

Κινητή εφαρμογή για την αυθόρμητη αναφορά προβλημάτων σε μία
πόλη

Mobile application for the spontaneous reporting of problems in a city

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

UNIVERSITY OF THESSALY

FOR GRADUATION OF BACHELOR OF SCIENCE

BY

Tsolakidis Stefanos

supervised by:

Spyros Lalis, Assistant Professor
Athanasios Korakis, Associate Professor



Greece 2015

Περίληψη

Σε αυτή την διπλωματική θα παρουσιάσουμε ένα σύστημα για αναφορά προβλημάτων που εντοπίζουν πολίτες στον αστικό ιστό, χρησιμοποιώντας έξυπνα κινητά. Εκμεταλλευόμενοι τους αισθητήρες που περιέχουν τα έξυπνα κινητά δημιουργήσαμε μία εφαρμογή η οποία συλλέγει δεδομένα για προβλήματα και τα στέλνει σε μορφή αναφοράς και μία εφαρμογή η οποία δέχεται και προβάλλει αυτές τις αναφορές.

Λέξεις κλειδιά

Έξυπνα κινητά, Αισθητήρες, Κατανεμημένη λύση προβλημάτων, Android εφαρμογή, Διαδικτυακή εφαρμογή

Abstract

In this thesis we present a system for reporting problems detected by citizens in the urban fabric, using smartphones. Taking advantage of the sensors that smartphones contain we have created an application that collects data on problems and sends them as a report and an application that accepts and displays these reports.

Keywords

Crowdsourcing; android development; web application development; mobile sensing; crowdsensing; smartphones

Ευχαριστίες

Για την διεκπεραίωση της παρούσας εργασίας, αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Λάλη Σπύρο για την πολύτιμη βοήθεια την οποία μου πρόσφερε όπως επίσης και για τον χρόνο τον οποίο μου αφιέρωσε.

Επίσης θα ήθελα να ευχαριστήσω τους καθηγητές του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών για τις πολύτιμες γνώσεις τις οποίες μου έδωσαν τα τελευταία χρόνια τα οποία φοιτούσα στον Βόλο.

Τέλος θα ήθελα να ευχαριστήσω τους φίλους μου και την οικογένεια μου οι οποίοι μου παρείχαν ότι χρειαζόμουν σε όλη την διάρκεια των σπουδών.

Την παρούσα εργασία την αφιερώνω στην οικογένεια μου.

Περιεχόμενα

Περίληψη.....	ii
Abstract	ii
Ευχαριστίες.....	iii
Κατάλογος εικόνων.....	v
Κεφάλαιο 1 Εισαγωγή	6
Κεφάλαιο 2 Γενικός σχεδιασμός συστήματος	7
2.1 Επισκόπηση συστήματος.....	7
2.2 Βασικά συστατικά	8
2.3 Ρόλος του καθενός και αλληλεπίδραση	8
Κεφάλαιο 3 Πρωτόκολλο αλληλεπίδρασης	10
3.1 Λεκτική περιγραφή	10
3.2 Υπηρεσία μεταφοράς	11
3.3 Ορισμός/μορφή μηνυμάτων	11
Κεφάλαιο 4 Εφαρμογή στο κινητό	12
4.1 Λειτουργικότητα	12
4.2 Τεχνικές λεπτομέρειες υλοποίησης	13
4.3 Διεπαφή χρήστη	15
Κεφάλαιο 4 Εφαρμογή στο διαδίκτυο	19
4.1 Λειτουργικότητα	19
4.2 Τεχνικές λεπτομέρειες υλοποίησης	19
4.3 Διεπαφή χρήστη	22
Κεφάλαιο 5 Συμπεράσματα - Επεκτάσεις	25
Βιβλιογραφία	26

Κατάλογος εικόνων

Εικόνα 2.1: Αρχιτεκτονική συστήματος.....	7
Εικόνα 2.2:βασική λειτουργία	8
Εικόνα 3.1:επιτυχής αποστολή αναφοράς	11
Εικόνα 3.2:http μήνυμα με αναφορά	11
Εικόνα 4.1: Τυπικό σενάριο χρήσης.....	12
Εικόνα 4.2: Μηχανισμός άμεσης αποστολής	13
Εικόνα 4.3: Μηχανισμός άμεσης αποστολής	13
Εικόνα 4.4:Αρχιτεκτονική Android	14
Εικόνα 4.5: Αρχιτεκτονική διεπαφής χρήση	16
Εικόνα 4.6:Αρχική	16
Εικόνα 4.7: Σελίδα δημιουργίας αναφοράς	16
Εικόνα 4.8: Ενδεικτική δημιουργία αναφοράς	17
Εικόνα 4.9: Ενημέρωση χρήση για αποστολή αναφορών	17
Εικόνα 4.10: How to.....	17
Εικόνα 4.11: About.....	17
Εικόνα 4.12: Ρυθμίσεις.....	18
Εικόνα 4.13: Ρυθμίσεις συστήματος	18
Εικόνα 5.1: model	20
Εικόνα 5.2: View	21
Εικόνα 5.3: Controller	21
Εικόνα 5.4: Entity relationship diagram.....	22
Εικόνα 5.5:Αρχική σελίδα.....	23
Εικόνα 5.6:Όλες οι αναφορές.....	23
Εικόνα 5.7:Συγκεκριμένη αναφορά στο χάρτη	24
Εικόνα 5.8:Αναφορά	24

Κεφάλαιο 1 Εισαγωγή

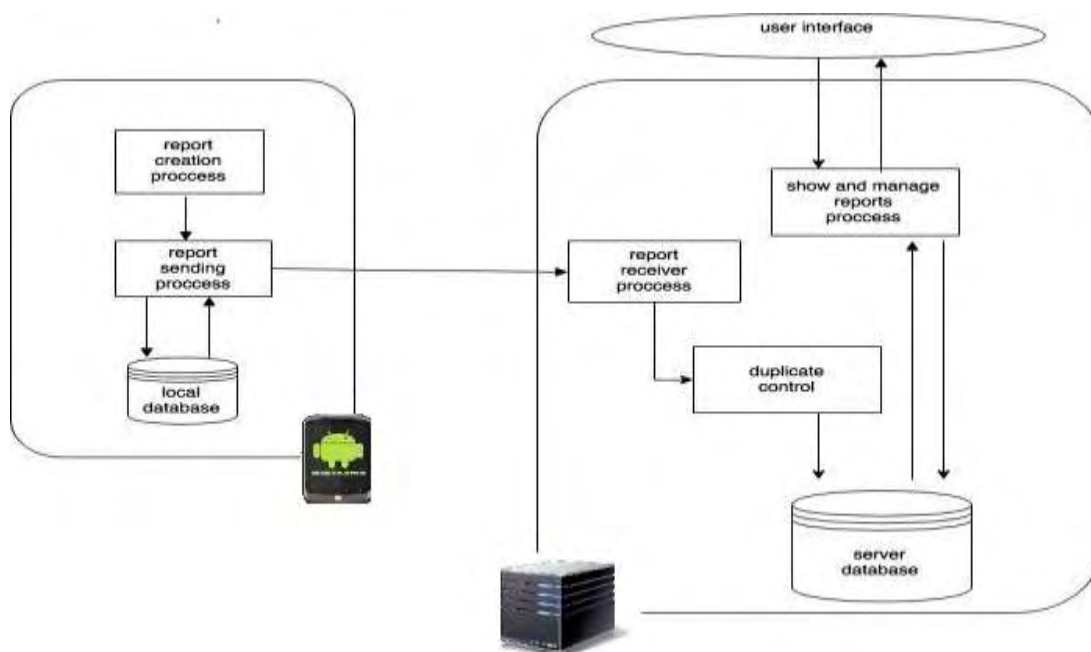
Τα τελευταία χρόνια παρατηρείται ραγδαία εξάπλωση συσκευών κινητής επικοινωνίας προηγμένης τεχνολογίας, τα οποία είναι γνωστά ως smartphones. Τα smartphones εκτός από δυνατότητα σύνδεσης στο ίντερνετ παρέχουν και ένα πλούσιο φάσμα από αισθητήρες όπως GPS, κάμερα, επιταχυνσιόμετρο. Σύμφωνα με πρόσφατες στατιστικές το 2014 οι πωλήσεις των έξυπνων κινητών αυξήθηκαν κατά 29 % σε σχέση με το 2013 κάτι που μεταφράζεται σε περίπου 66% των συνολικών πωλήσεων των κινητών συσκευών. Επίσης οι παγκόσμιοι χρήστες έξυπνων συσκευών υπολογίζονται σε 1.2 δισεκατομμύρια [1].

Ο χρήστης με την κατάλληλη αξιοποίηση των αισθητήρες μπορεί σε πραγματικό χρόνο να συλλέξει υψηλής ποιότητας δεδομένα. Η διαδικασία αυτή ανοίγει τον δρόμο για καινοτόμους τρόπους παρακολούθησης του αστικού περιβάλλοντος. Έτσι οι πολίτες μπορούν εύκολα να συλλέξουν πολυδιάστατα δεδομένα και να τα μοιραστούν χρησιμοποιώντας την υπάρχουσα υποδομή επικοινωνίας, όπως 3G ή Wi-Fi access points. Τα δεδομένα που συλλέγονται από πολλούς συμμετέχοντες μπορούν να συγκεντρωθούν και να συνδυαστούν, είτε δημιουργώντας χωροχρονικούς χάρτες είτε δίνοντας λύση σε κατανομημένα προβλήματα. Αυτός ο τρόπος συγκέντρωσης και χρησιμοποίησης δεδομένων, γνωστός ως crowdsourcing, μπορεί να οδηγήσει σε άνευ προηγουμένου βαθμό χωροχρονικής παρακολούθησης γεγονότων στον αστικό ιστό.

