

Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Μεταπτυχιακής Διατριβής	Σχεδιασμός και Ανάπτυξη Εφαρμογής για Κινητά Τηλέφωνα σε πλατφόρμα Android με υλοποίηση στο Υπολογιστικό Νέφος. Design and Development of a Mobile Application, on Android platform with implementation in the Cloud Computing.
Όνοματεπώνυμο Φοιτητή	Χαρίδημος Κουδουμάς
Πατρώνυμο	Ανδρέας
Αριθμός Μητρώου	ΜΠΣΜ 14041
Επιβλέπων	Καθ. Χρήστος Δουληγέρης
Συνεπιβλέπων	Δρ. Σαράντης Μητρόπουλος

Ημερομηνία Παράδοσης

Οκτώβριος 2017

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης
Καθηγητής

Μιχαήλ Ψαράκης
Επίκουρος Καθηγητής

Παναγιώτης
Κοτζανικολάου
Επίκουρος
Καθηγητής

Ευχαριστίες

Η εργασία αυτή δεν θα είχε ολοκληρωθεί χωρίς την ουσιαστική συμβολή του Σ. Μητρόπουλου με το οποίο συνεργαστήκαμε στενά κατά την εκπόνηση της συγκεκριμένης μεταπτυχιακής διατριβής. Θα ήθελα να τον ευχαριστήσω θερμά για την καθοδήγησή του και τις συμβουλές που απλόχερα μου έδωσε.

Επίσης θα ήθελα να ευχαριστήσω θερμά και να δείξω την εκτίμηση και το σεβασμό μου στον καθηγητή του Πανεπιστημίου Πειραιά, κ. Χρήστο Δουληγέρη, διότι μου έδωσε τη δυνατότητα να ολοκληρώσω τις μεταπτυχιακές σπουδές μου εκπονώντας τη μεταπτυχιακή διατριβή υπό την επίβλεψή του, αλλά και γιατί παρείχε κατευθύνσεις και διορθώσεις-βελτιώσεις ώστε να υπάρξει ένα πολύ καλό αποτέλεσμα.

Επιπλέον θα ήθελα να ευχαριστήσω την οικογένεια μου, τους φίλους μου και την δεσποινίδα Ν. Α. Κοκκινάκη για τη συνεχή τους υποστήριξη όλα αυτά τα χρόνια, αλλά και για το γεγονός ότι ήταν δίπλα μου και με στηρίζουν να πετυχαίνω κάθε στόχο που θέτω.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΛΙΣΤΑ ΕΙΚΟΝΩΝ	6
ΠΕΡΙΛΗΨΗ.....	7
Abstract	8
ΚΕΦΑΛΑΙΟ 1: Η ΕΜΦΑΝΙΣΗ ΤΩΝ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ	9
1.1 Η ΠΟΡΕΙΑ ΤΩΝ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ	9
ΚΕΦΑΛΑΙΟ 2: Η ΕΠΙΛΟΓΗ ΤΟΥ ANDROID	11
2.1 ΤΙ ΕΙΝΑΙ ΤΟ ANDROID.....	11
2.2 ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID.....	12
2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ	14
2.4 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΣΕ ΛΕΙΤΟΥΡΓΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ANDROID.....	15
2.5 ΑΙΣΘΗΤΗΡΕΣ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ (MOBILE CROWDSENSING).....	16
2.5.1 ΕΦΑΡΜΟΓΕΣ ΤΩΝ MOBILE CROWDSENSING (MCS)	16
2.5.2 ΜΟΝΑΔΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ MCS.....	17
ΚΕΦΑΛΑΙΟ 3: ΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ.....	19
3.1 ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ CLOUD.....	19
3.2 ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ.....	20
3.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ CLOUD COMPUTING	20
3.4 ΒΑΣΙΚΑ ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ	22
3.5 ΑΣΦΑΛΕΙΑ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ	23
ΚΕΦΑΛΑΙΟ 4: ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΒΑΣΙΣΜΕΝΗΣ ΣΤΟΝ ΙΣΤΟ ΚΑΙ ΤΗΣ ANDROID ΕΦΑΡΜΟΓΗΣ.....	24
4.1 NETBEANS IDE	24
4.2 GLASSFISH SERVER	26
4.3. JAVA SE 8.....	26
4.4 SERVLET	27
4.5 WEB SERVICES.....	30
4.6 JavaScript και jQuery.....	31
4.7 Android Studio	31
ΚΕΦΑΛΑΙΟ 5: ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ANDROID	32
5.1 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΦΙΛΟΞΕΝΟΥΜΕΝΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ANDROID.	32
5.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ANDROID ΣΕ ΕΙΚΟΝΕΣ	34

