.1.1 : Reports Operator & Administrator

Ο Operator έχει τη δυνατότητα να δει όλες τις πληροφορίες ενός report με τον τρόπο που τις βλέπει και ο Simple User, όπως φαίνονται στις εικόνες 4.2.4.2, 4.2.4.3 και 4.2.4.4. Η μόνη διαφορά είναι ότι μπορεί να αλλάξει το status επιλέγοντας μία από τις επιλογές: Unseen, Seen, In Progress,

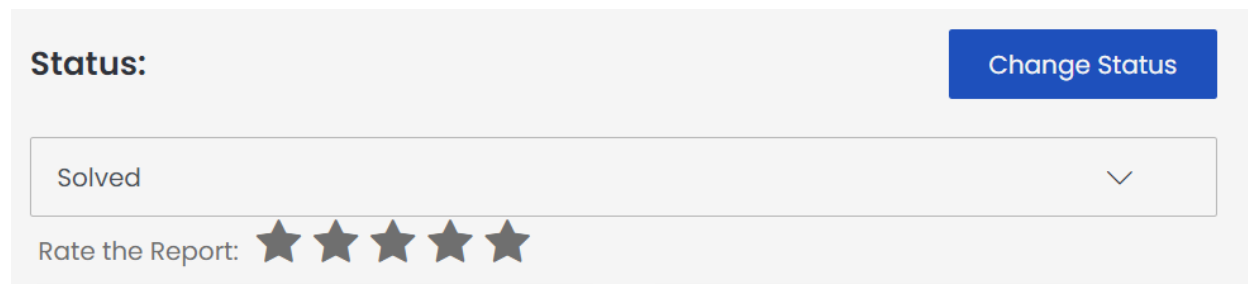
Solved ή Blocked και πατώντας το κουμπί “Change Status”. Η εικόνα 4.3.1.1.2 που ακολουθεί μας δείχνει αυτήν την επιπλέον δυνατότητα.



The screenshot shows a user interface element with the label "Status:" on the left. To its right is a blue button labeled "Change Status". Below the label is a dropdown menu that currently displays "In Progress" and has a downward arrow icon on its right side.

Εικόνα 4.3.1.1.2 : Single Report Operator & Administrator (1)

Επίσης, αν ο υπεύθυνος του δήμου επιλέξει το solved, δηλαδή εάν το πρόβλημα έχει πλέον επιλυθεί, τότε εμφανίζονται κάτω από το Status πέντε αστέρια, ώστε να βαθμολογήσει κατά πόσο το report ήταν έγκυρο και άξιο αναφοράς. Αυτό φαίνεται στην εικόνα 4.3.1.1.3. Χρειάζεται πρώτα να βαθμολογήσει και στη συνέχεια να πατήσει “Change Status”, για να καταχωρηθεί η αλλαγή. Εάν ο υπεύθυνος ορίσει την αναφορά ως blocked, τότε το score που μπαίνει για την αναφορά είναι αυτομάτως 0. Παράλληλα, εάν 3 διαφορετικά report του ίδιου χρήστη γίνουν blocked τότε γίνεται και ο ίδιος ο χρήστης blocked και ενώ μπορεί να κάνει login και να δει τα στοιχεία του, δεν του επιτρέπεται να αναφέρει πλέον περιστατικά.



The screenshot shows a user interface element with the label "Status:" on the left. To its right is a blue button labeled "Change Status". Below the label is a dropdown menu that currently displays "Solved" and has a downward arrow icon on its right side. Below the dropdown menu, there is a text label "Rate the Report:" followed by five gray star icons.

Εικόνα 4.3.1.1.3: Single Report Operator & Administrator (2)

4.3.1.2 Users

Εκτός από τις αναφορές, οι υπεύθυνοι των δήμων μπορούν να διαχειριστούν και τους πολίτες του δήμου τους. Οι χρήστες εμφανίζονται σε μορφή πίνακα όπως και οι αναφορές. Πάνω από τον πίνακα εμφανίζονται οι 3 τύποι χρηστών, ώστε να επιλέξουν ποια στοιχεία χρηστών θέλουν να

δουν (All, Simple, Operator, Administrator). Στον πίνακα φαίνονται κάποιες γενικές πληροφορίες, το id, το ονοματεπώνυμο, το username και ο τύπος του χρήστη. Μπορούν, επίσης, να δουν όλες τις πληροφορίες που αφορούν ένα χρήστη, πατώντας το κουμπί στη στήλη More. Οι πληροφορίες αυτές φαίνονται στην εικόνα 4.3.1.2.1.

Users

AllSimpleOperatorAdministrator

Id	Name & Surname	Username	Type	More
1	Chryso Makri	cmakri02	Simple	⊖
3	Chryso Makri	cmakri	Operator	⊕
4	Chryso Makri	admin	Administrator	⊕
20	Μαρία Μακρή	makri.maria	Simple	⊖
25	andrikkos makropoulos	makropoulos	Simple	⊖

Εικόνα 4.3.1.2.1: Users

Μόλις επιλέξει να δει όλες τις πληροφορίες ενός χρήστη μεταφέρεται στη σελίδα που φαίνεται στην εικόνα 4.3.1.2.2. Εκεί, ο υπεύθυνος του δήμου μπορεί να δει τις πληροφορίες στην κάρτα αριστερά, αλλά και να αλλάξει τις πληροφορίες του χρήστη, εφόσον ο ίδιος ο χρήστης το ζητήσει και γίνει η απαραίτητη επιβεβαίωση. Ακόμη δίνεται και η δυνατότητα για διαγραφή του χρήστη. Εάν ο χρήστης δεν είναι ένας simple user, αλλά ανήκει είτε στην κατηγορία των Operators ή των Administrators τότε δεν δίνεται η δυνατότητα για αλλαγή των στοιχείων του χρήστη ή για διαγραφή του. Στην περίπτωση αυτή φαίνεται μόνο η κάρτα με τα στοιχεία, σε περίπτωση που χρειαστεί ο υπεύθυνος του δήμου να επικοινωνήσει μαζί τους.

User Details

This is the data of user 1.



Chryso Makri

cmakri02

 **Credit Score: 100%**

Id: 1

Email: chrysomakril3@gmail.com

Phone: 97752375

Municipality: Λακατάμεια

Type: Simple

Reported times: 0

Blocked: No

Signed up at: 2021-02-12 16:45:27

Updated at: 2021-04-05 01:43:22

Edit Details

Fields with * are mandatory. Change them or leave them as it is.
City, Municipality & Password can be left blank if you don't want to change it.

Name: *

Chryso

Surname: *

Makri

Email: *

chrysomakril3@gmail.com

Phone: *

97752375

Reported Times: *

0

City

▼

Municipality

▼

Type

▼

Username: *

cmakri02

Password:

Password

Confirm Password:

Confirm Password

 EDIT DETAILS

 DELETE ACCOUNT

Εικόνα 4.3.1.2.2: User Details

4.4 Administrator – Διαχειριστής Συστήματος

Ο Administrator είναι ο γενικός διαχειριστής του συστήματος, του οποίου οι λειτουργίες μοιάζουν αρκετά με τις λειτουργίες του υπάλληλου του δήμου, με κάποιες μικρές διαφορές. Όπως και ο Operator, έχει όλες τις λειτουργίες του απλού χρήστη αλλά μπορεί και να διαχειρίζεται τα reports και τους χρήστες, όχι όμως μόνο του δήμου του αλλά όλους που είναι εγγεγραμμένοι στο σύστημα. Οι λειτουργίες που έχει ο Administrator παραμένουν ακριβώς οι ίδιες στη χρήση, μόνο που διαχειρίζεται περισσότερα άτομα και αναφορές. Η μόνη μικρή διαφορά είναι στην αλλαγή των στοιχείων ενός χρήστη. Έχει την ιδιότητα να μετατρέψει έναν απλό χρήστη ή έναν υπάλληλο του

δήμου σε διαχειριστή, κάτι που οι προηγούμενοι τύποι χρηστών δεν είχαν την εξουσιοδότηση να κάνουν. Τέλος, μπορούν να αλλάξουν τα στοιχεία ή να διαγράψουν το λογαριασμό και σε χρήστες που είναι Operators και Administrators.

Κεφάλαιο 5

Αξιολόγηση συστήματος

2.1 User Experience Questionnaire	49
2.2 Αποτελέσματα Ερωτηματολογίου	52

5.1 User Experience Questionnaire [32]

Με την ολοκλήρωση της εφαρμογής, χρησιμοποίησα το ερωτηματολόγιο User Experience Questionnaire (UEQ), για να αξιολογήσω το σύστημα μου όσον αφορά την εμπειρία χρήστη. Το συγκεκριμένο ερωτηματολόγιο δημιουργήθηκε αρχικά σαν πρώτη έκδοση στη γερμανική γλώσσα το 2005. Σήμερα, έχει ήδη μεταφραστεί σε περισσότερες από 30 γλώσσες και υπάρχει και η ελληνική έκδοση, την οποία και χρησιμοποιώ.

Το συγκεκριμένο ερωτηματολόγιο μετρά την εμπειρία χρήστη που έχει κάποιος όταν χρησιμοποιεί ένα σύστημα. Για να υπολογιστεί πόσο εύχρηστο είναι ένα διαδραστικό προϊόν μετριούνται οι διάφορες πτυχές της κλασικής χρηστικότητας, που είναι η αποτελεσματικότητα, η ευκρίνεια και η αξιοπιστία, αλλά και οι πτυχές τις εμπειρίας χρήστη, δηλαδή η ικανοποίηση και η καινοτομία. Συλλέγονται ορισμένα ποσοτικά δεδομένα σχετικά με την εντύπωση που δημιουργήσαν οι χρήστες για το σύστημα και στη συνέχεια αυτά αναλύονται.