Στόχος αυτής της πτυχιακής είναι η δημιουργία μιας crowdsourcing εφαρμογής για συλλογή αναφορών από πολίτες. Οι αναφορές εμπεριέχουν GPS δεδομένα, φωτογραφίες και σχόλια του κάθε χρήστη που αφορούν προβλήματα που εντόπισε στην πόλη του και χρήζουν άμεσης αντιμετώπισης από τις υπηρεσίες του δήμου. Τα δεδομένα συλλέγονται σε έναν κεντρικό web server από όπου ο αρμόδιος διαχειριστής μπορεί να βλέπει τις αναφορές είτε μέσω google maps είτε μέσω λίστας, να τις δέχεται ή να τις απορρίπτει και αναλόγως να τις δρομολογεί για επίλυση.

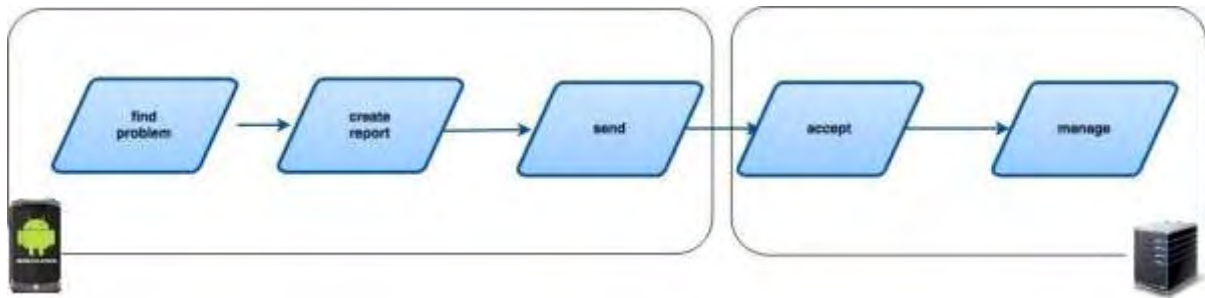
Κεφάλαιο 2 Γενικός σχεδιασμός συστήματος

2.1 Επισκόπηση συστήματος



Εικόνα 2.1: Αρχιτεκτονική συστήματος

Το σχεδιάγραμμα της αρχιτεκτονικής του συστήματός μας φαίνεται στην εικόνα 1.1. Το σύστημα δημιουργήθηκε σε μορφή πελάτη-εξυπηρετητή. Ο πελάτης είναι εφαρμογή για έξυπνα κινητά και ο εξυπηρετητής είναι η εφαρμογή που τρέχει στο διαδίκτυο. Με την εφαρμογή για έξυπνα κινητά ο πολίτης μαζεύει δεδομένα χρησιμοποιώντας αισθητήρες της συσκευής, προσθέτει μεταδεδομένα και τα στέλνει σε μορφή αναφοράς στον εξυπηρετητή (εικόνα 2.2). Συγκεκριμένα όταν ο χρήστης εντοπίσει ένα πρόβλημα στην πόλη, ανοίγει την εφαρμογή, τραβάει φωτογραφία το πρόβλημα, επιλέγει την κατηγορία του προβλήματος από μία λίστα και προσθέτει κάποια σχόλια. Αυτόματα με την λήψη της φωτογραφίας προστίθενται τα GPS δεδομένα της τοποθεσίας που βρίσκεται ο χρήστης. Αφού ολοκληρωθεί η διαδικασία δημιουργίας της αναφοράς ο χρήστης επιλέγει να στείλει την αναφορά στον εξυπηρετητή. Ο εξυπηρετητής ο οποίος τρέχει σε αξιόπιστη και πάντα διαθέσιμη υποδομή, ανήκει στην κοινότητα που τρέχει το στιγμιότυπο του συστήματός μας. Ο εξυπηρετητής όταν παραλάβει καινούργιες αναφορές ελέγχει αν είναι διπλότυπες. Αν δεν είναι τις δέχεται και τις αποθηκεύει σε μία βάση δεδομένων. Αλλιώς τις απορρίπτει. Τις αποθηκευμένες αναφορές τις προβάλλει σε ένα web portal.



Εικόνα 2.2:βασική λειτουργία

2.2 Βασικά συστατικά

Το σύστημα μας, όπως φαίνεται και στην εικόνα 2.1, αποτελείται από δύο μέρη. Την εφαρμογή στο κινητό και την εφαρμογή στο διαδίκτυο. Η επικοινωνία μεταξύ τους γίνεται μέσω διαδικτύου και όταν η εφαρμογή στο κινητό έχει συνδεσιμότητα. Η εφαρμογή στο κινητό αποτελείται από τρία βασικά συστατικά. Μια διαδικασία δημιουργίας αναφορών, μία διαδικασία αποστολής των αναφορών και μίας βάσης δεδομένων. Η εφαρμογή στο διαδίκτυο αποτελείται μία διαδικασία υποδοχής αναφορών, μία διαδικασία ελέγχου διπλοτύπων, μίας βάσης δεδομένων και μίας διαδικασίας προβολής και διαχείρισης των αναφορών. Οι αναφορές προβάλλονται στον διαχειριστή μέσω μίας ιστοσελίδας.

2.3 Ρόλος του καθενός και αλληλεπίδραση

Η εφαρμογή στο κινητό, όπως αναφέρθηκε παραπάνω, χρησιμοποιείται από τους πολίτες/χρήστες για να συλλέξουν και να στείλουν τις αναφορές στον εξυπηρετητή και η εφαρμογή στο διαδίκτυο από τον διαχειριστή του συστήματος για την παρακολούθηση και επίλυση των προβλημάτων που αναφέρονται.

Αφού ο χρήστης εντοπίσει κάποιο πρόβλημα στην πόλη του, ανοίγει την εφαρμογή αναφοράς προβλημάτων, δημιουργεί την αναφορά και πατάει αποστολή. Η αναφορά δημιουργείται με την διαδικασία δημιουργίας αναφορών και την αποστολή της αναλαμβάνει η διαδικασία αποστολής η οποία είτε στέλνει τις αναφορές άμεσα, είτε τις αποθηκεύει σε μία τοπική βάση δεδομένων για αποστολή σε μεταγενέστερο χρόνο. Η ακριβής λειτουργία και αλληλεπίδραση των βασικών συστατικών της εφαρμογής αποστολής αναφορών φαίνεται παρακάτω:

Διαδικασία δημιουργίας αναφορών: Με την διαδικασία δημιουργίας αναφορών ο χρήστης συλλέγει τα δεδομένα και δημιουργεί τις αναφορές. Αρχικά ο χρήστης τραβάει μία φωτογραφία για το πρόβλημα που εντόπισε στην πόλη του. Η εφαρμογή προσθέτει τα GPS δεδομένα του μέρους που τραβήχτηκε η φωτογραφία αυτόματα. Έπειτα ο χρήστης επιλέγει την κατηγορία προβλήματος, προσθέτει προαιρετικά κάποιο σχόλιο, πατάει αποστολή και η αναφορά μεταβιβάζεται στην διαδικασία αποστολής.

Διαδικασία αποστολής: Η διαδικασία αποστολής λαμβάνει τις αναφορές, τις μετασχηματίζει κατάλληλα και τις στέλνει στην εφαρμογή στο διαδίκτυο. Η διαδικασία αποστολής της εφαρμογής συνθέτει τα δεδομένα σε μορφή αναφοράς και δοκιμάζει αν η συσκευή που φιλοξενεί την εφαρμογή έχει συνδεσιμότητα στο διαδίκτυο. Αν ναι, τα στέλνει άμεσα στην διαδικασία υποδοχής αναφορών της εφαρμογής στο διαδίκτυο. Αλλιώς ο μηχανισμός αποστολής αποθηκεύει την αναφορά στην τοπική βάση δεδομένων. Όταν η συσκευή αποκτήσει συνδεσιμότητα στο διαδίκτυο ο μηχανισμός αποστολής ελέγχει τη τοπική βάση δεδομένων και αν υπάρχουν αποθηκευμένες αναφορές τις στέλνει στην διαδικασία υποδοχής αναφορών και τις διαγράφει από την τοπική βάση δεδομένων. Αφού η αναφορά φτάσει επιτυχώς στην διαδικασία υποδοχής αναφορών, ο μηχανισμός αποστολής ενημερώνει τον χρήστη της εφαρμογής του κινητού για την επιτυχή αποστολή της αναφοράς του μέσω της διεπαφής χρήστη.

Βάση δεδομένων: Στην βάση δεδομένων αποθηκεύονται οι αναφορές που δεν έχουν σταλεί ακόμα λόγω έλλειψης συνδεσιμότητας στο διαδίκτυο καθώς και πληροφορίες για τις αναφορές που έχουν σταλεί επιτυχώς.