5.3 ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΕΙΔΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	42
5.3.1 ΛΕΙΤΟΥΡΓΙΑ EMERGENCY.....	42
5.3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ.....	43
5.3.1.1 ΛΕΙΤΟΥΡΓΙΑ-ΑΓΝΟΙΑ ΘΕΣΗΣ	44
5.3.1.2 ΛΕΙΤΟΥΡΓΙΑ ΑΝΙΧΝΕΥΣΗΣ ΠΤΩΣΗΣ	48
5.3.1.3 ΕΝΗΜΕΡΩΣΗ ΕΦΑΡΜΟΓΗΣ ΦΙΛΟΞΕΝΟΥΜΕΝΗΣ ΣΤΟ ΝΕΦΟΣ.....	53
ΚΕΦΑΛΑΙΟ 6: ΠΑΡΟΥΣΙΑΣΗ ΤΗΝ ΕΦΑΡΜΟΓΗΣ ΦΙΛΞΕΝΟΥΜΕΝΗΣ ΣΤΟ ΝΕΦΟΣ	57
6.1 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	57
6.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΔΙΕΠΑΦΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΙΣΤΟΥ ΜΕ ΤΟ ΧΡΗΣΤΗ	60
ΚΕΦΑΛΑΙΟ 7: ΙΔΙΩΤΙΚΟΤΗΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	63
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	64
ΒΕΛΤΙΩΣΕΙΣ	65
ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΡΘΡΟΓΡΑΦΙΑ	66
ΔΙΑΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ.....	67
Παράρτημα Α	68
Κώδικας εφαρμογής Android	68
Παράρτημα Β	103
Κώδικας εφαρμογής φιλοξενούμενης στο σύννεφο	103
Παράρτημα Γ	117
Scripts για τη βάση σε ORACLE CLIENT.....	117

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1 Motorola DynaTAC	9
Εικόνα 2 Προβλέψεις μεριδίου αγοράς των smartphones	11
Εικόνα 3 Εκδόσεις Android	12
Εικόνα 4 Αρχιτεκτονική Android	13
Εικόνα 5 Εφαρμογές CrowdSensing	17
Εικόνα 6 Εμφάνιση όλων των συναρτήσεων μιας κλάσης στο περιβάλλον NetBeans	25
Εικόνα 7 Εργαλεία Εμφάνισης των στοιχείων μιας Εφαρμογής στο περιβάλλον NetBeans	26
Εικόνα 8 Πρωτόκολλο Client- Server	27
Εικόνα 9 Μείωση του φόρτου του επεξεργαστή με Servlet	28
Εικόνα 10 Κύκλος ζωής ενός Servlet	29
Εικόνα 11 Αρχική Οθόνη "Help The Old Ones"	34
Εικόνα 12 Εισαγωγή Επαφής Έκτακτης Ανάγκης	35
Εικόνα 13 Βασικές Λειτουργίες της Εφαρμογής (Κεντρικό Μενού)	36
Εικόνα 14 Λειτουργία «Τηλέφωνο» της Εφαρμογής	37
Εικόνα 15 Λειτουργία «Μηνύμα» της Εφαρμογής	38
Εικόνα 16 Λειτουργία «e-Mail» της εφαρμογής	39
Εικόνα 17 Λειτουργία «Επαφές» της Εφαρμογής	40
Εικόνα 18 Λειτουργία «Φωνητικές Εντολές-Αλλαγή αρ. Έκτακτης Ανάγκης» της Εφαρμογής	41
Εικόνα 19 Τιμές Αισθητήρων	42
Εικόνα 20 OnCreate της Λειτουργίας Emergency	43
Εικόνα 21 OnSensorChanged	44
Εικόνα 22 Ενημέρωση των τιμών στο GUI μέσω της UpdateUi	45
Εικόνα 23 Κλήση του Activity για την Αποστολή Μηνύματος	45
Εικόνα 24 Μήνυμα Έκτακτης Ανάγκης	46
Εικόνα 25 OnBackPressed του Activity της λειτουργία Emergency	47
Εικόνα 26 Μετρητές για την Αναγνώριση Πτώσης	48
Εικόνα 27 Έλεγχος των Μετρητών Πτώσης	49
Εικόνα 28 Activity-Button/Οθόνη που εμφανίζεται μετά από την παρέλευση τριάντα δευτερολέπτων, ύστερα από πτώση	50
Εικόνα 29 Αναπαραγωγή Ηχητικού Μηνύματος μετά από Πτώση	51
Εικόνα 30 Παύση Ηχητικού Μηνύματος	52
Εικόνα 31 Αποστολή Μηνύματος μετά από Πτώση	52
Εικόνα 32 Timmer τριών λεπτών για κλήση προς το Web Service	53
Εικόνα 33 Δημιουργία XML για αποστολή προς Webservice	54
Εικόνα 34 Δημιουργία Αντικειμένου SoapSerializationEnvelope	55
Εικόνα 35 Έλεγχος του Αποτελέσματος της 1 ^{ης} κλήσης προς το Web Service	56
Εικόνα 36 Publish to Web Service σε στατική IP	58
Εικόνα 37 Κλήση του operatrion sensorMeasurements	58
Εικόνα 38 Αποθήκευση των τιμών από τους Αισθητήρες στη βάση	59
Εικόνα 39 Αρχική σελίδα της Web Εφαρμογής	60
Εικόνα 40 Εμφάνιση της τελευταίας χρονικά θέσης της κινητής συσκευής	61
Εικόνα 41 Κλήση της Μεθόδου loadLatestData	61
Εικόνα 42 Ενημέρωση του API της Google για την εμφάνιση της τελευταίας Θέσης	62

ΠΕΡΙΛΗΨΗ

Στόχος της παρούσας μεταπτυχιακής διατριβής είναι η δημιουργία μιας εφαρμογής σε πλατφόρμα ANDROID, η οποία θα συνδέεται με μια εφαρμογή στον ιστό που θα φιλοξενείται στο υπολογιστικό νέφος.

Πρωταρχικός στόχος της εφαρμογής είναι να απλοποιήσει τη καθημερινότητα των χρηστών που δεν είναι εξοικειωμένοι με τη παραπάνω πλατφόρμα, συγκεντρώνοντας βασικές λειτουργίες σε ένα περιβάλλον. Γίνεται χρήση οπτικό-ακουστικών ερεθισμάτων ώστε να είναι σαφής και ξεκάθαρη η λειτουργικότητα που εμπεριέχεται κάνοντας ακόμη πιο εύκολη την επικοινωνία χρήστη- συσκευής.