Είναι ένα αρκετά γρήγορο ερωτηματολόγιο, που ο χρήστης θα συμπληρώσει μέσα σε 2 λεπτά ή λιγότερο και αποτελεί μία αξιόπιστη μεθοδολογία για αξιολόγηση διαφόρων συστημάτων και προϊόντων. Το UEQ χρησιμοποιεί μία συγκεκριμένη κλίμακα, που θα επεξηγηθεί στην επόμενη παράγραφο, για συλλογή των ποσοτικών δεδομένων από τους συμμετέχοντες στο ερωτηματολόγιο. Η κλίμακα αυτή αλλά και τα στοιχεία που ελέγχει το ερωτηματολόγιο έχουν επιλεγεί προσεκτικά μετά από έρευνες. Έγιναν 11 δοκιμές χρηστικότητας με τον συνολικό αριθμό συμμετεχόντων να είναι 144 άτομα και ακόμα μία διαδικτυακή έρευνα με 722 συμμετέχοντες για να ελέγξουν την συνοχή της κλίμακας που χρησιμοποιεί το UEQ και των

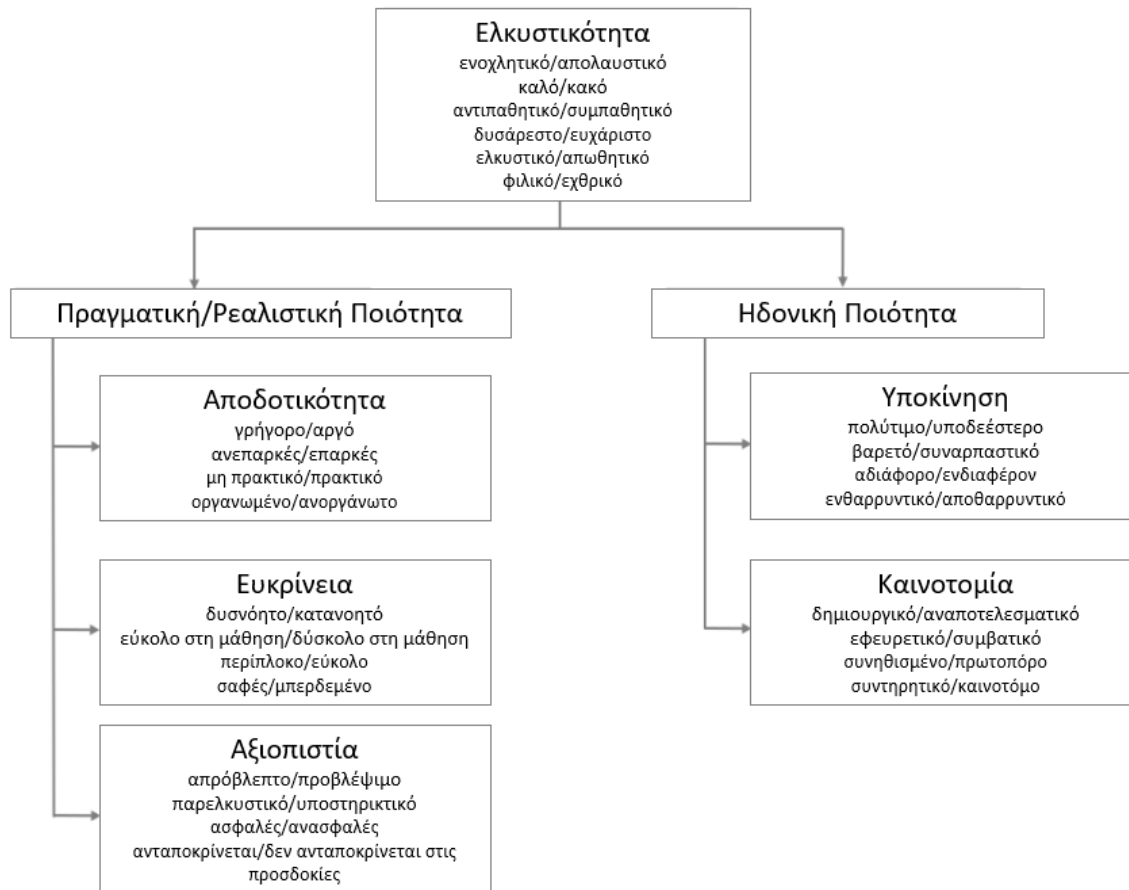
στοιχείων που συλλέγει. Τα αποτελέσματα έδειξαν υψηλή συνοχή, εγκυρότητα και καλά αποτελέσματα για την επιλογή των στοιχείων και της κλίμακας που χρησιμοποιούνται. Παράλληλα, είναι εύκολο στη χρήση λόγω του συμπληρωματικού υλικού και της ποικιλίας των γλωσσών στις οποίες έχει μεταφραστεί, οπότε χρησιμοποιήθηκε σε αρκετές παρόμοιες μελέτες. Για όλους αυτούς τους προαναφερθέντες λόγους έχει επιλεγεί για την αξιολόγηση και του δικού μου συστήματός.

Στο συγκεκριμένο ερωτηματολόγιο χρησιμοποιείται η κλίμακα των επτά σταδίων και τα στοιχεία στην κλίμακα αντιπροσωπεύονται από 2 όρους, το θετικό και τον αρνητικό. Για παράδειγμα, *οργανωμένο ο ο ο ο ο ο ο ανοργάνωτο*. Τα στοιχεία κλιμακώνονται από -3 έως +3. Το -3 αντιπροσωπεύει την πιο αρνητική απάντηση, 0 την ουδέτερη και +3 την πιο θετική.

Το UEQ περιέχει 6 κλίμακες με 26 στοιχεία, τα οποία φαίνονται αναλυτικά, στην εικόνα 5.1. Ουσιαστικά οι κλίμακες αυτές είναι οι εξής:

- Ελκυστικότητα: Αποτελεί την συνολική εντύπωση που δημιουργούν οι χρήστες για το προϊόν (αν το συμπαθούν ή όχι).
- Ευκρίνεια: Εξετάζει κατά πόσο το προϊόν είναι εύκολο στην εκμάθηση (εάν είναι εύκολο κάποιος να εξοικειωθεί μαζί του).
- Αποδοτικότητα: Αξιολογεί εάν οι χρήστες μπορούν να εκτελέσουν την εργασία που κάνουν σε λογικά χρονικά πλαίσια, χωρίς περιττή προσπάθεια.
- Αξιοπιστία: Ελέγχει εάν ο χρήστης έχει τον έλεγχο του συστήματος και μπορεί να καθορίσει το αποτέλεσμα της μελλοντικής του ενέργειας από τη γνώση που έχει λάβει μέχρι τώρα για το σύστημα. Επίσης, καθορίζει εάν το σύστημα είναι ασφαλές ή όχι.
- Υποκίνηση: Εξετάζει πόσο ενθαρρυντικό είναι ένα προϊόν.
- Καινοτομία: Αποτελεί το πόσο δημιουργικό και καινοτόμο είναι ένα σύστημα.

Η ελκυστικότητα αποτελεί μία ξεχωριστή κατηγορία η οποία χωρίζεται στην πραγματική και στην ηδονική ποιότητα. Η πραγματική ποιότητα αποτελείται από τις κλίμακες ευκρίνεια, αποδοτικότητα και αξιοπιστία και η ηδονική ποιότητα από τις κλίμακες υποκίνηση και καινοτομία.



Εικόνα 5.1.1: UEQ Evaluation Scales & Elements

Η σειρά των θετικών και αρνητικών όρων μπαίνει με τυχαία σειρά κάθε φορά. Οι μισές ερωτήσεις ξεκινούν με τον αρνητικό και οι άλλες μισές με τον θετικό όρο. Αυτό συμβαίνει για να γίνει ο έλεγχος κατά πόσο ο συμμετέχων στο σύστημα είναι συγκεντρωμένος καθώς απαντά τις ερωτήσεις. Για να μειωθούν τα άτομα που δεν απαντούν σοβαρά το ερωτηματολόγιο, το UEQ υιοθέτησε την τακτική με την τυχαιοποιημένη σειρά των όρων και επίσης γίνεται έλεγχος κατά πόσο διαφέρουν οι απαντήσεις που μετρούν περίπου την ίδια πτυχή ποιότητας. Για παράδειγμα, στην εικόνα 5.1.2 οι απαντήσεις στα στοιχεία της κλίμακας Ευκρίνεια δεν είναι αρκετά συνεπείς, αφού ο χρήστης έδωσε σαν απάντηση 6 για confusing και 5 για easy. Η απόσταση είναι αρκετά μεγάλη και επομένως η κλίμακα Ευκρίνεια θεωρείται ανακριβής. Εάν αυτό συμβαίνει για μία και μόνο κλίμακα τότε οι απαντήσεις του συγκεκριμένου συμμετέχοντα δεν αποκλείονται, Αν όμως αυτό συμβαίνει για 2 ή περισσότερες κλίμακες οι απαντήσεις του συμμετέχοντα αποκλείονται.

not understandable	o o o o o x o	understandable
easy to learn	o o o o o o x	difficult to learn
complicated	o o o o x o o	easy
clear	o o o o o x o	confusing

Εικόνα 5.1.2: Παράδειγμα Ανακριβής Κλίμακας

Το ερωτηματολόγιο στάλθηκε αμέσως μετά την ολοκλήρωση του συστήματος, κυρίως σε φοιτητές και καθηγητές του τμήματος Πληροφορικής ,που έχουν πρόσβαση να δουν το σύστημα. Το ερωτηματολόγιο δημιουργήθηκε με τη βοήθεια των Google Forms, βάση του UEQ και φωτογραφίες με τις ερωτήσεις φαίνονται στο Παράρτημα Γ. Επίσης, βρίσκεται στο σύνδεσμο https://docs.google.com/forms/d/1qYqGDp3dhfYjicY4ZpEz4oscQJuEsgx_FV0eZOyEYzw. Τα αποτελέσματα θα αναλυθούν στο επόμενο υποκεφάλαιο 5.2.