Η εφαρμογή στο διαδίκτυο δέχεται καινούργιες αναφορές και προβάλλει τις προυπάρχουσες στον διαχειριστή. Όταν παραλάβει μία καινούργια ελέγχει αν είναι διπλότυπη και αν δεν είναι τη αποθηκεύει στη βάση δεδομένων. Μέσω ενός web portal ο διαχειριστής του συστήματος δει τις αναφορές και να αλλάξει την κατάσταση τους σύμφωνα με εξωτερική λογική. Η υποδοχή των καινούργιων αναφορών γίνεται με τη διαδικασία υποδοχής αναφορών. Η διαδικασία υποδοχής αναφορών αφού μετασχηματίσει τις αναφορές σε κατάλληλη μορφή τις προωθεί στον έλεγχο διπλοτύπων. Ο έλεγχος διπλοτύπων αν δεν τις θεωρήσει διπλότυπες τις αποθηκεύει στην βάση δεδομένων, αλλιώς τις απορρίπτει. Ο διαχειριστής βλέπει και διαχειρίζεται τις αποθηκευμένες αναφορές μέσω της διεπαφής χρήστη. Η ακριβής λειτουργία και αλληλεπίδραση των βασικών συστατικών της εφαρμογής στο διαδίκτυο φαίνεται παρακάτω :

Ιστοσελίδα/διεπαφή χρήστη: Η διεπαφή χρήστη χρησιμοποιείται από τον διαχειριστή του συστήματος. Ο διαχειριστής μπορεί να δει όλα τις αναφορές με διάφορα κριτήρια επιλογής και με διαφορετικές κωδικοποιήσεις αναπαράστασης. Για κάθε αναφορά μπορεί να δει συγκεκριμένες πληροφορίες όπως ID,σχόλια χρήστη, είδος report, GPS και την παρούσα κατάσταση. Επίσης ο διαχειριστής μπορεί να αλλάξει την κατάσταση της αναφοράς ανάλογα με το στάδιο που βρίσκεται η διαδικασία επίλυσης του προβλήματος. Το στάδιο της διαδικασίας επίλυσης ορίζεται από εξωτερική λογική.

Διαδικασία υποδοχής αναφορών: Η διαδικασία υποδοχής αναφορών ορίζει το πως επικοινωνούν οι πελάτες/εφαρμογές στα κινητά με την εφαρμογή στο διαδίκτυο. Τα δεδομένα που δέχεται δομούνται σε μορφή αναφοράς και δρομολογούνται για αποθήκευση.

Έλεγχος διπλοτύπων: Ο έλεγχος διπλοτύπων είναι ένα φίλτρο για τις καινούργιες αναφορές που στέλνουν οι χρήστες. Οι καινούργιες αυτές αναφορές συγκρίνονται με τις υπόλοιπες αναφορές του συστήματος για διπλότυπα. Σε περίπτωση που εμφανιστεί μεγάλη ομοιότητα με κάποια προυπάρχουσα η καινούργια αναφορά απορρίπτεται.

Βάση δεδομένων: Στην βάση δεδομένων της εφαρμογής στο διαδίκτυο αποθηκεύονται όλες οι πληροφορίες για τις αναφορές και τον διαχειριστή του συστήματος.

Κεφάλαιο 3 Πρωτόκολλο αλληλεπίδρασης

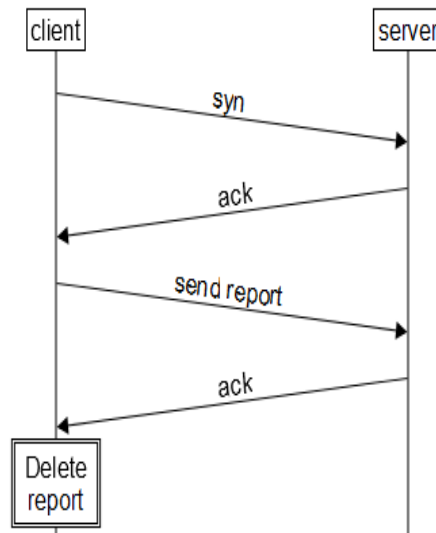
3.1 Περιγραφή

Η αποστολή των αναφορών από τον πελάτη στον εξυπηρετητή γίνεται ασύγχρονα και παρασκηνακά.

Αφού ο πελάτης επιλέξει αποστολή της αναφοράς, η εφαρμογή ελέγχει αν υπάρχει συνδεσιμότητα στο διαδίκτυο. Αν υπάρχει, στέλνει αίτηση στον εξυπηρετητή. Αν ο εξυπηρετητής απαντήσει και είναι διαθέσιμος ξεκινάει να στείλει τα δεδομένα αλλιώς αν ο εξυπηρετητής δεν είναι διαθέσιμος προσπαθεί συνολικά 4 φορές να επικοινωνήσει μαζί του. Αν δε τα καταφέρει το απορρίπτει. Αφού ο εξυπηρετητής λάβει τα δεδομένα στέλνει επιβεβαίωση στον πελάτη. Όταν ο πελάτης λάβει επιβεβαίωση για παραλαβή των δεδομένων τα διαγράφει. Όταν δεν υπάρχει άμεση συνδεσιμότητα η εφαρμογή αποθηκεύει τα δεδομένα στην τοπική βάση και περιμένει μέχρι η κινητή συσκευή να αποκτήσει συνδεσιμότητα. Όταν αποκτήσει ακολουθεί την ίδια διαδικασία με προηγούμενως.

Ο εξυπηρετητής αφού λάβει επιτυχώς τα δεδομένα, συγκρίνει την αναφορά που παρέλαβε με τα υπόλοιπα και σε περίπτωση που εντοπίσει μεγάλη ομοιότητα με κάποιο προυπάρχων χαρακτηρίζει την καινούργια αναφορά σαν διπλότυπο και το απορρίπτει. Αλλιώς, το δέχεται και το αποθηκεύει στην βάση δεδομένων του εξυπηρετητή.

Στην εικόνα 3.1 φαίνεται η ανταλλαγή μηνυμάτων για επιτυχή αποστολή μίας αναφοράς από τον πελάτη στον εξυπηρετητή. Αρχικά γίνεται ανταλλαγή 2 μηνυμάτων σε επίπεδο εφαρμογής ώστε να εγκαθιδρυθεί η επικοινωνία. Έπειτα ο πελάτης μέσω http post στέλνει το μήνυμα που περιέχει την αναφορά και όταν ο εξυπηρετητής το λάβει επιτυχώς απαντάει με επιβεβαίωση. Τέλος αφού επιβεβαιωθεί η επιτυχής παραλαβή της αναφοράς από τον εξυπηρετητή, ο πελάτης διαγράφει την αναφορά.



Εικόνα 3.1:επιτυχής αποστολή αναφοράς

3.2 Υπηρεσία μεταφοράς

Η αρχιτεκτονική του συστήματος είναι client-server. Η εφαρμογή στο κινητό αναπαριστά τον client και η εφαρμογή στο διαδίκτυο τον εξυπηρετητή. Η αποστολή των μηνυμάτων γίνεται μέσω διαδικτύου στο οποίο η κινητή συσκευή συνδέεται μέσω Wi-Fi access point ή mobile network. Η εφαρμογή δημιουργεί http αρχείο και το στέλνει χρησιμοποιώντας πρωτόκολλο http post.

Στον χρήστη παρέχεται η δυνατότητα να επιλέξει αν θέλει οι αναφορές να στέλνονται μέσω mobile networks και Wi-Fi access points ή μόνο μέσω Wi-Fi access points.

3.3 Ορισμός/μορφή μηνυμάτων

Για την δημιουργία των http αρχείων από τον πελάτη χρησιμοποιήθηκε η βιβλιοθήκη Android Asynchronous http client η οποία δημιουργεί Http αρχεία με κωδικοποίηση json [2].

Το μήνυμα με την αναφορά που στέλνει ο πελάτης στον εξυπηρετητή περιέχει:

- GPS δεδομένα
- Φωτογραφία προβλήματος
- Κατηγορία προβλήματος
- Σχόλια χρήστη (προαιρετικά)
-

Στην εικόνα 3.2 φαίνεται ένα παράδειγμα ενός http μηνύματος που δημιουργεί η εφαρμογή μας το οποίο περιέχει αναφορά προβλήματος και είναι σε κωδικοποίηση json.

```

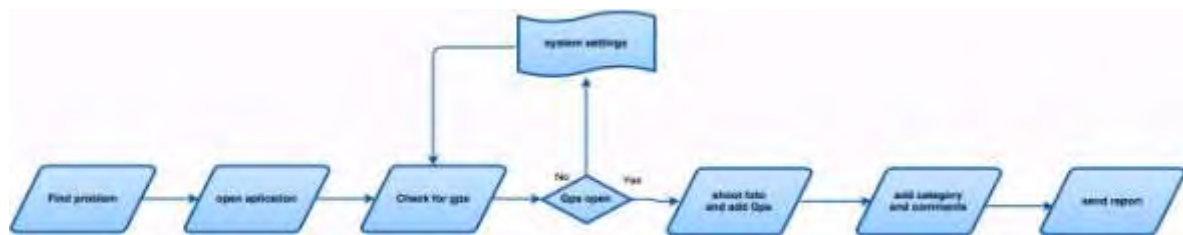
1 POST /myreceiver HTTP/1.1
2 Host: ...
3 Accept: application/json
4 Content-Type: application/json
5 Content-Length: ...
6
7 {"category":"Infrastructure",
8  "image":"image_converted_to_string",
9  "comments":"broken street lamp",
10 "langitude":"37.972492",
11 "longitude":" 23.730657"}
  
```

Εικόνα 3.2: http μήνυμα με αναφορά

Κεφάλαιο 4 Εφαρμογή αναφοράς προβλημάτων

4.1 Λειτουργικότητα

Για την συλλογή και αποστολή αναφορών από πολίτες δημιουργήθηκε μία απλή εφαρμογή για κινητές συσκευές Android. Με την εφαρμογή ο χρήστης μπορεί να διαβάσει πληροφορίες για το πώς να χρησιμοποιήσει την εφαρμογή, να έχει πρόσβαση σε ρυθμίσεις σχετικές με την εφαρμογή, να διαβάσει πληροφορίες σχετικά με εμάς και τέλος να επιλέξει να μπει στην σελίδα της αποστολής των αναφορών.