Δευτερεύων στόχος της εφαρμογής είναι η αποφυγή περιπτώσεων που ο χρήστης δεν είναι σε θέση να προσδιορίσει την ακριβή του θέση (έχει χαθεί), ή βρίσκεται στο έδαφος ύστερα από πτώση, και δεν είναι σε θέση να χρησιμοποιήσει κάποιες από τις εφαρμογές που υπάρχουν ήδη εγκατεστημένες στην συσκευή του για να γνωστοποιήσει την κατάστασή του. Η εφαρμογή θα είναι σε θέση να κατανοήσει τέτοιες περιπτώσεις χρησιμοποιώντας τους αισθητήρες που βρίσκονται εγκατεστημένοι στο κινητό τηλέφωνο. Ενημερώνει με αντίστοιχα μηνύματα μία επαφή που έχει οριστεί ως επαφή έκτακτης ανάγκης.

Παράλληλα, καθ' όλη τη διάρκεια χρήσης της εφαρμογής λαμβάνονται δείγματα από τις μετρήσεις των αισθητήρων και αποστέλλονται στην εφαρμογή ιστού η οποία φιλοξενείται στο νέφος προκειμένου άτομο/άτομα άμεσα ενδιαφερόμενα με τον χρήστη να είναι σε θέση να γνωρίζουν ανά πάσα ώρα και στιγμή τόσο την ακριβή τοποθεσία όσο και τον τρόπο με τον οποίο κινείται στο χώρο.

Abstract

The aim of this master dissertation is to create an application in the Android platform, which is linked to a web application hosted on the Cloud.

The primary goal of the application is to simplify the daily routine of users who are not familiar with this platform, by concentrating key functionalities in one application. As it is shown in the dissertation, audio-acoustic stimulation is used to clarify the functionality that is contained by making user-device communication even easier.

A secondary goal of the application is to avoid cases in which the user is unable to determine his or her exact location (lost) or is on the ground after a fall and he/she is unable to use some of the applications already installed on the device in order to communicate its' situation. The application is able to understand such cases using the sensors installed on the mobile phone. It will automatically inform a certain corresponding phone number set as an emergency contact.

At the same time throughout the use of the application, samples will be taken from the sensors and are sent to the web application hosted in the cloud so that a person / persons directly interested in the user know exactly the location and the movement of that person.

ΚΕΦΑΛΑΙΟ 1: Η ΕΜΦΑΝΙΣΗ ΤΩΝ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ

1.1 Η ΠΟΡΕΙΑ ΤΩΝ ΚΙΝΗΤΩΝ ΤΗΛΕΦΩΝΩΝ

Ο δόκτωρ Μάρτιν Κούπερ της Motorola, περπατώντας σ' ένα δρόμο αμερικανικής μεγαλούπολης κρατούσε στα δύο του χέρια μια συσκευή που έμοιαζε με φορητό ασύρματο. Είχε μήκος 25 εκατοστά και βάρος 900 γραμμάρια. Ήταν το πρώτο σύγχρονο κινητό τηλέφωνο με κωδικό Motorola DynaTAC (Εικόνα 1).



Εικόνα 1 Motorola DynaTAC

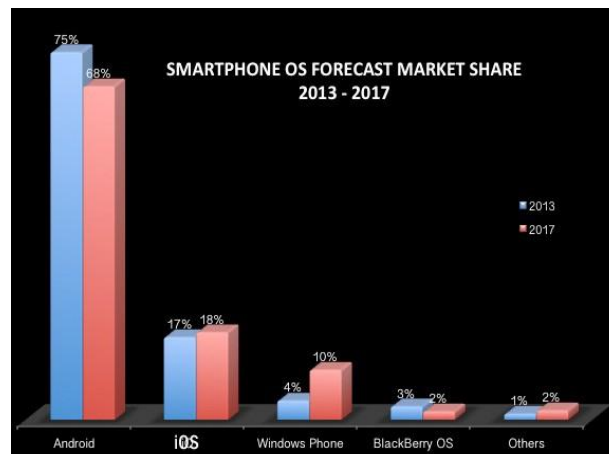
Τα κινητά τηλέφωνα δεν είχαν πάντοτε οθόνη αφής, κάμερες πολλών Μegarixels και τη δυνατότητα να χρησιμοποιηθούν ως ένας μικροσκοπικός Η/Υ, λύνοντας μας τα χέρια. Για την ακρίβεια τα πρώτα μοντέλα δεν ήταν μικρά σε μέγεθος. Εξαιτίας του μεγέθους τους δεν μπορούσαν να χαρακτηριστούν ιδιαίτερα ως κινητά. Όμως, οι όλο και αυξανόμενες απαιτήσεις των χρηστών κινητών τηλεφώνων υποχρέωσαν τους κατασκευαστές για συνεχή ανάπτυξη, με αποτέλεσμα από την κατηγορία των talkphones να φτάσουμε στην κατηγορία των smartphones. Παρακάτω αντικατοπτρίζεται ιστορικά η ανάπτυξη των κινητών τηλεφώνων και η μετάβαση στη γενιά των smartphones:

- 1979-1992: Τα κινητά τηλέφωνα έχουν ενσωματωμένα συστήματα για τον έλεγχο λειτουργίας τους.
- 1993: Το πρώτο smartphone, η IBM Simon, είχε μια οθόνη αφής, ηλεκτρονικό ταχυδρομείο και PDA χαρακτηριστικά.
- 1996: Palm Pilot 1000 προσωπικός βοηθός, στον οποίο εισάγεται το Palm OS.
- 1996: Windows CE, εισάγει φορητές συσκευές PC.
- 1999: Nokia S40 OS παρουσιάστηκε επίσημα με το λανσάρισμα του Nokia 7110.