5.2 Αποτελέσματα Ερωτηματολογίου

Το ερωτηματολόγιο απαντήθηκε συνολικά από 33 άτομα. Καταρχήν, έπρεπε να ελεγχθεί αν οι συμμετέχοντες στο ερωτηματολόγιο απάντησαν με σοβαρότητα. Αυτό ελέγχθηκε με τον τρόπο που αναφέρθηκε και προηγουμένως στο υποκεφάλαιο 5.1. Σε περίπτωση που ο χρήστης απάντησε με ασυνέπεια και αυτό επηρεάζει περισσότερες από 3 κλίμακες, τότε διαγράφονται οι απαντήσεις του χρήστη αυτού από το σύνολο απαντήσεων.

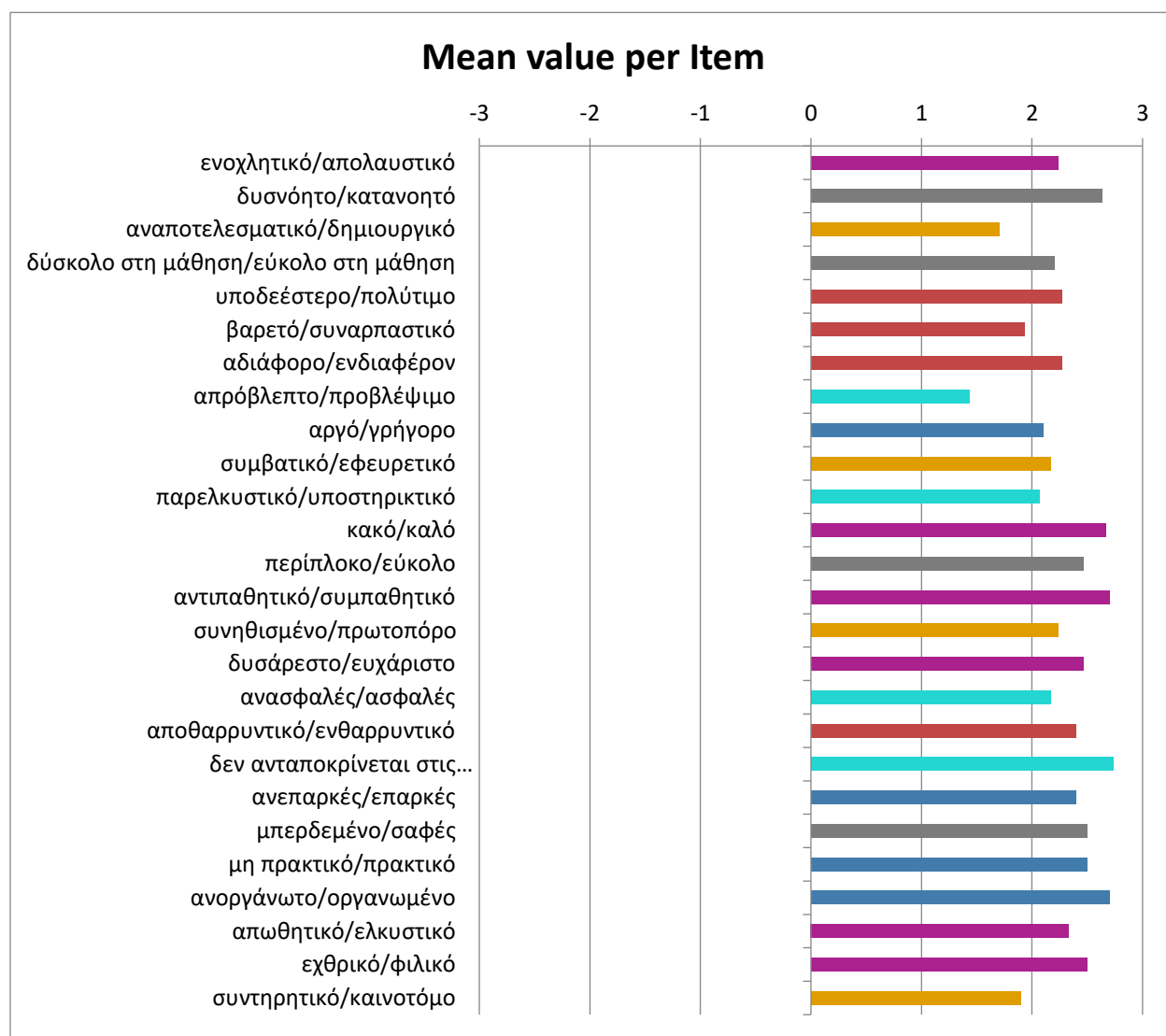
Στον πίνακα που φαίνεται στην εικόνα 5.2.1 βλέπουμε για κάθε χρήστη σε ποιες κλίμακες υπάρχει ασυνέπεια στις απαντήσεις του. Τα κελιά με πορτοκαλί, που περιέχουν τον αριθμό 1 στις διάφορες κλίμακες, σημειώνουν ότι στην συγκεκριμένη κλίμακα ο συμμετέχοντας δεν απάντησε με συνέπεια στις ερωτήσεις που την αφορούν. Αντιθέτως, τα άσπρα κελιά, που είναι κενά, σημαίνει πως με την συγκεκριμένη κλίμακα και τις απαντήσεις που έδωσε ο χρήστης όλα είναι εντάξει. Η στήλη “Critical?” παρουσιάζει το άθροισμα των κλιμάκων που ο χρήστης δεν απάντησε με συνέπεια. Αν αυτός ο αριθμός είναι 3 και πάνω (φαίνεται με κόκκινο), οι απαντήσεις του χρήστη διαγράφονται. Όπως παρατηρούμε στον πίνακα έχουμε 3 περιπτώσεις χρηστών που το πεδίο

“Critical?” περιέχει τον αριθμό 3. Επομένως, οι 3 αυτές εγγραφές στο πίνακα απαντήσεων των συμμετεχόντων διαγράφηκαν, για μεγαλύτερη ακρίβεια των αποτελεσμάτων. Επίσης, υπάρχει και μία περίπτωση με 2 ασυνεπείς κλίμακες, όμως αυτή θεωρείται αποτέλεσμα απροσεξίας και δεν θα επηρεάσει ιδιαίτερα τα αποτελέσματα, άρα δεν διαγράφεται.

Scales with inconsistent answers						
Ελκυστικότητα	Ευκρίνεια	Αποδοτικότητα	Αξιοπιστία	Διέγερση	Καινοτομία	Critical?
						0
						0
						0
						0
		1				1
			1	1	1	3
	1			1	1	3
						0
						0
					1	1
			1			1
					1	1
						0
	1					1
	1					1
						0
						0
						0
					1	1
						0
						0
	1			1	1	3
						0
						0
						0
						0
						0
						0
						0
						0
						0
					1	1
1			1			2
			1			1

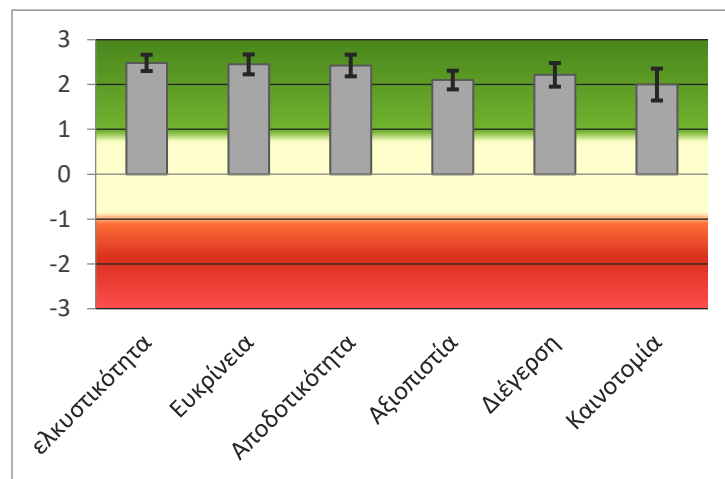
Εικόνα 5.2.1: Πίνακας Εύρεσης Ασυνέπειας

Εφόσον διαγράψαμε τις εγγραφές που αλλοίωναν τα αποτελέσματα, πλέον μπορούμε να δούμε τις παραγόμενες γραφικές. Η πρώτη γραφική παράσταση, που φαίνεται στην εικόνα 5.2.2 δείχνει την μέση τιμή για το κάθε στοιχείο. Οι τιμές κυμαίνονται από -3, που αντιστοιχεί στην λιγότερη καλή αξιολόγηση, μέχρι 3 που αντιστοιχεί στη καλύτερη αξιολόγηση. Με 0 θεωρείται η ουδέτερη αξιολόγηση. Γενικότερα, βλέπουμε πως όλα τα στοιχεία αξιολογήθηκαν με τιμή μεγαλύτερη από 1 και αρκετά από αυτά έχουν τιμή μεγαλύτερη από 2. Αυτό σημαίνει πως γενικότερα το σύστημα αξιολογήθηκε θετικά από τους συμμετέχοντες για όλα τα στοιχεία.