Εικόνα 4.1: Τυπικό σενάριο χρήσης

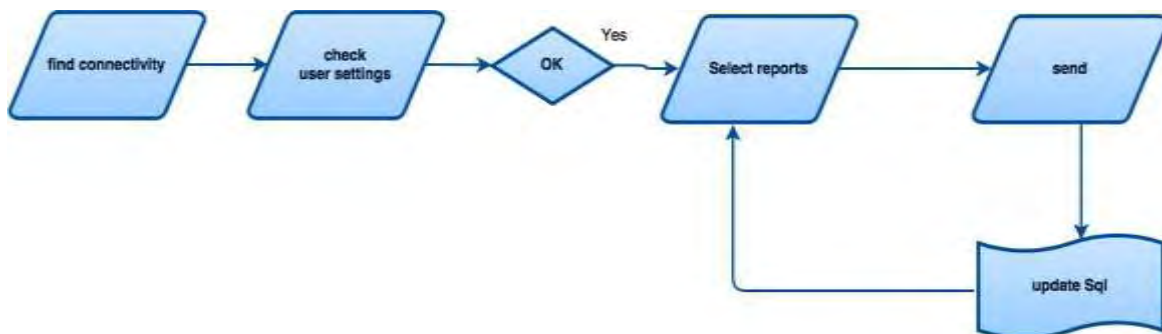
Ένα τυπικό σενάριο χρήσης φαίνεται στην εικόνα 4.1. Αρχικά ο χρήστης εντοπίζει ένα πρόβλημα στην πόλη του. Έπειτα ανοίγει την εφαρμογή αναφοράς προβλημάτων. Αφού επιλέξει να μπει στην σελίδα αποστολής αναφοράς ξεκινάει η διαδικασία συλλογής και αποστολής δεδομένων. Πριν του επιτραπεί η είσοδος στην σελίδα αποστολής αναφοράς η εφαρμογή ελέγχει αν ο χρήστης έχει ανοιχτό το GPS. Αν όχι παραπέμπεται να το ανοίξει χρησιμοποιώντας τις εσωτερικές ρυθμίσεις του κινητού του. Όταν εισέλθει επιτυχώς ο χρήστης έχει επιλογή να τραβήξει φωτογραφία, χρησιμοποιώντας την εφαρμογή λήψης φωτογραφιών του κινητού του, να επιλέξει κατηγορία προβλήματος ανάμεσα από κάποιες προεπιλογές (infrastructure, safety, service), προαιρετικά να εισάγει σχόλια και τέλος να πατήσει αποστολή ώστε το report του να μεταβεί στον μηχανισμό αποστολής. Τα μεταδεδομένα GPS προστίθενται αυτόματα με την λήψη της φωτογραφίας. Σημειώνεται ότι ο χρήστης δεν μπορεί να στείλει αναφορά χωρίς να έχει εισάγει όλα τα δεδομένα πλην των σχολίων, που είναι προαιρετικά. Ο χρήστης έχει την δυνατότητα να δει ποιες από τις αναφορές του έχουν φτάσει επιτυχώς στην εφαρμογή στο διαδίκτυο και ποιες περιμένουν ακόμα για αποστολή. Τέλος μέσω των ρυθμίσεων μπορεί να επιλέξει ποιες υποδομές θέλει να χρησιμοποιούνται για την αποστολή των δεδομένων στον εξυπηρετητή (3G ή Wi-Fi).

Έπειτα η αναφορά προωθείται στην διαδικασία/μηχανισμό αποστολής. Ο μηχανισμός αποστολής αποτελείται από δύο κομμάτια που τρέχουν σε διαφορετική χρονική στιγμή.



Εικόνα 4.2: Μηχανισμός άμεσης αποστολής

Το πρώτο (εικόνα 4.2) τρέχει μόλις ο χρήστης δρομολογήσει την αναφορά για αποστολή και ελέγχει αν η συσκευή του χρήστη έχει συνδεσιμότητα στο διαδίκτυο καθώς και αν αυτή η συνδεσιμότητα επιτυγχάνεται μέσω της υποδομής που έχει επιλέξει ο χρήστης για να γίνει η επικοινωνία με την εφαρμογή στο διαδίκτυο. Αν υπάρχει η επιλεγμένη συνδεσιμότητα η αναφορά στέλνεται άμεσα αφού μετασχηματιστεί σε κατάλληλη μορφή για αποστολή μέσω διαδικτύου και ενημερώνεται η τοπική βάση δεδομένων για την επιτυχή αποστολή της. Αλλιώς αποθηκεύεται στη τοπική βάση δεδομένων.



Εικόνα 4.3: Μηχανισμός άμεσης αποστολής

Το δεύτερο κομμάτι (εικόνα 4.3) τρέχει σε δύο χρονικές στιγμές. Όταν ο χρήστης ανοίγει την εφαρμογή και όταν η εφαρμογή αποκτήσει συνδεσιμότητα στο διαδίκτυο, ενώ δεν είχε. Έπειτα ελέγχει αν αυτή η συνδεσιμότητα επιτυγχάνεται μέσω της υποδομής που έχει επιλέξει ο χρήστης για να γίνει η επικοινωνία με την εφαρμογή στο διαδίκτυο. Αν ναι, ελέγχει αν υπάρχουν αναφορές οι οποίες είναι αποθηκευμένες στην τοπική βάση δεδομένων και δεν έχουν σταλεί ακόμα. Αν υπάρχουν τις στέλνει, τις διαγράφει από τις προς αποστολή και ενημερώνει την βάση δεδομένων για την επιτυχή αποστολή τους.

4.2 Τεχνικές λεπτομέρειες υλοποίησης

Η εφαρμογή σχεδιάστηκε για συσκευές που τρέχουν Android v4.2.2 Jelly Bean αλλά μπορεί να τρέξει χωρίς προβλήματα και σε νεότερες εκδόσεις. Η ανάπτυξη της έγινε σε Android studio v1.1 και χρησιμοποιήθηκε το επίσημο Android Software Development Kit. Παρακάτω παρατίθενται τα δομικά συστατικά των Android εφαρμογών και πως τα χρησιμοποιήσαμε.

4.2.1 Συστατικά Android εφαρμογών

Τα συστατικά των Android εφαρμογών είναι τα βασικά δομικά υλικά κάθε εφαρμογής τα οποία παρέχονται από το λειτουργικό σύστημα [3]. Τα συστατικά αυτά είναι:

Δραστηριότητα: Κάθε δραστηριότητα αναπαριστά μία οθόνη από την διεπαφή χρήστη μαζί με την λογική που υλοποιεί η οθόνη.

Υπηρεσία: Η υπηρεσία είναι ένα συστατικό που τρέχει στο παρασκήνιο και εκτελεί λειτουργίες, όπως επικοινωνία, χωρίς να επηρεάζουν την διεπαφή χρήστη.

Πάροχοι περιεχομένου: Οι πάροχοι περιεχομένου επιτρέπουν την αποθήκευση και διαχείριση δεδομένων σε βάση δεδομένων.

Παραλήπτες εκπομπής: Οι παραλήπτες εκπομπής αντιδρούν σε μηνύματα που εκπέμπει το σύστημα.

4.2.2 Αρχιτεκτονική Android

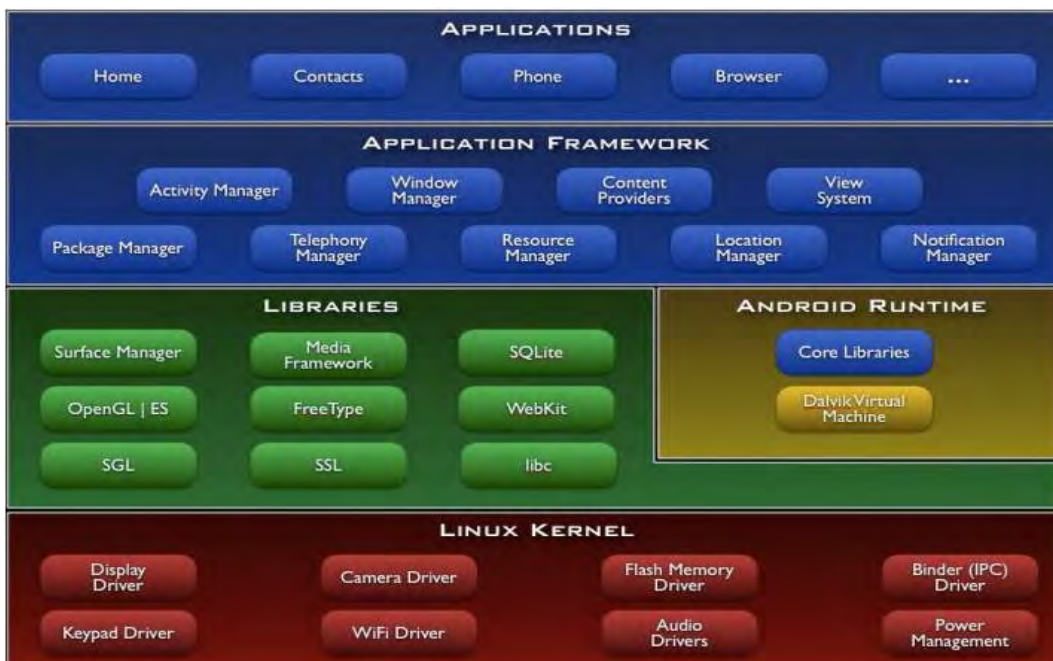
Το λειτουργικό σύστημα Android (εικόνα 4.4) βασίζεται σε πυρήνα Linux και αποτελείται από διάφορα επίπεδα που φαίνονται στην εικόνα 4.4. Κάθε επίπεδο αλληλεπιδρά με τα δίπλα του. Τα επίπεδα αυτά είναι:

Επίπεδο εφαρμογής: Σε αυτό το επίπεδο ανήκουν όλες οι βασικές εφαρμογές του λειτουργικού καθώς και εφαρμογές που έχουν δημιουργήσει τρίτοι.