- 2000: Symbian έγινε το πρώτο σύγχρονο λειτουργικό σύστημα σε ένα smartphone με την έναρξη της Ericsson R380.
- 2001: Η Kyocera 6035 ήταν το πρώτο smartphone με Palm OS.
- 2002: Η Microsoft εισήγαγε το πρώτο (Pocket PC) smartphone.
- 2002: Η BlackBerry εισήγαγε το πρώτο της smartphone.
- 2007: Η Apple φέρνει το πρώτο iPhone, που συνδυάζει το ' κινητό τηλέφωνο' με την επικοινωνία στο διαδίκτυο.
- 2010: Κυκλοφόρησαν τα Windows Phone OS τηλέφωνα.
- 2014: Η Apple φέρνει το πρώτο κινητό τηλέφωνο με αναγνώριση δακτυλικών αποτυπωμάτων.
- 2016: Η Samsung τοποθετεί λογισμικό προκειμένου να σκανάρει την ίριδα του ματιού με τον νέο Galaxy Note 7.

ΚΕΦΑΛΑΙΟ 2: Η ΕΠΙΛΟΓΗ ΤΟΥ ANDROID

Σκοπός της μεταπτυχιακής διατριβής είναι η ανάπτυξη μιας εφαρμογής σε μια πλατφόρμα που ικανοποιεί δύο πολύ σημαντικές παραμέτρους. Η πρώτη εξ αυτών είναι να αποτελεί μια ευρέως διαδεδομένη πλατφόρμα σε παγκόσμιο επίπεδο, και η δεύτερη να δίνει τη δυνατότητα χρήσης σε κινητά τηλέφωνα (Εικόνα 2).



Εικόνα 2 Προβλέψεις μεριδίου αγοράς των smartphones

Το Android είναι μια πλατφόρμα ανοιχτού κώδικα, το οποίο προσφέρει τη δυνατότητα εύκολης πρόσβασης σε πηγαίο κώδικα και σε εφαρμογές αυτής. Επίσης οι πωλήσεις των κινητών τηλεφώνων android παρουσιάζουν συνεχής άνοδο που σε συνδυασμό με την πληθώρα των συσκευών που κυκλοφορούν στην αγορά, η επιλογή για την ανάπτυξη της εφαρμογής σε Android αποτέλεσε μονόδρομος.

2.1 ΤΙ ΕΙΝΑΙ ΤΟ ANDROID

Το Android είναι μια στοίβα λογισμικού για κινητές συσκευές, η οποία περιλαμβάνει λειτουργικό σύστημα, ενδιάμεσο σύστημα(Middleware) και βασικές εφαρμογές. Το Android τρέχει τον πυρήνα του λειτουργικού Linux και μέσω της δικής του εργαλειοθήκης ανάπτυξης συστήματος λογισμικού (Software Development Kit), επιτρέπει στους κατασκευαστές πρωτοποριακές εφαρμογές. Αρχικά αναπτύχθηκε από την Google και αργότερα συνεχίστηκε σε συνεργασία με την Open Handset Alliance(OHA).

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού OHA, μιας κοινοπραξίας τηλεπικοινωνιακών εταιρειών, εταιρειών λογισμικού καθώς και κατασκευής υλικού, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές ανοιχτής τηλεφωνίας. Ενδεικτικά αναφέρουμε μερικά μέλη αυτού για να δείξουμε την τεράστια προοπτική που δημιουργείται:

- Sprint Nextel
- T-Mobile
- Motorola
- Samsung
- Sony Ericsson
- Vodafone
- Google
- Verizon
- HTC