Εικόνα 5.2.2: Γραφική Παράσταση Μέσου Όρου των Στοιχείων

Η γραφική παράσταση που ακολουθεί στην εικόνα 5.2.3 μας δείχνει τη γραφική παράσταση των 6 κλιμάκων που δημιουργήθηκε από το μέσο όρο των στοιχείων που αναλογούν στην κάθε μία. Ο χώρος στη γραφική παράσταση χωρίζεται σε πράσινο για τα καλά αποτελέσματα, κίτρινο για τα ουδέτερα ή λιγότερα καλά και με κόκκινο για τα καθόλου καλά αποτελέσματα. Με γκρι χρώμα φαίνεται η μέση τιμή για την κάθε κλίμακα και με τη μαύρη γραμμή φαίνεται η τυπική απόκλιση. Και οι 6 κλίμακες καταμέτρησης της ευχρηστίας έχουν μέσο όρο 2 και πάνω, βρίσκονται στο πράσινο τμήμα, κάτι που καθορίζει την εφαρμογή ως ένα πολύ εύχρηστο σύστημα. Η τυπική απόκλιση είναι αρκετά μικρή και στις 6 κλίμακες, ελάχιστα μεγαλύτερη στην κλίμακα της καινοτομίας. Αυτό καθορίζει πως δεν είχαμε μεγάλη διαφορά στις απαντήσεις των συμμετεχόντων και αυτό δείχνει πως τα δεδομένα του ερωτηματολογίου είναι αξιόπιστα. Την πιο μικρή μέση τιμή έχει η κλίμακα της καινοτομίας, αν και υπάρχει περισσότερη απόκλιση στις απαντήσεις των χρηστών από τις υπόλοιπες κλίμακες. Αυτό ίσως να δείχνει πως το προϊόν δεν είναι υπερβολικά καινοτόμο, όμως εφόσον βρίσκεται ακριβώς στην τιμή 2 δεν θεωρούνται ανησυχητικά τα αποτελέσματα. Πιο αναλυτικά μπορούμε να δούμε τη μέση τιμή και την τυπική απόκλιση στον πίνακα της εικόνας 5.2.4.

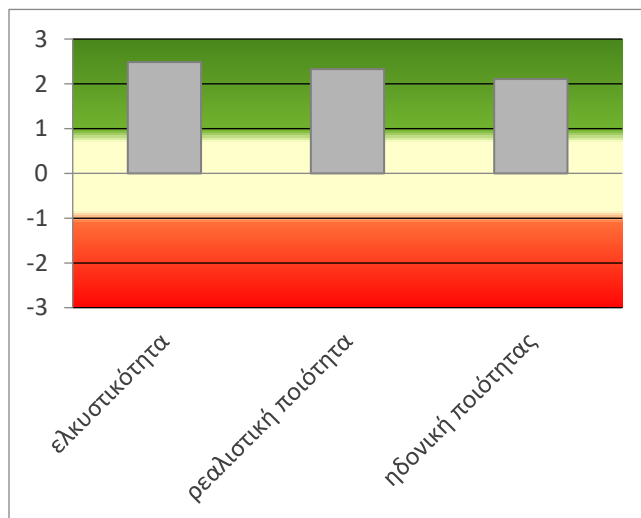


Εικόνα 5.2.3: Γραφική Παράσταση Κλιμάκων Ευχρηστίας

UEQ Scales (Mean and Variance)		
ελκυστικότητα	↑ 2.483	0.26
Ευκρίνεια	↑ 2.450	0.39
Αποδοτικότητα	↑ 2.425	0.46
Αξιοπιστία	↑ 2.100	0.34
Διέγερση	↑ 2.217	0.54
Καινοτομία	↑ 2.000	0.99

Εικόνα 5.2.4: Πίνακας Κλιμάκων Ευχρηστίας

Όπως είδαμε και στο κεφάλαιο 5.1 οι 5 κλίμακες αξιολογούν ουσιαστικά την ελκυστικότητα, τη ρεαλιστική ποιότητα και την ηδονική ποιότητα. Αυτές οι 3 κατηγορίες φαίνονται στη γραφική της εικόνας 5.2.5. Η μέση τιμή και των τριών βρίσκεται από το 2 και πάνω, στα εξαιρετικά καλά αποτελέσματα. Εδώ, προφανώς, θα είχαμε καλά αποτελέσματα αφού οι 3 κατηγορίες προέρχονται από τις πιο πάνω 6 κλίμακες, που και αυτές έχουν μία εξαιρετικά θετική αξιολόγηση. Πιο αναλυτικά φαίνεται η μέση τιμή των τριών κατηγοριών στον πίνακα της εικόνας 5.2.6.



Εικόνα 5.2.5: Γραφική Παράσταση Μέσης Τιμής Κατηγοριών Ευχρηστίας

Pragmatic and Hedonic Quality	
ελκυστικότητα	2.48
ρεαλιστική ποιότητα	2.33
ηδονική ποιότητας	2.11

Εικόνα 5.2.6: Πίνακας Μέσης Τιμής Κατηγοριών Ευχρηστίας

Κεφάλαιο 6

Μελλοντική Εργασία & Συμπεράσματα

6.1 Μελλοντική Εργασία	57
6.2 Συμπεράσματα	58

6.1 Μελλοντική Εργασία

Το σύστημα αναφοράς περιστατικών, παρόλο που έχει ολοκληρωθεί και οι απαραίτητες λειτουργίες είναι έτοιμες για χρήση, μπορεί να βελτιωθεί με μερικές επιπλέον αναβαθμίσεις.

Αρχικά, θα ήθελα προσθέσω μία λειτουργία που θα μετατρέπει το κείμενο τις ιστοσελίδας από ελληνικά σε αγγλικά και αντίστροφα. Πιστεύω πως η συγκεκριμένη λειτουργία θα προσελκύσει κι άλλα άτομα για να χρησιμοποιούν το σύστημα και να συμβάλουν και αυτοί στην ανάπτυξη των δήμων της Κύπρου. Κάποιοι που είναι πιο εξοικειωμένοι με την ελληνική γλώσσα, σίγουρα θα νιώθουν μεγαλύτερη άνεση κατά την αλληλεπίδραση τους με το σύστημα στην ελληνική γλώσσα παρά στην αγγλική, που είναι τη δεδομένη στιγμή.

Επίσης, καλό θα ήταν στο σύστημα εκτός από την κατηγοριοποίηση των προβλημάτων στους διάφορους δήμους, να κατηγοριοποιούνται και στις διάφορες υπηρεσίες όπως Αστυνομία, Πυροσβεστική, Αρχή Ηλεκτρισμού Κύπρου κ.ά. Έτσι, το πρόβλημα θα μεταφερόταν αμέσως στην αρμόδια υπηρεσία, η οποία θα αναλάμβανε την επίλυσή του.

Τέλος, παρόλο που η διπλωματική εργασία μου είχε ως στόχο τη δημιουργία μίας ιστοσελίδας, μπορεί η συγκεκριμένη ιδέα να επεκταθεί και σε εφαρμογή για κινητά σε Android και iOS. Έτσι, θα προσελκύσουμε και τα άτομα που είναι πιο εξοικειωμένα ή προτιμούν τις εφαρμογές στα κινητά παρά τις ιστοσελίδες.

6.2 Συμπεράσματα

Με την ολοκλήρωση της διπλωματικής εργασίας μου, έχω κατανοήσει σε μεγάλο βαθμό τη χρησιμότητα των “έξυπνων” πόλεων στη σημερινή κοινωνία. Είναι απαραίτητη η συνεχής βελτίωση των πόλεων μας, έτσι ώστε να ζούμε σε ένα πιο βιώσιμο περιβάλλον και σε αυτό σίγουρα μπορεί να βοηθήσει η σωστή χρήση της Επιστήμης της Πληροφορικής και της Τεχνολογίας.

Επίσης, συμπέρανα πως οι πόλεις που δίνουν την ευκαιρία στους πολίτες να συμβάλλουν στα κοινά και στην καταπολέμηση των διαφόρων προβλημάτων που εμφανίζονται, έχουν μεγαλύτερη ανάπτυξη και βελτίωση. Είναι σημαντική η άποψη και η γνώμη των πολιτών, οπότε εφαρμογές και διάφορα τεχνολογικά μέσα, που δίνουν φωνή στους πολίτες για να ειδοποιούν τις αρχές για ορισμένα προβληματικά περιστατικά, είναι απαραίτητες.

Ακόμη, παρόλο που οι πολίτες σίγουρα νοιάζονται για το καλό των συμπολιτών τους και του περιβάλλοντος που ζουν, χρειάζονται και μερικές επιπλέον επιβραβεύσεις ή κάποια άλλα έξυπνα τεχνάσματα που θα τους ωθούν στο να γίνονται καλύτεροι και να βοηθούν στην ανάπτυξη της κοινωνίας. Παράλληλα, όταν οι χρήστες χρησιμοποιούν κάποιο συγκεκριμένο σύστημα, πρέπει αυτό να είναι εύκολο στη χρήση, αποτελεσματικό και φιλικό προς τον χρήστη, αλλιώς ο χρήστης θα αποφασίσει να μην το χρησιμοποιήσει εν τέλει. Οπότε, κατά την ολοκλήρωση των εφαρμογών που δημιουργούνται θα πρέπει να παίρνουμε ανατροφοδότηση από τους χρήστες, να μετρούμε την εμπειρία χρήστη και ακολούθως να βελτιώνονται οι εφαρμογές ανάλογα.

Τελειώνοντας, νιώθω πως έχω αποκομίσει πολλές εμπειρίες και γνώσεις από την ανάπτυξη αυτού του συστήματος και θα με χαροποιούσε πολύ αν η εφαρμογή αυτή χρησιμοποιηθεί στο μέλλον από τους Δήμους του νησιού μας. Το framework Laravel σίγουρα είναι ένα εργαλείο που θα συνεχίσω να χρησιμοποιώ, όπως και όλα τα εργαλεία που χρησιμοποίησα για την ανάπτυξη της ιστοσελίδας αυτής. Ακόμα, έμαθα να οργανώνομαι και να κατανέμω το χρόνο μου ορθολογιστικά, έτσι ώστε να είναι εφικτή η διεκπεραίωση αυτού του έργου αλλά και των υπόλοιπων πανεπιστημιακών μου υποχρεώσεων. Πιστεύω πως πλέον είμαι πιο έτοιμη να ασχοληθώ με τη δημιουργία διαδικτυακών εφαρμογών και θα ήθελα να συνεχίσω να εξελίσσομαι, όπως και η επιστήμη της Πληροφορικής, γνωρίζοντας κι άλλες τεχνολογίες.