Επίπεδο πλαισίου εφαρμογής: Αυτό το επίπεδο περιέχει τα βασικά συστατικά του Android (Δραστηριότητα, Υπηρεσία, Πάροχοι περιεχομένου, Παραλήπτες εκπομπής) καθώς και άλλα συστατικά του συστήματος.

Βιβλιοθήκες: Σε αυτό το επίπεδο περιέχονται οι βασικές βιβλιοθήκες του λειτουργικού καθώς και ένα σύστημα διαχείρισης βάσης δεδομένων.

Πυρήνας Linux: Εδώ περιέχονται οι οδηγοί των συσκευών οι οποίοι παρέχουν επικοινωνία με το υλικό της συσκευής.



Εικόνα 4.4: Αρχιτεκτονική Android

4.2.3 Υλοποίηση

Η εφαρμογή αναφοράς προβλημάτων έχει δύο βασικές λειτουργίες την συλλογή δεδομένων και την αποστολή των αναφορών. Αυτές ελέγχονται από μια διεπαφή χρήστη. Για την υλοποίηση των παραπάνω

χρησιμοποιήθηκαν οι μηχανισμοί και οι βιβλιοθήκες που παρέχει το Android και αναφέρθηκαν παραπάνω. Παρακάτω παρατίθενται αναλυτικά τεχνικές λεπτομέρειες υλοποίησης για κάθε κομμάτι της εφαρμογής:

Διεπαφή χρήστη: Η διεπαφή χρήστη δημιουργήθηκε χρησιμοποιώντας το Activity Manager. Ο Activity Manager ουσιαστικά ελέγχει ολόκληρο τον κύκλο ζωής της κάθε δραστηριότητας. Ο σχεδιασμός της διεπαφής χρήστη έγινε έτσι ώστε ο χρήστης να έχει πλήρη ευελιξία στην περιήγηση του.

Ρυθμίσεις χρήστη: Για την ορθή λειτουργία της εφαρμογής το GPS πρέπει να είναι ανοιχτό. Για τον έλεγχο της κατάστασης του GPS χρησιμοποιήθηκε το LocationManager το οποίο ανήκει στο επίπεδο πλαισίου της εφαρμογής. Όταν ο χρήστης προσπαθήσει να στείλει αναφορά αν δεν το έχει ανοίξει, η εφαρμογή δεν του επιτρέπει να συνεχίσει και τον παραπέμπει να το ανοίξει. Για το άνοιγμα του GPS η εφαρμογή ανοίγει της ρυθμίσεις του κινητού.

Συλλογή δεδομένων: Η συλλογή δεδομένων γίνεται σε συνεργασία του χρήστη με την συσκευή. Την φωτογραφία, την κατηγορία και τα σχόλια τα εισάγει ο χρήστης ενώ το GPS εισάγεται αυτόματα από την εφαρμογή.

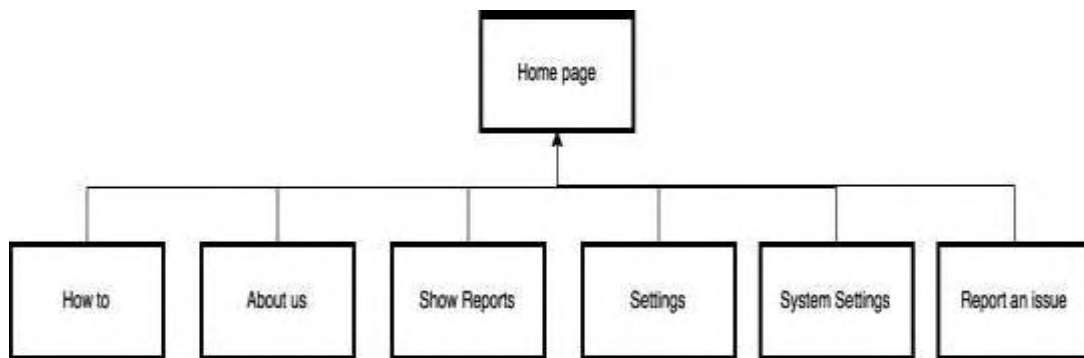
Φωτογραφίες: Για την εισαγωγή των φωτογραφιών χρησιμοποιείται η εσωτερική εφαρμογή του κινητού ώστε ο χρήστης να έχει πλήρη ευελιξία στο πώς θα τραβήξει τις φωτογραφίες. Για το άνοιγμα της εφαρμογής κάμερας του κινητού χρησιμοποιήθηκε ο μηχανισμός Intent, ο οποίος ανήκει στο επίπεδο πλαισίου εφαρμογής και επιτρέπει την εύκολη αλληλεπίδραση διαφόρων δραστηριοτήτων. Αφού τραβηχτεί η φωτογραφία εισάγεται στην εφαρμογή μας.

GPS: Για την εισαγωγή των GPS δεδομένων χρησιμοποιήθηκε το LocationManager. Το σύστημα δέχεται μόνο δορυφορικά δεδομένα καθώς μας ενδιαφέρει η ακριβής τοποθεσία της αναφοράς και όχι κατά προσέγγιση.

Αποστολή αναφορών: Η αποστολή αναφορών από την εφαρμογή στο κινητό με την εφαρμογή στο διαδίκτυο γίνεται παρασκηνακά, ασύγχρονα και εκτός του κύριου νήματος. Για την δημιουργία των μηνυμάτων και της επικοινωνίας χρησιμοποιήθηκε η βιβλιοθήκη Android Asynchronous Http Client. Πριν μπορέσει να γίνει οποιαδήποτε αποστολή, πρώτα ελέγχεται αν υπάρχει συνδεσιμότητα στο διαδίκτυο. Αυτό γίνεται χρησιμοποιώντας το ConnectivityManager. Αν δεν υπάρχει, η αναφορά αποθηκεύεται στην τοπική βάση δεδομένων και ο μηχανισμός αποστολής περιμένει μέχρι η συσκευή να αποκτήσει συνδεσιμότητα. Ο έλεγχος για την απόκτηση συνδεσιμότητας γίνεται με το NetworkStateReceiver ο οποίος είναι παραλήπτης εκπομπής και τρέχει όταν εμφανιστούν αλλαγές στην κατάσταση της σύνδεσης της συσκευής με το διαδίκτυο.

Βάση δεδομένων: Η προσωρινή αποθήκευση των αναφορών γίνεται σε Sqlite βάση δεδομένων. Η sqlite είναι μία πολύ ελαφριά βάση δεδομένων η οποία παρέχεται από το λειτουργικό σύστημα των Android. Η βάση έχει μόνο έναν πίνακα όπου αποθηκεύονται όλα τα δεδομένα της αναφοράς. Για την υλοποίηση και την αλληλεπίδραση με τη βάση δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη sqlLite που παρέχει το λειτουργικό σύστημα.

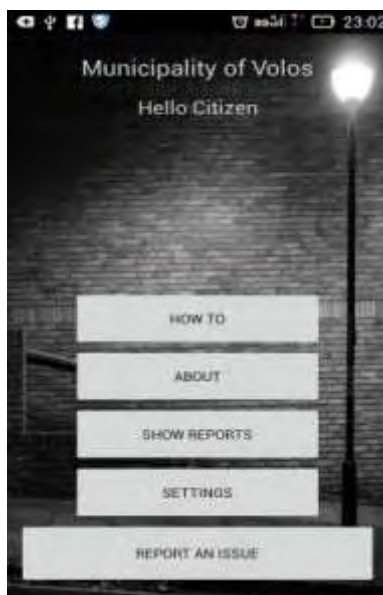
4.3 Διεπαφή χρήστη



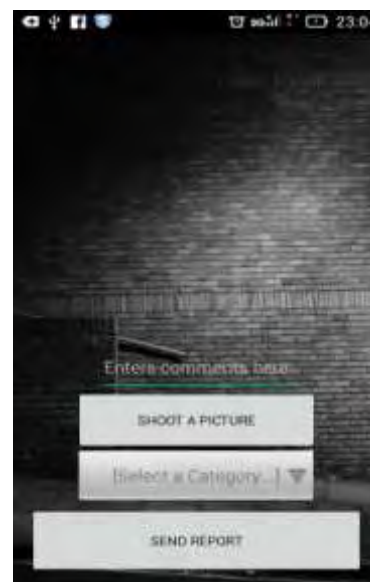
Εικόνα 4.5: Αρχιτεκτονική διεπαφής χρήστη

Η αρχιτεκτονική της διεπαφής χρήστη φαίνεται στην εικόνα 4.5. Η διεπαφή χρήστη είναι δομημένη ιεραρχικά. Παρακάτω φαίνεται το κάθε κομμάτι ξεχωριστά.