2.2 ΕΚΔΟΣΕΙΣ ΤΟΥ ANDROID

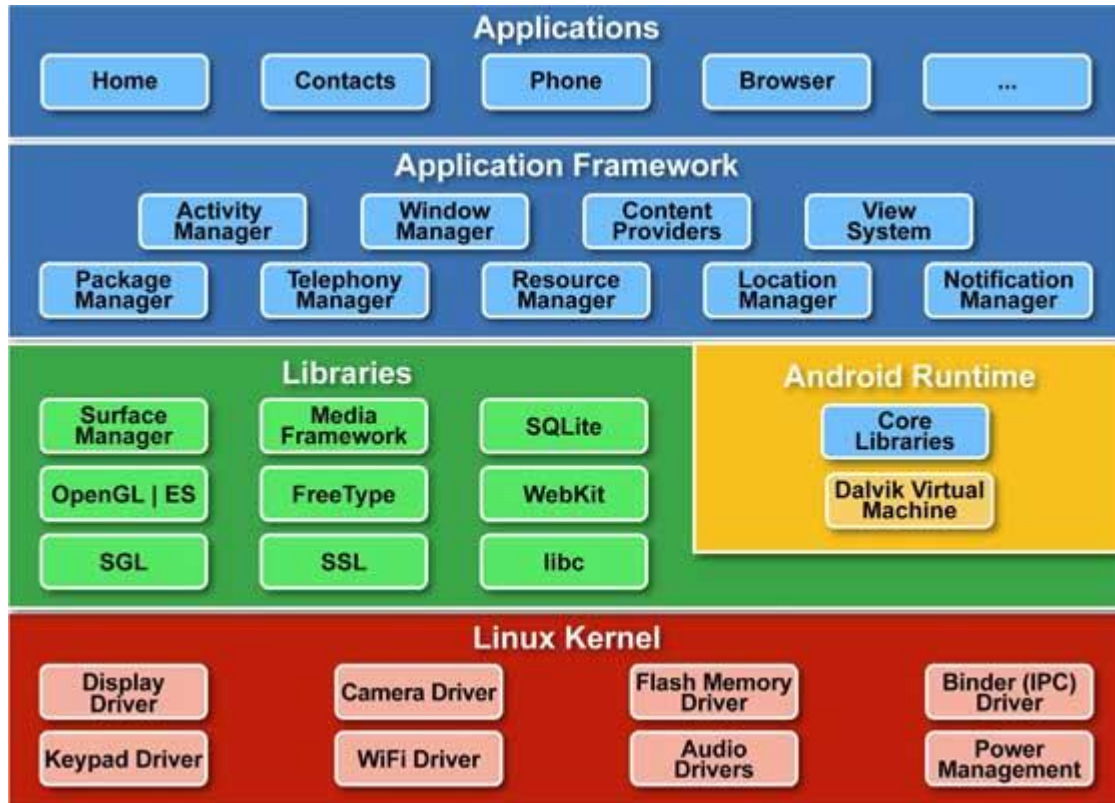
Από το 2007, όταν και εμφανίστηκε για πρώτη φορά το Android, έχουν πραγματοποιηθεί πάρα πολλές ενημερώσεις του λογισμικού με την τελευταία να γίνεται το 2016 με την κωδική ονομασία Nougat. Στην Εικόνα 3 φαίνονται όλες οι εκδόσεις του Android μέχρι και σήμερα.

Code name	Version number	Initial release date	API level	Support status
Alpha	1.0	September 23, 2008	1	Discontinued
Beta	1.1	February 9, 2009	2	Discontinued
Cupcake	1.5	April 27, 2009	3	Discontinued
Donut	1.6	September 15, 2009	4	Discontinued
Eclair	2.0 – 2.1	October 23, 2009	5–7	Discontinued
Froyo	2.2 – 2.2.3	May 20, 2010	8	Discontinued
Gingerbread	2.3 – 2.3.7	December 6, 2010	9–10	Discontinued
Honeycomb	3.0 – 3.2.6	February 22, 2011	11–13	Discontinued
Ice Cream Sandwich	4.0 – 4.0.4	October 19, 2011	14–15	Discontinued
Jelly Bean	4.1 – 4.3.1	June 27, 2012	16–18	Discontinued
KitKat	4.4 – 4.4.4	October 31, 2013	19	Discontinued
Lollipop	5.0 – 5.1.1	November 12, 2014	21–22	Discontinued
Marshmallow	6.0 – 6.0.1	October 5, 2015	23	Supported
Nougat	7.0 – 7.1.1	August 22, 2016	24–25	Supported

Εικόνα 3 Εκδόσεις Android

Σύμφωνα με τα τελευταία στοιχεία που δημοσίευσε η Google, η έκδοση Lollipop αποτελεί την πλέον διαδεδομένη έκδοση στις ενεργές συσκευές Android, παίρνοντας ένα ποσοστό της τάξεως του 36,1%, ενώ η αμέσως προηγούμενη έκδοση (Kit Kat) κατέχει το 34,3%. Επομένως είναι απόλυτα λογικό στη παρούσα διπλωματική να γίνει χρήση της έκδοσης 5.0.1 (Lollipop).

Στη Εικόνα 4 απεικονίζεται το διάγραμμα Αρχιτεκτονικής της πλατφόρμας Android:



Εικόνα 4 Αρχιτεκτονική Android

Το Android αποτελείται από:

- Εφαρμογές γραμμένες σε Java, σε εκτέλεση Dalvik
- Πλαίσιο υπηρεσιών και βιβλιοθηκών γραμμένες σε Java
- Βιβλιοθήκες γραμμένες σε C++ και C
- Πυρήνας Linux
- Περιλαμβάνει τους οδηγούς για υλικό (Hardware), δικτύωση, και την πρόσβαση του σύστημα αρχείων
- Στρώματα (layers)

2.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ

Η αρχιτεκτονική στην οποία είναι βασισμένο το Android περιλαμβάνει τα παρακάτω επίπεδα πηγαίνοντας από το χαμηλότερο στο υψηλότερο:

- Πυρήνας Linux: Το Android βασίζεται στον πυρήνα Linux έκδοση 2.6 για βασικές υπηρεσίες συστήματος όπως είναι η ασφάλεια, η διαχείριση μνήμης, η διαχείριση διεργασιών, η στοίβα δικτύου και οι οδηγίες συσκευών. Ο πυρήνας λειτουργεί επίσης ως ένα ενδιάμεσο επίπεδο αφαίρεσης μεταξύ στοίβας λογισμικού και υλικού.
- Επίπεδο Εκτέλεσης(Android Runner): Το οποίο αποτελείται από ένα σύνολο από βασικές βιβλιοθήκες και την Dalvik Virtual Machine.
- Επίπεδο Βιβλιοθηκών (Libraries): Περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C/C++ οι οποίες χρησιμοποιούνται από διάφορα στοιχεία του συστήματος του Android. Οι δυνατότητες που προσφέρουν αυτές οι βιβλιοθήκες είναι η προσβασιμότητα στους προγραμματιστές του επιπέδου πλαισίου εφαρμογών.
- Επίπεδο Πλαισίου Εφαρμογών (Application Framework): Ακολουθώντας μια ανοιχτή πλατφόρμα ανάπτυξης, το Android προσφέρει στους προγραμματιστές τη δυνατότητα να κατασκευάσουν πλούσιες και καινοτόμες εφαρμογές. Οι προγραμματιστές είναι ελεύθεροι να αξιοποιήσουν πλήρως το υλικό της συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να τρέχουν υπηρεσίες στο παρασκήνιο, και να κάνουν χρήση νημάτων για την εκτέλεση πολλών διεργασιών ταυτόχρονα. Επίσης, έχουν πλήρη πρόσβαση στο ίδιο πλαίσιο των APIs που παρέχει η εκάστοτε έκδοση του Android. Η συγκεκριμένη αρχιτεκτονική είναι με τέτοιο τρόπο διαμορφωμένη που κάθε εφαρμογή μπορεί να χρησιμοποιήσει ή να χρησιμοποιηθεί από μια άλλη, ενώ ταυτόχρονα ο χρήστης μπορεί να κάνει μικρές ή μεγάλες αλλαγές στα συστατικά της εφαρμογής.
- Ένα σύνολο από γραφικά στοιχεία (Views) για τη δημιουργία γραφικού περιβάλλοντος συμπεριλαμβανομένων λιστών, πλεγμάτων, πεδία κειμένου, κουμπιά, και επιπλέον γραφικών στοιχείων (Fragments).
- Ένα διαχειριστή περιεχομένου (Content Manager) που επιτρέπει στην εφαρμογή την πρόσβαση σε δεδομένα άλλων εφαρμογών.
- Ένα διαχειριστή πόρων (Resource Manager) για πρόσβαση στους πόρους όπως εικόνες, αρχεία ήχου κ.α.
- Ένα διαχειριστή ειδοποιήσεων(Notification Manager), ο οποίος επιτρέπει την προβολή ειδοποιήσεων στην μπάρα κατάστασης.
- Ένα διαχειριστή δραστηριοτήτων (Activity Manager), οποίος διαχειρίζεται το κύκλο ζωής των εφαρμογών.
- Επίπεδο Εφαρμογών: Η πλατφόρμα παρέχει στο χρήστη ένα σύνολο από βασικές εφαρμογές που περιλαμβάνουν πελάτη ηλ. ταχυδρομείου, SMS client, χάρτες, περιηγητή ιστού, δομή δεδομένων για την αποθήκευση επαφών κ.α, όλα γραμμένα σε γλώσσα Java.

2.4 ΒΑΣΙΚΕΣ ΑΡΧΕΣ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΣΕ ΛΕΙΤΟΥΡΓΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ANDROID

Σε αυτή την ενότητα γίνεται μια σύντομη ανάλυση των βασικών αρχών ανάπτυξης εφαρμογών σε λειτουργικό περιβάλλον Android. Για να καταλάβει κανείς τις βασικές αρχές για την ανάπτυξη εφαρμογών σε αυτό το λειτουργικό περιβάλλον πρέπει να κατέχει βασικές γνώσεις στον προγραμματισμό με γλώσσα Java, στον αντικειμενοστραφή προγραμματισμό και στα νήματα.

Μια εφαρμογή ή αλλιώς app αποτελείται από μία ή περισσότερες δραστηριότητες (εφεξής activities).

Ένα activity, είναι ένα στοιχείο εφαρμογής που παρέχει στον χρήστη μια οθόνη την οποία μπορεί να χρησιμοποιήσει για να πραγματοποιήσει μια ενέργεια, όπως για παράδειγμα να τηλεφωνήσει, να στείλει ένα email, ή να κάνει αναζήτηση μιας οδού μέσω της εφαρμογής των χαρτών. Η σχεδίαση της οθόνης ακολουθεί τις βασικές αρχές της τεχνολογίας XML. Κάθε activity έχει πολλές καταστάσεις κατά την εκτέλεσή του. Δημιουργείται, σταματάει, βρίσκεται σε παύση, συνεχίζει ή καταστρέφεται ανάλογα με το τι θέλει να κάνει ο χρήστης κάθε φορά. Για παράδειγμα, ένα activity μπορεί να παρέχει στο χρήστη ένα κουμπί, το οποίο πατώντας το να θέσει τον εαυτό του σε παύση και να ξεκινήσει ένα νέο activity. Πατώντας το κουμπί επιστροφής της συσκευής, σταματάει το activity που μόλις δημιουργήθηκε και συνεχίζει το παλιό που βρισκόταν σε παύση. Μία εφαρμογή μπορεί να αποτελείται από πολλά activities, συνδεδεμένα μεταξύ τους. Έτσι προσθέτουμε στο activity μεθόδους οι οποίες καλούνται από το σύστημα όταν η συσκευή μεταβαίνει σε μία κατάσταση. Ενδεικτικά, μερικές από αυτές είναι οι : onStart, onPause, onDestroy. Τέλος μέσα στο activity δεν είναι δυνατό να προστεθεί κώδικας ο οποίος καθυστερεί να εκτελέσει βασικές λειτουργίες του, δηλαδή την εμφάνιση της οθόνης και την κλήση μεθόδων κατάστασης. Αν γινόταν κάτι τέτοιο, η εφαρμογή θα έδειχνε να “κολλάει”. Αυτή τη δουλειά την αναθέτουμε στα νήματα (εφεξής threads).