Βιβλιογραφία

- [1] “What is Crowdsourcing”, Πάρθηκε από: <https://crowdsourcingweek.com/what-is-crowdsourcing/>
- [2] “Crowdsourcing”, 4 Μαΐου, 2021, Πάρθηκε από: <https://en.wikipedia.org/wiki/Crowdsourcing>
- [3] Marshall Hargrave, “Crowdsourcing”, 8 Ιουλίου, 2019, Πάρθηκε από: <https://www.investopedia.com/terms/c/crowdsourcing.asp>
- [4] Alexander S. Gillis, “internet of things (IoT)”, Φεβρουάριος, 2020, Πάρθηκε από: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>
- [5] “Internet of things”, 3 Μαΐου, 2021, Πάρθηκε από: https://en.wikipedia.org/wiki/Internet_of_things
- [6] Matt Burgess, “What is the Internet of Things? WIRED explains”, 16 Φεβρουαρίου, 2018, Πάρθηκε από: <https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot>
- [7] “Smart city”, 30 Απριλίου, 2021, Πάρθηκε από: https://en.wikipedia.org/wiki/Smart_city
- [8] Matt Burgess, “What is a smart city? Definition and examples”, Πάρθηκε από: <https://www.twi-global.com/technical-knowledge/faqs/what-is-a-smart-city>
- [9] “What Exactly is a Smart City?”, Πάρθηκε από: <https://blog.bismart.com/en/what-is-a-smart-city>
- [10] James Bucki, “The Benefits of Mobile Computing”, 9 Δεκεμβρίου, 2019, Πάρθηκε από: <https://www.thebalancesmb.com/definition-of-mobile-computing-2533640>

- [11] “Mobile Computing”, 14 Απριλίου, 2021, Πάρθηκε από: https://en.wikipedia.org/wiki/Mobile_computing
- [12] “Snap Send Solve”, Πάρθηκε από: <https://www.snapsendsolve.com/>
- [13] “Improve my City – Δήμος Θέρμης”, Πάρθηκε από: <https://improve.smartcity.thermi.gov.gr/>
- [14] “Laravel - Overview”, 2021, Πάρθηκε από: https://www.tutorialspoint.com/laravel/laravel_overview.htm
- [15] Alex Coleman, “Creating a Basic Laravel 5 MVC Application in 10 Minutes”, 2021, Πάρθηκε από: <https://selftaughtcoders.com/from-idea-to-launch/lesson-17/laravel-5-mvc-application-in-10-minutes/>
- [16] “Laravel”, Πάρθηκε από: <https://laravel.com/>
- [17] “Laracasts”, Πάρθηκε από: <https://laracasts.com/>
- [18] “Sublime Text”, Πάρθηκε από: <https://www.sublimetext.com/3>
- [19] “Composer”, Πάρθηκε από: <https://getcomposer.org/>
- [20] “Composer (software)”, 26 Απριλίου, 2021, Πάρθηκε από: [https://en.wikipedia.org/wiki/Composer_\(software\)](https://en.wikipedia.org/wiki/Composer_(software))
- [21] “PHP”, 26 Απριλίου, 2021, Πάρθηκε από: <https://en.wikipedia.org/wiki/PHP>
- [22] “The Best Ways to Learn HTML”, 24 Ιουλίου, 2020, Πάρθηκε από: <https://www.udacity.com/blog/2020/06/the-best-ways-to-learn-html.html>

- [23] MDN contributors, “HTML: HyperText Markup Language”, 14 Απριλίου, 2021, Πάρθηκε από: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [24] “HTML”, 9 Απριλίου, 2021, Πάρθηκε από: <https://en.wikipedia.org/wiki/HTML>
- [25] “CSS”, 4 Μαΐου, 2021, Πάρθηκε από: <https://en.wikipedia.org/wiki/CSS>
- [26] “JavaScript”, 29 Απριλίου, 2021, Πάρθηκε από: <https://en.wikipedia.org/wiki/JavaScript>
- [27] MDN contributors, “What is JavaScript?”, 27 Απριλίου, 2021, Πάρθηκε από: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [28] “MySQL”, 30 Απριλίου, 2021, Πάρθηκε από: <https://en.wikipedia.org/wiki/MySQL>
- [29] “MySQL Tutorial”, Πάρθηκε από: <https://www.tutorialspoint.com/mysql/index.htm>
- [30] “Database: Getting Started”, Πάρθηκε από: <https://laravel.com/docs/8.x/database>
- [31] “Laravel Eloquent ORM”, Πάρθηκε από: <https://www.w3adda.com/laravel-tutorial/laravel-eloquent-orm#:~:text=Eloquent%20ORM%20refer%20to%20an,much%20eloquent%20and%20expressive%20syntax>.
- [32] “UEQ User Experience Questionnaire”, Πάρθηκε από: <https://www.ueq-online.org/>

Παράρτημα Α

Στο Παράρτημα αυτό θα παρουσιαστούν μερικά κομμάτια του κώδικα των migrations, που χρησιμοποιήθηκαν για να κατασκευαστούν οι πίνακες στη βάση δεδομένων.

Η εικόνα Α.1 μας δείχνει το migration που δημιουργεί τον πίνακα των photos. Η συνάρτηση up() δημιουργεί τον πίνακα ενώ η down() τον διαγράφει. Τα πεδία που δημιουργούνται στον πίνακα είναι τα id, photopath και report_id. Μαζί με τα πεδία δημιουργούνται και κάποιοι περιορισμοί, όπως το autoIncrement που ορίζει ότι το id θα αυξάνεται αυτόματα και θα είναι και το primary key του πίνακα. Επίσης, photopath που θα μπορεί να φτάσει μέχρι τους 500 χαρακτήρες, ενώ το report_id θα είναι foreign key για τον πίνακα reports και αυτό ορίζεται με την εντολή “references”.

Παρομοίως, περιγράφονται και τα υπόλοιπα migrations που φαίνονται στο Παράρτημα αυτό.

```
class CreatePhotosTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('photos', function (Blueprint $table) {
            $table->integer('id')->autoIncrement();
            $table->string('photopath', 500);
            $table->integer('report_id');
            $table->foreign('report_id')->references('id')->on('reports')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('photos');
    }
}
```

Εικόνες Α.1: create_photos_table.php

```

class CreateContactsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('contacts', function (Blueprint $table) {
            $table->id();
            $table->string('name',100);
            $table->string('email',150);
            $table->string('subject',100);
            $table->text('message',5000);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('contacts');
    }
}

```

Εικόνα A.2: create_contacts_table.php

```

class CreateMunicipalitiesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('municipalities', function (Blueprint $table) {
            $table->integer('id')->autoIncrement();
            $table->string('name',50);
            $table->multiPolygon('coordinates');
            $table->integer('city_id');
            $table->foreign('city_id')->references('id')->on('cities')->onDelete('cascade');
            $table->spatialIndex('coordinates');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('municipalities');
    }
}

```

Εικόνα A.3: create_municipalities_table.php

```

class CreateUsersTable extends Migration{
    public function up(){
        Schema::create('users', function (Blueprint $table) {

            $table->integer('id')->autoIncrement();
            $table->string('name',30);
            $table->string('surname',30);
            $table->string('email',150);
            $table->integer('phone');
            $table->string('username',20)->unique();
            $table->string('password');
            $table->string('type',30)->default('Simple');
            $table->integer('report_times')->default(0);
            $table->tinyInteger('blocked')->default(0);
            $table->integer('municipality_id');
            $table->foreign('municipality_id')->references('id')->on('municipalities')->onDelete('cascade');
            $table->rememberToken();
            $table->timestamp('email_verified_at')->nullable();
            $table->float('score')->nullable();
            $table->tinyInteger('verified')->default(0);
            $table->string('verification_token',15);
            $table->timestamp('username_changed_at')->nullable();
            $table->string('username_token',15);
            $table->timestamps();

        });
    }
    public function down(){
        Schema::dropIfExists('users');
    }
}

```

Εικόνες A.4: create_users_table.php

```

class CreateCitiesTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('cities', function (Blueprint $table) {
            $table->integer('id')->autoIncrement();
            $table->string('name',10);
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('cities');
    }
}

```

Εικόνες A.5: create_cities_table.php

```

class CreateReportsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('reports', function (Blueprint $table) {
            $table->integer('id')->autoIncrement();
            $table->string('type',200);
            $table->string('description',5000);
            $table->string('status',20)->default('Unseen');
            $table->string('latitude',20);
            $table->string('longitude',20);
            $table->integer('user_id');
            $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
            $table->integer('municipality_id');
            $table->foreign('municipality_id')->references('id')->on('municipalities')->onDelete('cascade');
            $table->string('photopath1',500)->nullable();
            $table->string('photopath2',500)->nullable();
            $table->string('photopath3',500)->nullable();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('reports');
    }
}

```

Εικόνα A.6: create_reports_table.php

```

class CreatePasswordResetsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('password_resets', function (Blueprint $table) {
            $table->string('email')->index();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('password_resets');
    }
}

```

Εικόνα A.6: create_password_resers_table.php

Παράρτημα Β

Σε αυτό το Παράρτημα θα παρουσιαστούν μερικά κύρια κομμάτια του κώδικα των βασικότερων λειτουργιών του Συστήματος Αναφοράς Περιστατικών και μερική εξήγησή τους.