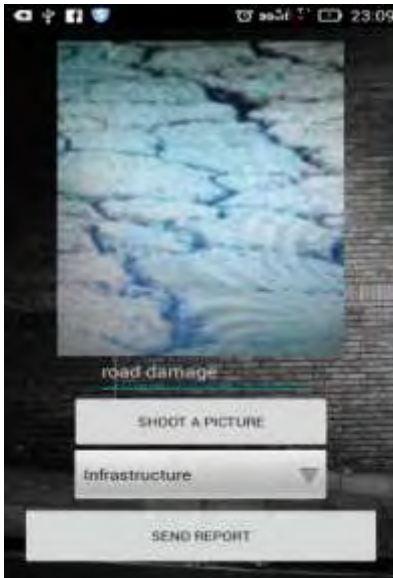
Όσον αφορά την διαδικασία δημιουργίας και αποστολής αναφοράς, όταν ανοίξει ο χρήστης την εφαρμογή εμφανίζεται η εικόνα 4.6. Όταν πατήσει αποστολή αναφοράς εμφανίζεται η εικόνα 4.7. Μία εικόνα ενδεικτικής δημιουργίας αναφοράς είναι η εικόνα 4.8. Και αφού πατήσει αποστολή μπορεί να δει το αν η αναφορά του έφτασε επιτυχώς στην εφαρμογή στο διαδίκτυο μέσω της εικόνας 4.9.



Εικόνα 4.6: Αρχική



Εικόνα 4.7: Σελίδα δημιουργίας αναφοράς



Εικόνα 4.8: Ενδεικτική δημιουργία αναφοράς



Εικόνα 4.9: Ενημέρωση χρήστη για αποστολή αναφορών

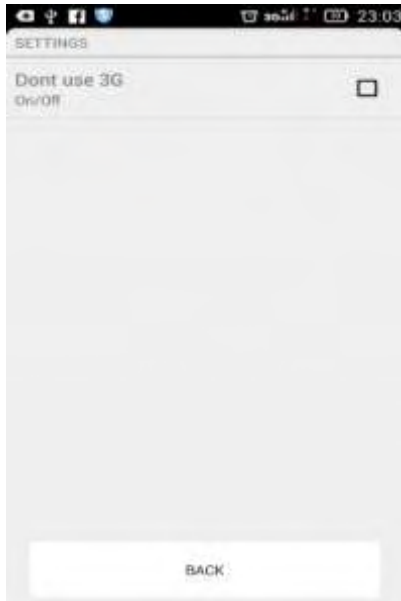
Παρατίθενται επίσης και οι εικόνες της διεπαφής για τις δευτερεύουσες λειτουργίες. Οι πληροφοριακές σελίδες How to και About φαίνονται στις εικόνες 4.10 και 4.11 αντίστοιχα, οι ρυθμίσεις στην εικόνα 4.12 και οι ρυθμίσεις συστήματος στην εικόνα 4.13.



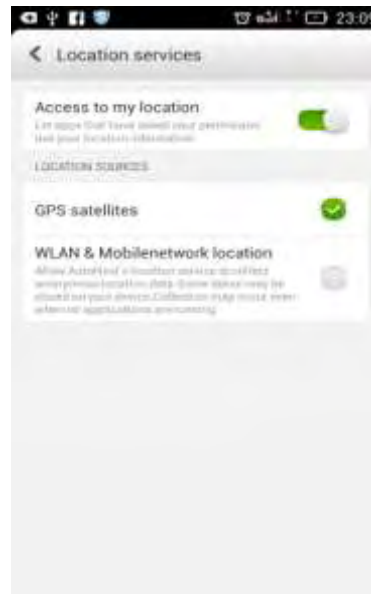
Εικόνα 4.10:How to



Εικόνα 4.11:About



Εικόνα 4.12:Ρυθμίσεις



Εικόνα 4.13:Ρυθμίσεις συστήματος

Στην προσπάθεια να κάνουμε την εφαρμογή περισσότερο φιλική προς τον χρήστη σε κάθε σελίδα της διεπαφής εκτός από την αποστολή μήκε κουμπί για επιστροφή στην προηγούμενη σελίδα. Στην σελίδα της αποστολής παραλήφθηκε καθώς συνηθίζεται ο χρήστης να πατάει το κουμπί της επιστροφής που του παρέχει η συσκευή του όταν βρίσκεται στην κύρια λειτουργία κάποιας εφαρμογής.

Κεφάλαιο 5 Εφαρμογή στο διαδίκτυο

5.1 Λειτουργικότητα

Στην εφαρμογή στο Διαδίκτυο γίνεται η συλλογή, η παρουσίαση και η διαχείριση των αναφορών που έστειλαν οι πολίτες. Η εφαρμογή στο διαδίκτυο χρησιμοποιείται από τον διαχειριστή του συστήματος. Ο διαχειριστής έχει την δυνατότητα:

- να δει όλες τις αναφορές ανάλογα την κατηγορία ή τον χρόνο υποβολής τους.
- Να δέχεται ή να απορρίπτει κάποια αναφορά
- Να βλέπει όλα τις αναφορές σε google maps
- Να βλέπει όλα τις αναφορές σε μορφή λίστας
- Να αλλάζει κατάσταση στις αναφορές ανάλογα το που βρίσκεται η διαδικασία επίλυσης του προβλήματος
- να βλέπει τη κάθε αναφορά σε ξεχωριστή καρτέλα όπου εμφανίζονται:
 - η φωτογραφία του προβλήματος
 - τα σχόλια του χρήστη
 - η ημερομηνία υποβολής της αναφοράς
 - ο μοναδικός αριθμός της αναφοράς
 - τα GPS δεδομένα της αναφοράς τα οποία παρουσιάζονται πάνω σε χάρτη
 - η παρούσα κατάστασή της αναφοράς

Η πρόσβαση στον εξυπηρετητή γίνεται μέσω κωδικών που μόνο ο διαχειριστής γνωρίζει.

Παρασκηνακά όταν λαμβάνεται μία αναφορά ελέγχεται αν η αναφορά είναι διπλότυπο και αν είναι απορρίπτετε από το σύστημα.

5.2 Τεχνικές λεπτομέρειες υλοποίησης

Η web εφαρμογή τρέχει πάνω σε Windows Server ο οποίος είναι εξοπλισμένος με 2 επεξεργαστές και 2048MB Ram. Η web εφαρμογή φιλοξενείται σε Tomcat Server v7 και είναι γραμμένη σε JSP και javascript.

5.2.1 Web application

Η εφαρμογή είναι σχεδιασμένη με βάση το γνωστό MCV μοντέλο [5]. Το κύριο πλεονέκτημα του MCV μοντέλου είναι ότι διαχωρίζει την παρουσίαση των html σελίδων από τον java κώδικα. Το model είναι αρμόδιο για την επικοινωνία της εφαρμογής με τη βάση δεδομένων, το view είναι αρμόδιο για τη δημιουργία της διεπαφής χρήστη και το controller για την αλληλεπίδραση μεταξύ model και view. Τα τρία αυτά συστατικά αναλύονται παρακάτω.

Models

Τα Model αναπαριστούν δομές δεδομένων. Το model περιέχει τις συναρτήσεις που χρησιμοποιούνται για σύνδεση με τη βάση δεδομένων και εκτέλεση βασικών Sql εντολών, όπως SELECT, INSERT, DELETE, UPDATE.

Στο model του συστήματος μας περιέχονται όλες τις συναρτήσεις που χρησιμοποιούνται κατά την διαχείριση των αναφορών καθώς και για παρασκηνακές λειτουργίες που εκτελούνται αποκλειστικά από την εφαρμογή όπως ο έλεγχος διπλοτύπων. Ένα παράδειγμα για εισαγωγή δεδομένων φαίνεται στην εικόνα 5.1. Η συνάρτηση παίρνει τα δεδομένα από τη βάση δεδομένων και δημιουργεί τις κατάλληλες δομές ώστε να σταλούν αυτά τα δεδομένα. Σημειώνεται εδώ ότι το model δεν είναι αρμόδιο για κανένα έλεγχο δεδομένων αλλά το controller που το κάλεσε.

```
while(rs2.next()){  
  
    Date = rs2.getString(1);  
    id = rs2.getString(2);  
    pst = conn  
    .prepareStatement("select id,tag,state,lat,lng,category from markers Where id="+id);  
  
    ResultSet rs = pst.executeQuery();  
  
    if(rs.next()){  
        Report report = new Report(rs.getString(1),Date,  
        getState(rs.getString(3)),rs.getString(2),rs.getString(6));  
        list.add(report);  
    }  
}
```

Εικόνα 5.1: model

Views

Τα views χρησιμοποιούνται για να δημιουργήσουν την σελίδα που θα δει ο χρήστης. Μπορούν να περιέχουν Html και Jsp κώδικα. Ένα view δεν φορτώνεται από μόνο του, ένας κατάλληλος controller πρέπει να χρησιμοποιηθεί για να καλέσει το view και να του παρέχει όποια δυναμικά δεδομένα χρειαστούν. Ουσιαστικά τα views δημιουργούν το user interface της web εφαρμογής.