Μία εφαρμογή μπορεί να περιέχει πολλά threads τα οποία εκτελούν τις διάφορες χρονοβόρες εργασίες που απαιτούν τα activities. Κάθε activity μπορεί να έχει το δικό του thread ή και threads. Αυτά τα threads όμως κάποιος πρέπει να τα διαχειρίζεται, να τα δημιουργεί, να τους δίνει σήμα να σταματήσουν, και να αλλάζει τις μεταβλητές τους. Αρχικά θα σκεφτόταν κάποιος ότι κάθε activity θα μπορούσε να διαχειρίζεται τα threads του. Αυτό δεν είναι εφικτό όμως για διάφορους λόγους. Για παράδειγμα, είναι επιθυμητό να τροποποιηθούν οι μεταβλητές του thread ή ακόμη και να γίνει τερματισμός του, ενώ το activity που το δημιούργησε να είναι σταματημένο. Τη δουλειά του διαχειριστή την αναλαμβάνει ο controller ο οποίος είναι υπεύθυνος για τη διαχείριση των νημάτων, για την αποθήκευση των μεταβλητών στις οποίες θέλουμε να έχουμε πρόσβαση συνέχεια ανεξαρτήτως του activity που τρέχει και πολλές φορές υπεύθυνος για τη διαχείριση των πατημάτων κουμπιών, δηλαδή ποια ενέργεια να εκτελεσθεί ή ποιο activity να ξεκινήσει με βάση το κουμπί που επιλέχθηκε. Συνήθως κάθε activity έχει τον δικό του controller.

Η αποστολή δεδομένων από τα threads στα αντίστοιχα activities επιτυγχάνεται μέσω μιας οντότητας που ονομάζεται handler. Το thread στέλνει τα δεδομένα στην κατάλληλη μορφή στον handler ο οποίος με τη σειρά του τα στέλνει στο activity εφόσον αυτό εκτελείται. Σε περίπτωση που το activity είναι σταματημένο, απλά δεν θα γίνει τίποτα. Σε περίπτωση που καλούνταν απευθείας το activity από το thread και το πρώτο ήταν σταματημένο, τότε η εφαρμογή θα παρουσίαζε πρόβλημα. Συνήθως κάθε thread έχει τον δικό του handler ή handlers αν δουλεύει για περισσότερα από ένα activities τους οποίους λαμβάνει από τους αντίστοιχους controllers.

2.5 ΑΙΣΘΗΤΗΡΕΣ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ (MOBILE CROWDSOURCING)

Με την συνεχή βελτίωση της τεχνολογίας στα κινητά τηλέφωνα ολοένα αυξάνεται η υπολογιστική ισχύς τους. Έτσι ένα μεγάλο ποσοστό ατόμων χρησιμοποιεί τα «έξυπνα» κινητά τηλέφωνα που έχουν εξοπλιστεί με πληθώρα αισθητήρων. Η πλειοψηφία των κινητών αυτών συσκευών δίνει τη δυνατότητα στους προγραμματιστές να μπορούν να έχουν πρόσβαση στα σήματα των αισθητήρων. Επιπλέον, οι crowdsensing εφαρμογές μέσα από την ικανότητα συλλογής μετρήσεων από τους αισθητήρες μπορούν να τις επεξεργαστούν με σκοπό την ορθή χρήση της πληροφορίας.

2.5.1 ΕΦΑΡΜΟΓΕΣ ΤΩΝ MOBILE CROWDSOURCING (MCS)

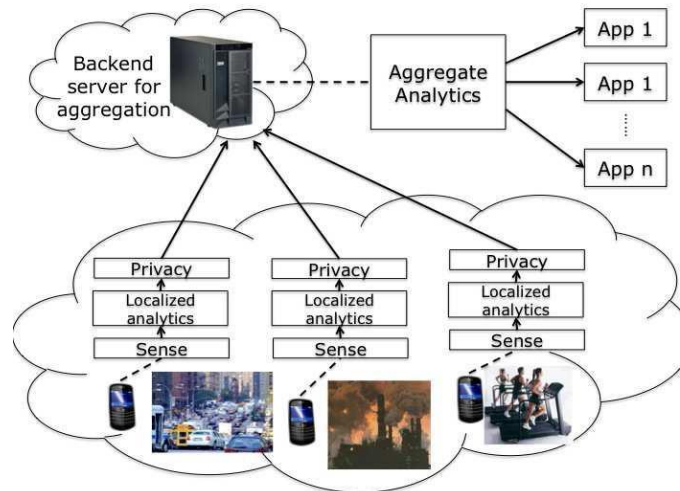
Μια αναδυόμενη κατηγορία έξυπνων κινητών συσκευών είναι αυτές που διαθέτουν αισθητήρες (sensors). Οι συσκευές αυτές μοιράζονται δεδομένα και εξάγουν πληροφορίες για τη μέτρηση και χαρτογράφηση φαινομένων κοινού ενδιαφέροντος. Έχουν γίνει εξαιρετικά δημοφιλείς στοχεύοντας στην εξέλιξη του Διαδικτύου, αφού έχοντας ως βασικό εξοπλισμό πλήθος αισθητήρων αποτελούν σημαντικές πηγές συλλογής δεδομένων, ενώ λόγω της ασύρματης επικοινωνίας τους επιτρέπεται η άμεση παραγωγή πληροφορίας και διάδοσή της.

Η έννοια crowdsensing, είναι σύνθετη και προέρχεται από τις λέξεις crowd (πλήθος) και sensing (αίσθηση). Σύμφωνα με αυτή καθορίζεται εκείνη η διαδικασία μέσα από την οποία γίνεται συλλογή δεδομένων από αισθητήρες ενός πλήθους χρηστών με στόχο την παραγωγή μιας καθολικής εικόνας. Το πλήθος στις εφαρμογές αυτές είναι συνήθως κινητές συσκευές (mobile devices).