Menu: Ο πιο κάτω κώδικας, στις εικόνες B.1, αντιστοιχεί στο menu και βρίσκεται στο layout.blade file, από το οποίο επαναχρησιμοποιούν όλα τα υπόλοιπα blade files τον κώδικα που περιέχει. Το Menu αλλάζει αναλόγως εάν ο χρήστης είναι guest ή auth και αναλόγως αλλάζει και με τον τύπο του χρήστη.

```
@guest
<!-- Navbar Items -->
<ul class="navbar-nav items">
  <li class="nav-item">
    <a href="/irs/" class="nav-link">Home</a>
  </li>
  <li class="nav-item">
    <a href="/irs/contactus" class="nav-link">Contact Us</a>
  </li>
  <li class="nav-item">
    <a href="/irs/signup" class="nav-link">Sign Up</a>
  </li>
</ul>
@endguest
```

```
@auth
@if(Auth::user()->type=="Simple")
  <!-- Navbar Items -->
  <ul class="navbar-nav items">
    <li class="nav-item">
      <a href="/irs/" class="nav-link">Home</a>
    </li>
    <li class="nav-item">
      <a href="/irs/contactus" class="nav-link">Contact Us</a>
    </li>
    <li class="nav-item">
      <a href="/irs/signup" class="nav-link">Sign Up</a>
    </li>
    <li class="nav-item">
      <a href="/irs/reportProblem" class="nav-link">Report A Problem</a>
    </li>
  </ul>
  <!-- Navbar Icons -->
  <ul class="navbar-nav icons">
    <li class="nav-item">
      <a href="/irs/userDetails" class="nav-link">
        <i class="icon-user"></i>
      </a>
    </li>
  </ul>
</if>
```

Εικόνες B.1: Κώδικας menu

Layout File: Ο κώδικας στο layout.blade file που ορίζει ποια κομμάτια θα χρησιμοποιηθούν σε άλλα blade files είναι το @yield('header'), @yield('content') και @yield('footerScripts'). Αντίστοιχα στα blade files που θα επαναχρησιμοποιήσουν τον κώδικα θα συναντήσουμε τα εξής:

@extends ('layout'), που ορίζει από πιο file θα πάρουν τον κώδικα και @section('header') . . . @endsection, που χρησιμοποιείται για να συμπληρώσει κομμάτια κώδικα στο κομμάτι αυτό.

Sign Up: Το πιο κάτω κομμάτι κώδικα, στην εικόνα B.2, προέρχεται από το web.php file το οποίο αποτελεί το κομμάτι των routes. Ουσιαστικά, ανάλογα με το link που εισάγει ο χρήστης μας καθοδηγεί στον ανάλογο Controller και στην ανάλογη μέθοδό του.

```
//Index - Home Page
Route::get('/', 'ReportsController@index');

//Sign Up
Route::post('/signup', 'UsersController@store');
Route::get('/signup', 'CitiesController@getCities');
Route::get('/signup/getMunicipalities/{id}', 'CitiesController@getMunicipalities');
Route::get('/signup/usernameCheck/{username}', 'UsersController@isUsernameUnique');
```

Εικόνες B.2: Web.php

Η εικόνα B.3, μας προέρχεται από τον UsersController και τη μέθοδο store. Πιο συγκεκριμένα, βλέπουμε τη δημιουργία ενός αντικειμένου User με τη βοήθεια ενός μοντέλου που ονομάζεται User(). Στη συνέχεια το μοντέλο με τα δεδομένα του αποθηκεύονται στη βάση. Επίσης ένα τυχαίο token δημιουργείται και στέλνεται με email στον χρήστη. Στο τέλος επιστρέφεται με json ένα μήνυμα επιτυχίας στην οθόνη.

```
$token = Str::random(15);
$toEmail = request('email');
$now = now();
$user = new User();
$user->name = request('name');
$user->surname = request('surname');
$user->email = request('email');
$user->phone = request('phone');
$user->username = request('username');
$user->password = Hash::make(request('password'));
$user->municipality_id = request('municipality');
$user->created_at = $now;
$user->updated_at = $now;
$user->verification_token = $token;
$user->save();

\Mail::send('mail.verification',
    array(
        'token' => $token,
    ), function($message) use ($request, $toEmail)
    {
        $message->from('cmakri02@gmail.com');
        $message->to($toEmail)->subject('subject');
    }
);
return response()->json(['Success'=>"True"]);
```

Εικόνες B.3: UsersController – store

User Delete: Η εικόνα B.4 δείχνει πως ένας User γίνεται destroy στον UsersController. Πρώτα γίνεται ένα select statement από τη βάση δεδομένων για συλλογή των paths των φωτογραφιών που ανέκτησε ο χρήστης. Στη συνέχεια διαγράφονται αυτές οι φωτογραφίες από τον server, και τέλος διαγράφεται ο χρήστης και όλα τα σχετικά στοιχεία του, μαζί και οι αναφορές που έχει κάνει, από τη βάση δεδομένων.

```

$paths = DB::table('reports')->select('photopath1','photopath2','photopath3')->where('user_id',Auth::user()->id)->get();
foreach ($paths as $path) {
    if($path->photopath1 != NULL)
        Storage::delete('public/' . $path->photopath1);
    if($path->photopath2 != NULL)
        Storage::delete('public/' . $path->photopath2);
    if($path->photopath3 != NULL)
        Storage::delete('public/' . $path->photopath3);
}
$query = $user->delete();
if ($query) {
    return redirect('/')->with('success', 'Your account have been deleted successfully.');
```

Εικόνες B.4: UsersController – destroy

Report Problem: Η πιο κάτω εικόνα, B.5, μας δείχνει την μέθοδο create από τον ReportsController. Γίνεται έλεγχος στην αρχή εάν ο χρήστης είναι συνδεδεμένος στο λογαριασμό του και εάν ναι, τότε ο controller μας επιστρέφει το view reportProblem, αλλιώς μας μεταφέρει πίσω στην αρχική σελίδα.

```

public function create()
{
    if(Auth::user()){
        return view('reportProblem');
```

Εικόνες B.5: ReportsController – create

Το κομμάτι κώδικα στην εικόνα B.6, βρίσκεται στη συνάρτηση store στον ReportsController και χρησιμοποιεί κάποια Raw Queries για να ελέγξει εάν το σημείο της αναφοράς βρίσκεται μέσα στα γεωγραφικά όρια κάποιου δήμου. Αν όχι, επιστρέφει ένα μήνυμα λάθους στην οθόνη του χρήστη για να τον ενημερώσει, αλλιώς προχωράει στην ολοκλήρωση της αποθήκευσης των δεδομένων.

```

DB::statement("SET @p = ST_GeomFromText('POINT(".$lat." ".$lng."')");

$municipalityId = DB::table('municipalities')
    ->whereRaw('ST_Contains(municipalities.coordinates, @p)')
    ->pluck("id");

if(count($municipalityId)==0){
    return response()->json(['Message'=>"The location of the incident is outside the geographical boundaries of
        a municipality.",
        'Error'=>"Municipality",
        'Success'=>"False"
    ]);
}
```

Εικόνες B.6: ReportsController – store – Γεωγραφικά Όρια Δήμων

Η εικόνα B.7, μας δείχνει πως γίνεται η αποθήκευση των εικόνων στον server. Χρησιμοποιείται η μέθοδος store που αναθέτει ένα τυχαίο όνομα στο file που θα αποθηκεύσει και αποθηκεύει τα files στον φάκελο store/public/reportImages της Laravel.

```
if($isValidImage1){  
    $path1 = $request->file('ReportImage1')->store('public/reportImages');  
}
```

Εικόνες B.7: ReportsController – store –Αποθήκευση Εικόνας

User Details – Get Cities & Municipalities: Η εικόνα B.8, μας δείχνει τη συνάρτηση getCitiesSimple() στον CitiesController. Η μέθοδος αυτή ελέγχει εάν ο χρήστης είναι authenticated και αν ναι τότε επιλέγει από τη βάση τις πόλεις και τις στέλνει στην οθόνη userDetails, όπου ο χρήστης έχει τη δυνατότητα να αλλάξει τα στοιχεία του. Ένα από αυτά τα στοιχεία που μπορεί να αλλάξει είναι και ο δήμος σε περίπτωση που μετακομίσει. Οπότε στέλνονται οι πόλεις στην οθόνη, για να επιλέξει πόλη εάν το θελήσει και ακολούθως να εμφανιστούν οι ανάλογοι δήμοι που ανήκουν στην συγκεκριμένη πόλη. Ο ανάλογος κώδικας που στέλνει τους ανάλογους δήμους στην οθόνη, βάση της πόλης που θα επιλεγεί, φαίνεται στην εικόνα B.9 και βρίσκεται και αυτός στον CitiesController(). Παρόμοια διαδικασία ακολουθείται και στη λειτουργία signup.

```
public function getCitiesSimple(){  
    if(Auth::user()){  
        $cities = DB::table('cities')->pluck("name","id");  
        return view('/userDetails',compact('cities'));  
    } else{  
        return redirect('/');  
    }  
}
```

Εικόνες B.8: CitiesController – getCitiesSimple

```
public function getMunicipalities($id) {  
  
    $municipalities = DB::table("municipalities")->where("city_id",$id)->orderBy('name','ASC')->pluck("name","id");  
    return json_encode($municipalities);  
}
```

Εικόνες B.9: CitiesController – getCitiesMunicipalities

Επίσης στην εικόνα B.10 βλέπουμε το κομμάτι στο userDetails.blade και συγκεκριμένα το σημείο στη φόρμα που αφορά την επιλογή πόλης και δήμου.

```
<div class="row form-group-margin">
  <div class="col-12 col-md-12 m-0 p-2 input-group">
    <i class="icon-arrow-down mr-3"></i>
    <select id="city" name="city" class="form-control field-budget">
      <option value="" selected disabled>City</option>
      @foreach ($cities as $key => $value)
        <option value="{{ $key }}">{{ $value }}</option>
      @endforeach
    </select>
  </div>
</div>
<div class="row form-group-margin">
  <div class="col-12 col-md-12 m-0 p-2 input-group">
    <i class="icon-arrow-down mr-3"></i>
    <select id="municipality" name="municipality" class="form-control field-budget">
      <option value="" selected disabled>Municipality</option>
    </select>
  </div>
</div>
```