Για να προσφέρουμε στον χρήστη ένα διαδραστικό γραφικό περιβάλλον χρησιμοποιήσαμε state of the art τεχνολογίες διαδικτύου όπως HTML5, CSS3, javascript, jquery, jstl. Για την παρουσίαση χωρικών δεδομένων κάναμε χρήση Google map Javascript Api v3. Τέλος τα view μας σχεδιάστηκαν χρησιμοποιώντας το easy admin, ένα γρήγορο, εύρωστο και ευπροσάρμοστο bootstrap template. Ένα παράδειγμα view φαίνεται στην εικόνα 5.2 όπου παρουσιάζονται τις αναφορές που έχει δώσει ο κατάλληλος controller σε μορφή λίστας.

```

<c:forEach var="movie" items="${requestScope.reports}"
  <tr>
    <td><c:out value="${movie.id}"/></td>
    <td><c:out value="${movie.date}"/></td>
    <td><c:out value="${movie.comments}"/></td>
    <td><c:out value="${movie.category}"/></td>
    <td><c:out value="${movie.state}"/></td>
    <td><a class="btn btn-success" href="getImageDetails_v2.jsp?id=${movie.id}">
      <i class="icon_plus_alt2"> </i> </a> </td>
  </tr>
</c:forEach>

```

Εικόνα 5.2: View

Controllers

Τα controllers είναι οι ενδιάμεσοι μεταξύ των models, views και οποιονδήποτε άλλων πόρων χρειάζονται για την επεξεργασία των http requests και την δημιουργία της σελίδας για απάντηση. Επίσης προσφέρουν ένα interface μεταξύ της web εφαρμογής και των κινητών συσκευών για την παραλαβή και έλεγχο νέων αναφορών.

Όλα τα http requests γίνονται σε κάποιο controller το οποίο σχετίζεται με ένα συγκεκριμένο url. Το controller μετά το request συλλέγει τα δεδομένα από την βάση και περνάει όλες πληροφορίες χρειάζονται στο αντίστοιχο view. Εν τέλει επιστρέφει την δημιουργημένη html σελίδα.

Ένα παράδειγμα controller φαίνεται στην εικόνα κάτι όπου ο controller συλλέγει τα δεδομένα και τα προωθεί για παρουσίαση.

```

ArrayList<testJsonForNew.place> gps = model.getPlaces();

String json = new Gson().toJson(gps);
response.setContentType("application/json");
response.setCharacterEncoding("UTF-8");
response.getWriter().write(json);

```

Εικόνα 5.3: Controller

5.2.2 Βάση δεδομένων

Για την αποθήκευση των δεδομένων στον εξυπηρετητή χρησιμοποιήθηκε MySql βάση δεδομένων. Η διαδικασία για τη δημιουργία βάσης δεδομένων είναι η αναγνώριση των οντοτήτων, των ιδιοτήτων τους καθώς και η σχέση μεταξύ τους. Οι οντότητες αναπαριστούν τους πίνακες που θα δημιουργηθούν στη βάση δεδομένων και οι ιδιότητες τα πεδία του κάθε πίνακα. Οι οντότητες που αναγνωρίσαμε στη βάση μας περιγράφονται παρακάτω:

Admin Table

Οι πληροφορίες για τον διαχειριστή ή τους διαχειριστές του συστήματος αποθηκεύονται σε αυτόν τον πίνακα. Ο διαχειριστής είναι αυτός που βλέπει τις αναφορές και καθορίζει ποιες εξωτερικές δουλειές θα γίνουν και πότε.

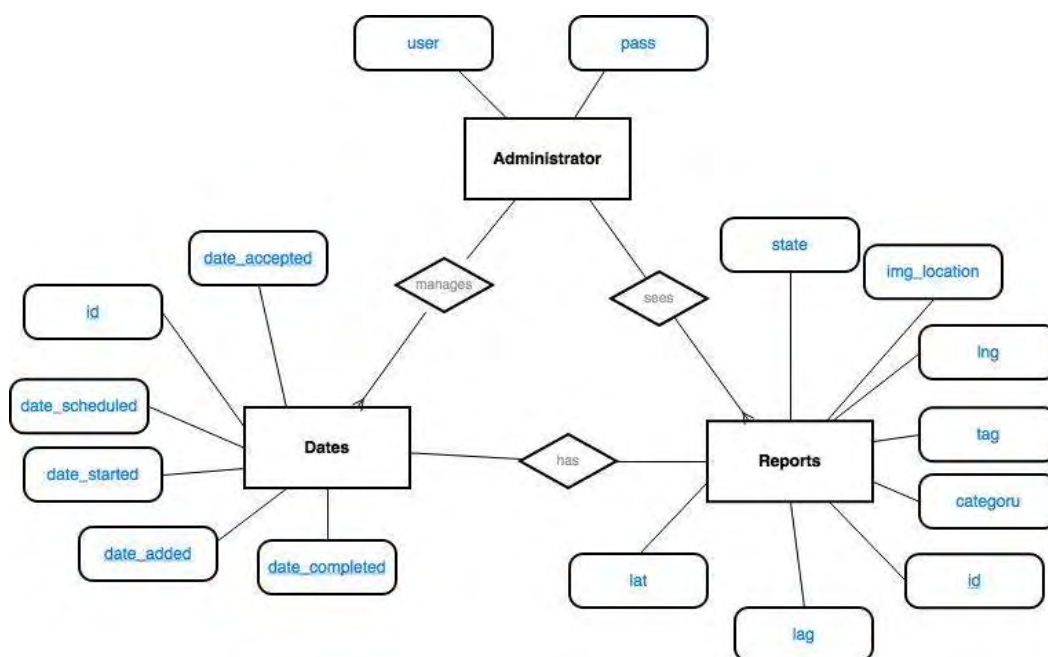
Report Table

Σε αυτόν τον πίνακα αποθηκεύονται όλα τις αναφορές που ανεβάζει ο χρήστης.

Dates Table

Σε αυτόν τον πίνακα κρατάμε δεδομένα για το πότε έγιναν εξωτερικές εργασίες οι οποίες είχαν σαν αποτέλεσμα την αλλαγή της κατάστασης κάποιας αναφοράς.

Οι οντότητες, οι ιδιότητες και οι σχέσεις μεταξύ των οντοτήτων περιγράφονται στο entity relation diagram του συστήματος, το οποίο φαίνεται στην εικόνα 5.4.



Εικόνα 5.4: Entity relationship diagram

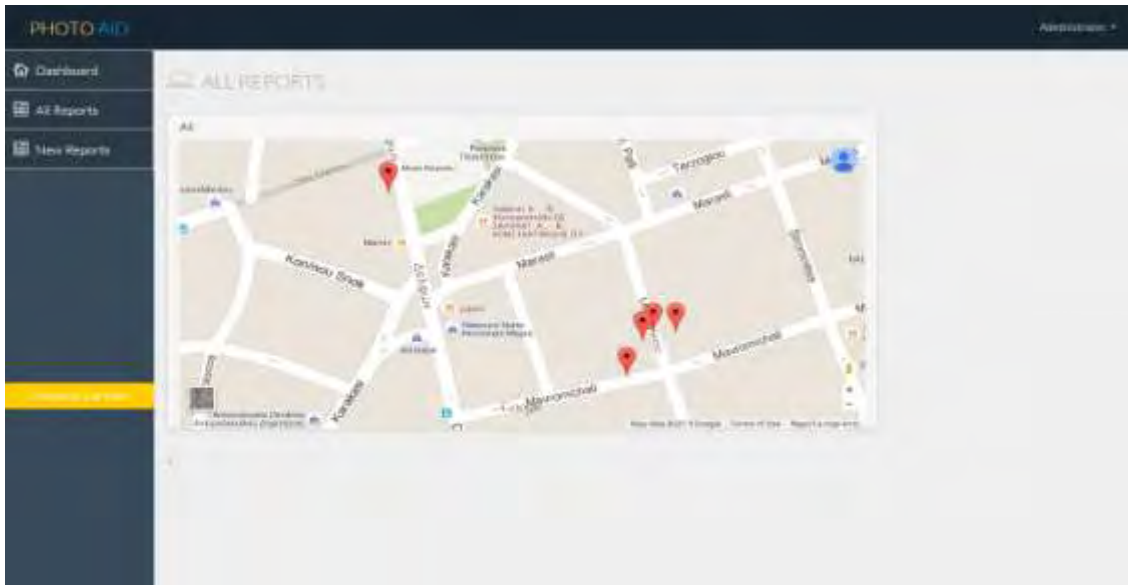
5.2.3 Αλγόριθμος ελέγχου ομοιότητας

Αφού η εφαρμογή στο διαδίκτυο παραλάβει επιτυχημένα κάποια αναφορά την προωθεί για έλεγχο διπλοτύπων. Ο έλεγχος διπλοτύπων συγκρίνει την καινούργια αναφορά με όλες τις υπόλοιπες. Αρχικά γίνεται μία πρώτη σύγκριση μετρώντας την ευκλείδεια απόσταση των GPS δεδομένων της καινούργιας αναφοράς σε σχέση με τις υπόλοιπες. Αν παρατηρηθεί προϋπάρχουσα αναφορά σε κοντινή απόσταση με την καινούργια ελέγχονται και τα σχόλια των χρηστών για παρόμοιες λέξεις καθώς επίσης λαμβάνεται υπόψιν η κατηγορία των αναφορών και η κρισιμότητά της. Τέλος η καινούργια αναφορά συγκεντρώνει κάποιο βαθμό μοναδικότητας με βάση τον οποίο η εφαρμογή επιλέγει να δεχτεί ή να απορρίψει τη νέα αναφορά.