Εικόνες B.10: UserDetails.blade

Ακόμη, η εικόνα B.11, μας δείχνει τον κώδικα σε JavaScript που στέλνει την επιλεγμένη πόλη με ajax στον CitiesController, για να γίνει η διαδικασία εύρεσης και αποστολής των δήμων της συγκεκριμένης πόλης πίσω στην οθόνη του χρήστη.

```
<script type="text/javascript">
  jQuery(document).ready(function ()
  {
    jQuery('select[name="city"]').on('change',function(){
      var cityID = jQuery(this).val();
      if(cityID)
      {
        jQuery.ajax({
          url : '/irs/userDetails/getMunicipalities/' +cityID,
          type : "GET",
          dataType : "json",
          success:function(data)
          {
            jQuery('select[name="municipality"]').empty();
            jQuery.each(data, function(key,value){
              $('select[name="municipality"]').append('<option value="'+ key +' ">'+ value +'</option>');
            });
          }
        });
      }
      else
      {
        $('select[name="municipality"]').empty();
      }
    });
  });
</script>
```

Εικόνες B.11: UserDetails.blade – JavaScript

Contact Us: Όσον αφορά την αποστολή emails, μέσω της φόρμας contact us, μπορούμε να δούμε τον κώδικα της φόρμας στην εικόνα B.12. Η φόρμα έχει μία σταθερή μορφή και περιλαμβάνει και τα μηνύματα λάθους που θα φανούν μόνο εάν ο χρήστης εισάγει κάτι λανθασμένα ή ξεχάσει να συμπληρώσει κάποιο στοιχείο. Επίσης, σημαντικό είναι να εξηγηθεί το csrf_field, το οποίο είναι ένα κρυφό πεδίο, που είναι απαραίτητο σε κάθε φόρμα που δημιουργούμε. Η Laravel διευκολύνει την προστασία των διαδικτυακών εφαρμογών, από επιθέσεις πλαστογραφίας αιτήσεων (CSRF), οι οποίες είναι ένας τύπος κακόβουλης ενέργειας στο διαδίκτυο, με την οποία εκτελούνται μη εξουσιοδοτημένες εντολές για λογαριασμό ενός πιστοποιημένου χρήστη. Αυτό το CSRF πεδίο, που τοποθετούμε στην φόρμα, χρησιμοποιείται για να επαληθεύσει ότι ο επικυρωμένος χρήστης είναι αυτός που κάνει τα αιτήματα στην εφαρμογή και όχι κάποιος άλλος.

```
<form method="post" action="contactus">
  {{csrf_field()}}
  <div class="row form-group-margin">
    <div class="col-12 col-md-6 m-0 p-2 input-group">
      <input type="text" class="form-control @error('name') is-invalid @enderror" placeholder="Name" name="name" value="{{ old('name') }}" autofocus>
      @error('name')
        <span class="invalid-feedback" role="alert">
          <strong>{{ $message }}</strong>
        </span>
      @enderror
    </div>
    <div class="col-12 col-md-6 m-0 p-2 input-group">
      <input type="text" class="form-control @error('email') is-invalid @enderror" placeholder="Email" name="email" value="{{ old('email') }}">
      @error('email')
        <span class="invalid-feedback" role="alert">
          <strong>{{ $message }}</strong>
        </span>
      @enderror
    </div>
    <div class="col-12 col-md-12 m-0 p-2 input-group">
      <input type="text" class="form-control @error('subject') is-invalid @enderror" placeholder="Subject" name="subject">
      @error('subject')
        <span class="invalid-feedback" role="alert">
          <strong>{{ $message }}</strong>
        </span>
      @enderror
    </div>
    <div class="col-12 m-0 p-2 input-group">
      <textarea class="form-control textarea @error('message') is-invalid @enderror" placeholder="Message" name="message"></textarea>
      @error('message')
        <span class="invalid-feedback" role="alert">
          <strong>{{ $message }}</strong>
        </span>
      @enderror
    </div>
    <div class="col-12 input-group m-0 p-2">
      <button type="submit" class="btn primary-button"><i class="icon-paper-plane large"></i>Send</button>
    </div>
  </div>
</form>
```

Εικόνες B.12: contactus.blade

Παράλληλα στις εικόνες B.13 και B.14 βλέπουμε τον κώδικα στον ContactController, που αρχικά επικυρώνει τα δεδομένα, στη συνέχεια δημιουργεί ένα μοντέλο με αυτά και τα αποθηκεύει στη

βάση δεδομένων. Επίσης, αφού πάρουμε από τη βάση τα emails των διαχειριστών και τα τοποθετήσουμε σε πίνακα, η Laravel στέλνει σε όλους τους διαχειριστές ένα τυποποιημένο email με τα στοιχεία από τη φόρμα.

```
public function saveContact(Request $request) {
    //Validations
    $this->validate($request, [
        'name' => 'required|max:100',
        'email' => 'required|email|max:150',
        'subject' => 'required|max:100',
        'message' => 'required|max:5000'
    ]);

    //Save the contact email to DB.
    $contact = new Contact;

    $contact->name = $request->name;
    $contact->email = $request->email;
    $contact->subject = $request->subject;
    $contact->message = $request->message;

    $contact->save();

    //Get a collection of the administrators from the table users and eloquent model.
    $aemails = User::where('type','Administrator')->get();
```

Εικόνες B.13: ContactController – saveContact(1)

```
$adminEmails = array();
foreach($aemails as $aemail) {
    $adminEmails[] = $aemail->email;
}
Mail::send('mail.mail',
    array(
        'name' => $request->get('name'),
        'email' => $request->get('email'),
        'subject' => $request->get('subject'),
        'user_message' => $request->get('message'),
    ), function($message) use ($request,$adminEmails)
    {
        $message->from($request->email);
        $message->to($adminEmails)->subject($request->get('subject'));
    }
);
//Return back with success message.
return back()->with('success', 'Your message has been successfully sent. We will contact you soon!');
```

Εικόνες B.14: ContactController – saveContact(2)

Το email που στέλνει η Laravel φυλάγεται με τη μορφή ενός blade αρχείου, όπως φαίνεται στην εικόνα B.15.

```
<h2>Hello Admin,</h2>
<br>
<p>You received an email from : {{ $name }}</p>
<br>
<p>Here are the details:</p>
<p><b>Name:</b> {{ $name }}</p>
<p><b>Email:</b> {{ $email }}</p>
<p><b>Subject:</b> {{ $subject }}</p>
<p><b>Message:</b> {{ $user_message }}</p>
</p>
<br>
<h4><b>Sent from the site:</b>Incident Reporting System</h4>
```

Εικόνες B.15: mail.blade

Παράρτημα Γ

Στο Παράρτημα Γ θα παρουσιαστούν φωτογραφίες από το ερωτηματολόγιο που δόθηκε στο κοινό για αξιολόγηση του συστήματος.

Τίτλος έργου: Smart Citizens Incident Reporting System for municipalities

Στόχος της έρευνας: Οι στόχοι αυτής της έρευνας είναι η ανάλυση και αξιολόγηση της λειτουργικότητας του Συστήματος Αναφοράς Περιστατικών στους δήμους της Κύπρου.

Αυτό το σύστημα θα αξιολογηθεί ακολουθώντας το ερωτηματολόγιο User Experience Questionnaire (UEQ), που χρησιμοποιήθηκε σε πολλές μελέτες. Η Ρεαλιστική Ποιότητα αξιολογείται με βάση τα χαρακτηριστικά της ευκρίνειας, της αποδοτικότητας και της αξιοπιστίας, ενώ η Ήθονική Ποιότητα αξιολογείται με βάση την ευχαρίστηση και την καινοτομία που προσφέρει το προϊόν στον χρήστη.

Οδηγίες:

Για το σκοπό αυτής της έρευνας θα χρειαστεί να ενωθείτε στο VPN του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου και να μπειτε στην ιστοσελίδα <https://10.16.20.149/irs/>. Κατά την 1η επίσκεψη σας, στη σελίδα θα σας εμφανιστεί το μήνυμα "Your connection is not private", οπότε θα χρειαστεί να πατήσετε "Advanced" και "Proceed to 10.16.20.149 (unsafe)". Μετά από αυτά τα βήματα θα μπορέσετε να πλοηγηθείτε στη σελίδα, να δημιουργήσετε λογαριασμό και αφού το κάνετε αυτό, θα μπορείτε πλέον να συμπληρώσετε αυτό το ερωτηματολόγιο για την εμπειρία που είχατε στην πλοήγησή σας.

Εικόνες Γ.1: Ερωτηματολόγιο(1)

Πιο συγκεκριμένα, ακολουθήστε τα πιο κάτω βήματα για την πλοήγησή σας:

1. Δημιουργήστε λογαριασμό στο σύστημα πατώντας το sign up.
2. Κάντε login στο σύστημα με τα στοιχεία του λογαριασμού σας.
3. Αναρτήστε ένα περιστατικό που έχετε εντοπίσει πατώντας το "Report A Problem" από το menu.
 - a. Επιλέξτε τον τύπο του περιστατικού
 - b. Γράψτε μία μικρή περιγραφή
 - c. Αναρτήστε από 1 μέχρι 3 φωτογραφίες για το περιστατικό
 - d. Επιλέξτε την τοποθεσία στο χάρτη που έχει γίνει το περιστατικό
4. Τώρα μπορείτε να δείτε τα στοιχεία του λογαριασμού σας πατώντας το εικονίδιο του user στο menu. Εκτός από τα στοιχεία σας υπάρχει και η επιλογή να δείτε το Report History σας. Πατήστε το κουμπί "Report History", και θα δείτε το report που μόλις αναρτήσατε.
5. Το report που αναρτήσατε θα μελετηθεί από τους υπαλλήλους του δήμου και μόλις γίνει "Solved", θα εμφανιστεί στον χάρτη της αρχικής οθόνης με τα reports του Δήμου σας (αν βέβαια το report βρίσκεται στα γεωγραφικά όρια του δικού σας δήμου και όχι κάποιου άλλου δήμου).