5.3 Διεπαφή χρήστη

Η διεπαφή χρήστη της web εφαρμογής είναι προσβάσιμη μέσω ενός web browser. Παρακάτω φαίνονται οι σελίδες της εφαρμογής. Συγκεκριμένα στην εικόνα 5.5 φαίνεται η αρχική σελίδα, στην εικόνα 5.6 φαίνεται η παρουσίαση αναφορών σε μορφή λίστας, στην εικόνα 5.7 φαίνεται μία συγκεκριμένη αναφορά

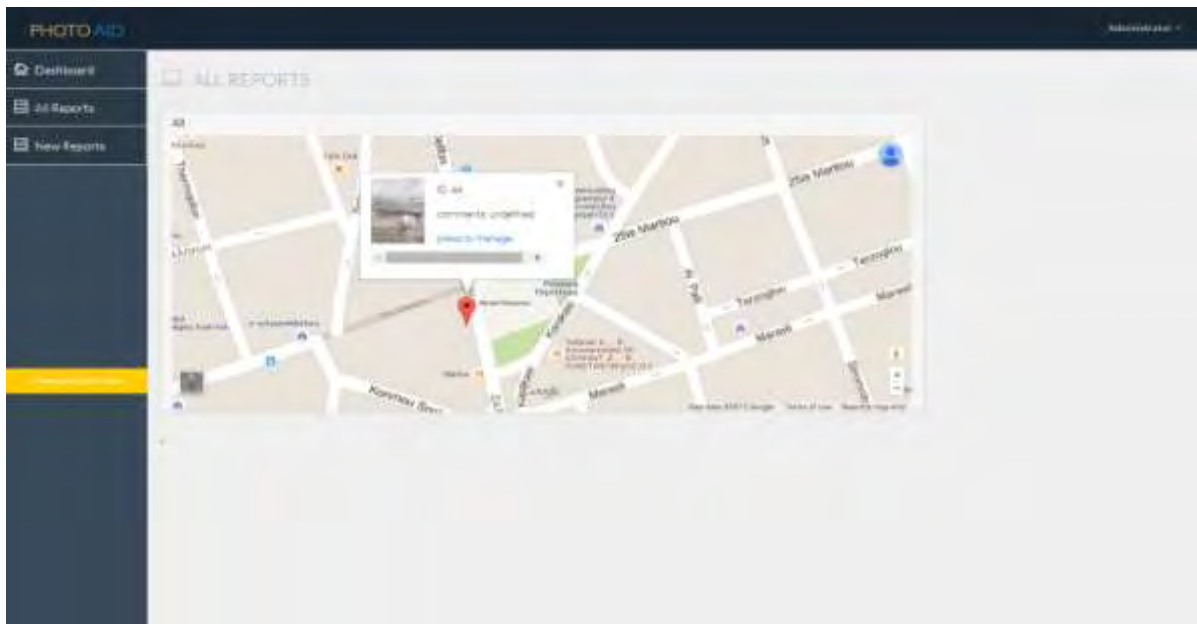
στον χάρτη μαζί με τις λεπτομέρειές της και στην εικόνα 5.8 φαίνεται η σελίδα διαχείρισης της κάθε αναφοράς. Ο χρήστης μπορεί εύκολα ανά πάσα στιγμή να αλλάξει την παρουσίαση των αναφορών από λίστα σε χάρτη και το αντίστροφο με το πάτημα ενός κουμπιού.



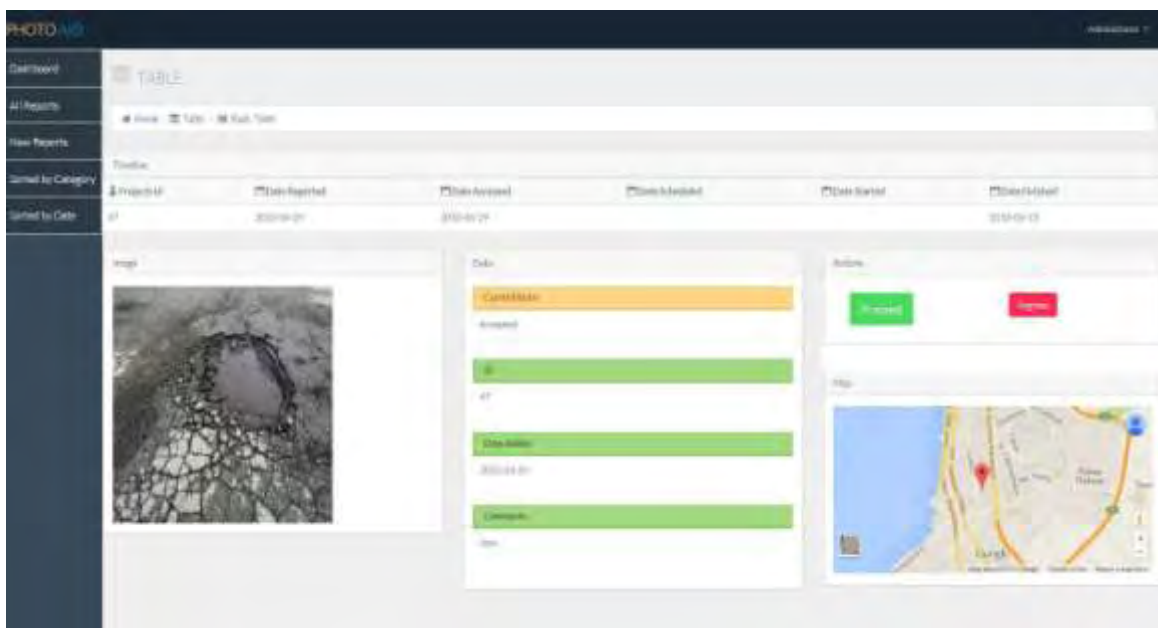
Εικόνα 5.5: Αρχική σελίδα

PHOTO AID						
						Administrator
ALL REPORTS						
New Reports						
Protocol ID	Date Added	Comments	Category	Status	Actions	
43	2015-04-29	stand alone	Infrastructure	New	[Action]	
47	2015-04-29	new	Service	New	[Action]	
Direct						
Protocol ID	Date Added	Comments	Category	Status	Actions	
32	2015-04-28	bug 34 47th 43	Infrastructure	Accepted	[Action]	
41	2015-04-28	bug 34 47th 43	Infrastructure	To Do	[Action]	
Completed						
Protocol ID	Date Added	Comments	Category	Status	Actions	
46	2015-04-29	lets	Service	Finished	[Action]	
Reported						
Protocol ID	Date Added	Comments	Category	Status	Actions	

Εικόνα 5.6: Όλες οι αναφορές



Εικόνα 5.7: Συγκεκριμένη αναφορά στο χάρτη



Εικόνα 5.8: Αναφορά

Κεφάλαιο 6 Συμπεράσματα – Επεκτάσεις

6.1 Συμπεράσματα

Σε αυτήν την εργασία παρουσιάσαμε ένα crowdsourcing σύστημα όπου οι πολίτες μπορούν να αναφέρουν προβλήματα που εντοπίζουν στην πόλη τους εύκολα και αυθόρμητα. Το σύστημα μας είναι μια εφαρμογή στον δρόμο της ανάπτυξης και καλύτερης αξιοποίησης των δυνατοτήτων των έξυπνων κινητών που εκατοντάδες εκατομμύρια άνθρωποι φέρουν μαζί τους καθημερινά.

6.2 Επεκτάσεις

Το σύστημα αν και ολοκληρωμένο επιδέχεται επιπλέον ανάπτυξης ώστε να αναβαθμιστεί και να εξαπλωθεί η χρήση του. Σαν μελλοντική δουλειά προτείνουμε την προσθήκη σύγκρισης φωτογραφιών στον έλεγχο διπλοτύπων καθώς και την δυνατότητα συγχώνευσης αναφορών όταν εντοπίζονται διπλότυπα. Επίσης την αντικατάσταση της βάσης δεδομένων με χωρική βάση για καλύτερους χρόνους απόκρισης όταν αυξάνονται τα δεδομένα. Τέλος θέλουμε να μελετήσουμε το σύστημα μας σε πραγματικό κοινό ώστε να μπορέσουμε να αξιολογήσουμε την ποιότητα των δεδομένων που παρέχουν οι χρήστες.

Βιβλιογραφία

1. Gartners Report on smartphone sales <http://www.gartner.com/newsroom/id/2996817>
2. Android Asynchronous Http Client <http://loopj.com/android-async-http/>
3. Android developer <https://developer.android.com/>
4. <http://www.eazytutz.com/android/android-architecture/>
5. tomdalling <http://www.tomdalling.com/blog/software-design/model-view-controller-explained/>