* Απαιτείται

Εικόνες Γ.2: Ερωτηματολόγιο(2)

Εισαγωγή

Στην εποχή μας, τα πάντα πλέον, έχουν αυτοματοποιηθεί. Έτσι ένα online σύστημα στο οποίο θα αναφέρονται διάφορα περιστατικά που συμβαίνουν σε κάθε δήμο της Κύπρου, θεωρείται πλέον απαραίτητο. Ο στόχος αυτού του έργου είναι να παρέχει μια απλοποιημένη διαδικασία και μέθοδο που επιτρέπει την αναφορά διαφόρων περιστατικών που συμβαίνουν στους δήμους της Κύπρου, όπως παράνομο παρκάρισμα, δρόμους και πεζοδρόμια που θέλουν επισκευή, σκουπίδια, γκράφιτι και άλλα. Συγκεκριμένα, οι χρήστες θα δημιουργούν λογαριασμό στην ιστοσελίδα, και στη συνέχεια θα μπορούν να ανεβάζουν μέχρι 3 φωτογραφίες του προβλήματος που έχουν εντοπίσει, να γράφουν μία μικρή περιγραφή και να επιλέγουν την τοποθεσία του περιστατικού. Το συγκεκριμένο σύστημα θα αξιολογηθεί μέσω μιας μελέτης χρήστη, που θα περιλαμβάνει τους τελικούς χρήστες και θα χρησιμοποιεί αυτό το Ερωτηματολόγιο Εμπειρίας Χρήστη (UEQ). Το UEQ παρέχει ένα γρήγορο και αξιόπιστο ερωτηματολόγιο, που χρησιμοποιείται σε διάφορα ερευνητικά έργα για τη μέτρηση της εμπειρίας χρήστη των διαδραστικών προϊόντων.

Ενημέρωση για τη Συναίνεσή σας στο Ερωτηματολόγιο

Αυτή η έρευνα δεν έχει εμπορικούς σκοπούς, οι εμπλεκόμενοι ερευνητές δεν έχουν κανένα χρηματικό όφελος από τη διεξαγωγή της και τα αποτελέσματα θα δημοσιευθούν με τη μορφή εκθέσεων και ερευνητικών εγγράφων με βάση την έρευνα. Δεν θα σας ζητήσουμε να παρέχετε προσωπικά ευαίσθητα δεδομένα σε αυτήν την έρευνα. Όλες οι πληροφορίες που συλλέγονται μέσω αυτής της έρευνας θα παραμείνουν εμπιστευτικές. Η εμπιστευτικότητα των πληροφοριών προστατεύεται υπό την επιφύλαξη τυχόν νομικών απαιτήσεων. Τα δεδομένα που θα συλλεχθούν θα προστατεύονται με κωδικό πρόσβασης και θα είναι προσβάσιμα μόνο από τους ερευνητές.

Απαντώντας σε αυτό το ερωτηματολόγιο, επιβεβαιώνετε τα πιο κάτω:

Εικόνες Γ.3: Ερωτηματολόγιο(3)

Έχω διαβάσει και κατανοήσι τον σκοπό της έρευνας. *

☐ Επιβεβαίωση

Κατανοώ ότι η συμμετοχή μου είναι εθελοντική. Μπορώ να αποχωρήσω από τη μελέτη ανά πάσα στιγμή κατά τη διάρκεια της έρευνας και δεν χρειάζεται να δώσω λόγους για τους οποίους δεν θέλω πλέον να συμμετέχω. *

☐ Επιβεβαίωση

Συμφωνώ ότι οι απαντήσεις που δίνω θα αποθηκευτούν σε ψηφιακή μορφή. Μόνο οι εμπλεκόμενοι ερευνητές θα έχουν πρόσβαση σε αυτές τις πληροφορίες και αυτές οι πληροφορίες δεν θα διανεμηθούν σε άλλο πρόσωπο ή οντότητα. *

☐ Επιβεβαίωση

Εικόνες Γ.4: Ερωτηματολόγιο(4)

Έχω διαβάσει αυτήν τη φόρμα συναίνεσης και εξουσιοδότησης. *

☐ Επιβεβαίωση

Φύλο *

☐ Άντρας

☐ Γυναίκα

Ηλικία *

☐ Κάτω των 18

☐ 18-30

☐ 31-40

☐ 41-50

☐ Άνω των 50

Επόμενο

Εικόνες Γ.5: Ερωτηματολόγιο(5)

Τίτλος έργου: Smart Citizens Incident Reporting System for municipalities

* Απαιτείται

Part 1: Αξιολόγηση του Συστήματος Αναφοράς Περιστατικών στους δήμους (IRS)

<https://10.16.20.149/irs/>

Ενοχλητικό - Απολαυστικό *

	1	2	3	4	5	6	7	
Ενοχλητικό	☐	☐	☐	☐	☐	☐	☐	Απολαυστικό

Εικόνες Γ.6: Ερωτηματολόγιο(6)

Δυσνόητο - Κατανοητό *

1 2 3 4 5 6 7

Δυσνόητο ☐ ☐ ☐ ☐ ☐ ☐ ☐ Κατανοητό

Δημιουργικό - Αναποτελεσματικό *

1 2 3 4 5 6 7

Δημιουργικό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Αναποτελεσματικό

Εύκολο - Δύσκολο στη εκμάθηση *

1 2 3 4 5 6 7

Εύκολο στη εκμάθηση ☐ ☐ ☐ ☐ ☐ ☐ ☐ Δύσκολο στη εκμάθηση

Εικόνες Γ.7: Ερωτηματολόγιο(7)

Πολύτιμο - Υποδεέστερο *

1 2 3 4 5 6 7

Πολύτιμο ☐ ☐ ☐ ☐ ☐ ☐ ☐ Υποδεέστερο

Βαρετό - Συναρπαστικό *

1 2 3 4 5 6 7

Βαρετό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Συναρπαστικό

Αδιάφορο - Ενδιαφέρον *

1 2 3 4 5 6 7

Αδιάφορο ☐ ☐ ☐ ☐ ☐ ☐ ☐ Ενδιαφέρον

Εικόνες Γ.8: Ερωτηματολόγιο(8)

Απρόβλεπτο - Προβλέψιμο *

1234567

Απρόβλεπτο
☐
☐
☐
☐
☐
☐
☐
Προβλέψιμο

Γρήγορο - Αργό *

1234567

Γρήγορο
☐
☐
☐
☐
☐
☐
☐
Αργό

Εφευρετικό - Συμβατικό *

1234567

Εφευρετικό
☐
☐
☐
☐
☐
☐
☐
Συμβατικό

Εικόνες Γ.9: Ερωτηματολόγιο(9)

Παρελκυστικό - Υποστηρικτικό *

1234567

Παρελκυστικό
☐
☐
☐
☐
☐
☐
☐
Υποστηρικτικό

Καλό - Κακό *

1234567

Καλό
☐
☐
☐
☐
☐
☐
☐
Κακό

Περίπλοκο - Εύκολο *

1234567

Περίπλοκο
☐
☐
☐
☐
☐
☐
☐
Εύκολο

Εικόνες Γ.10: Ερωτηματολόγιο(10)

Αντιπαθητικό - Συμπαθητικό *

1 2 3 4 5 6 7

Αντιπαθητικό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Συμπαθητικό

Συνηθισμένο - Πρωτοπόρο *

1 2 3 4 5 6 7

Συνηθισμένο ☐ ☐ ☐ ☐ ☐ ☐ ☐ Πρωτοπόρο

Δυσάρεστο - Ευχάριστο *

1 2 3 4 5 6 7

Δυσάρεστο ☐ ☐ ☐ ☐ ☐ ☐ ☐ Ευχάριστο

Εικόνες Γ.11: Ερωτηματολόγιο(11)

Ασφαλές - Ανασφαλές *

1 2 3 4 5 6 7

Ασφαλές ☐ ☐ ☐ ☐ ☐ ☐ ☐ Ανασφαλές

Ενθαρρυντικό - Αποθαρρυντικό *

1 2 3 4 5 6 7

Ενθαρρυντικό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Αποθαρρυντικό

Ανταποκρίνεται / Δεν ανταποκρίνεται στις προσδοκίες *

1 2 3 4 5 6 7

Ανταποκρίνεται στις προσδοκίες ☐ ☐ ☐ ☐ ☐ ☐ ☐ Δεν ανταποκρίνεται στις προσδοκίες

Εικόνες Γ.12: Ερωτηματολόγιο(12)

Ανεπαρκές - Επαρκές *

1 2 3 4 5 6 7

Ανεπαρκές ☐ ☐ ☐ ☐ ☐ ☐ ☐ Επαρκές

Σαφές -Μπερδεμένο *

1 2 3 4 5 6 7

Σαφές ☐ ☐ ☐ ☐ ☐ ☐ ☐ Μπερδεμένο

Μη πρακτικό - Πρακτικό *

1 2 3 4 5 6 7

Μη πρακτικό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Πρακτικό

Εικόνες Γ.13: Ερωτηματολόγιο(13)

Οργανωμένο - Ανοργάνωτο *

1 2 3 4 5 6 7

Οργανωμένο ☐ ☐ ☐ ☐ ☐ ☐ ☐ Ανοργάνωτο

Ελκυστικό - Απωθητικό *

1 2 3 4 5 6 7

Ελκυστικό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Απωθητικό

Φιλικό - Εχθρικό *

1 2 3 4 5 6 7

Φιλικό ☐ ☐ ☐ ☐ ☐ ☐ ☐ Εχθρικό

Εικόνες Γ.14: Ερωτηματολόγιο(14)

Συντηρητικό - Καινοτόμο *

	1	2	3	4	5	6	7	
Συντηρητικό	☐	☐	☐	☐	☐	☐	☐	Καινοτόμο

[Πίσω](#) [Υποβολή](#)

Αυτό το περιεχόμενο δεν έχει δημιουργηθεί και δεν έχει εγκριθεί από την Google. [Αναφορά κακής χρήσης](#) - [Όροι Παροχής Υπηρεσιών](#) - [Πολιτική απορρήτου](#)

Google Φόρμες

Εικόνες Γ.15: Ερωτηματολόγιο(15)