Οι Mobile CrowdSourcing εφαρμογές ταξινομούνται σε τρεις (3) μεγάλες κατηγορίες με βάση τον τύπο του φαινομένου που μελετάται, μετρείται ή χαρτογραφείται. Αυτές περιλαμβάνουν τις εφαρμογές που ασχολούνται με α) περιβαλλοντικά ζητήματα, β) ζητήματα που ασχολούνται με τις υποδομές και γ) κοινωνικά φαινόμενα. Στα επίπεδα που σχετίζονται με το περιβάλλον υπάρχουν ποικίλα παραδείγματα που περιλαμβάνουν τη μέτρηση των επιπέδων ρύπανσης σε μια πόλη, της στάθμης των υδάτων, και την παρακολούθηση της πανίδας οικοτύπων. Οι εφαρμογές αυτές επιτρέπουν τη χαρτογράφηση των διάφορων μεγάλων περιβαλλοντικών φαινομένων που αφορούν τον κοινό άνθρωπο.

Αναφορικά με τα ζητήματα υποδομών μπορεί να υπάρχουν εφαρμογές CrowdSourcing για την ανάλυση της κυκλοφοριακής κίνησης σε κεντρικές οδικές αρτηρίες κάνοντας χρήση δεδομένων από GPS. Τέλος, η τρίτη κατηγορία είναι οι κοινωνικές εφαρμογές, όπου τα άτομα μοιράζονται μέσω αισθητήρων πληροφορίες μεταξύ τους. Ως παράδειγμα, τα άτομα μπορούν να μοιράζονται τα δεδομένα της καθημερινής τους εξάσκησης και να τα συγκρίνουν με τα επίπεδα των υπολοίπων. Με τον τρόπο αυτό γίνεται μια καλύτερη αξιολόγηση των αποτελεσμάτων και μετρήσεων, συμβάλλοντας στη βελτίωση της καθημερινής τους άσκησης. Εφαρμογές που έχουν αναπτυχθεί για τον σκοπό αυτό είναι το BikeNet και το DietSense. Στην πρώτη εφαρμογή τα άτομα έχουν την δυνατότητα να αξιολογήσουν τη τοποθεσία και την ποιότητα διαδρομής με ένα ποδήλατο μέσα από την συλλογή δεδομένων για το σκοπό αυτό καθώς και τις εναλλακτικές διαδρομές βάσει των χαρακτηριστικών λήψεων που πραγματοποιούνται. Στην εφαρμογή που ονομάζεται DietSense, τα διαβητικά άτομα μπορούν να παρακολουθούν την διατροφή τους καθώς και να την συγκρίνουν με αυτήν που προέρχεται από άλλες ομάδες νοσούντων, έτσι ώστε να διατηρούν ισορροπημένες τις διατροφικές τους συνήθειες.

Συνοψίζοντας η λειτουργία των τυπικών εφαρμογών CrowdSourcing απεικονίζεται στην εικόνα 5, όπου ανεπεξέργαστα δεδομένα κάθε αισθητήρα που συλλέγονται μέσω των συσκευών αναλύονται τοπικά και στην συνέχεια, αφού ακολουθούν κατάλληλη επεξεργασία μέσω αλγορίθμων παράγουν νέα δεδομένα προς χρήση. Τα δεδομένα μπορεί στη συνέχεια να τροποποιηθούν και να αποσταλούν, διατηρώντας το απόρρητο των προσωπικών στοιχείων, στο κεντρικό επεξεργαστή για συνυπολογισμό και να γίνει εκ νέου διαμοίραση.



Εικόνα 5 Εφαρμογές CrowdSensing

2.5.2 ΜΟΝΑΔΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ MCS

Σύμφωνα με τον Merlino (2015) οι προκλήσεις από τις εφαρμογές CrowdSourcing είναι πολλές. Οι χρήστες τέτοιων εφαρμογών που θα είναι υπεύθυνοι στην λήψη των μετρήσεων πρέπει να κατανοήσουν πως αυτές πρέπει να πραγματοποιούνται σε πραγματικό χρόνο (real time), ώστε η σύνθεση της διαμόρφωσης του σήματος να είναι έγκυρη και επίκαιρη.

Επιπρόσθετα, λόγω της έντονης αύξησης των διαφόρων τύπων λογισμικού που παρατηρείται οι διάφορες κινητές συσκευές, που λειτουργούν ως συλλέκτες, αποκτούν αυτόματα πολυπλοκότητα. Επιπλέον, οι MCS εφαρμογές δεν έχουν την δυνατότητα της παράλληλης χρήσης κοινών πόρων μεταξύ τους, παρότι πολλές λειτουργίες των MCS, όπως η συλλογή και αποστολή μετρήσεων, είναι παρόμοιες και θα μπορούσαν να ομαδοποιηθούν. Η συνεχής συλλογή μετρήσεων από την crowdsensing εφαρμογή, έχει επίπτωση σε βασικούς πόρους της κινητής συσκευής του χρήστη, όπως είναι η μπαταρία και η μνήμη. Ο περιορισμός της υπερβολικής χρήσης έγκειται στην προσεκτική δομημένη σχεδίαση του λογισμικού και της εφαρμογής ώστε να έχει την δυνατότητα να παρακολουθεί και κατάλληλα να εστιάζει στο πρόβλημα χειρισμού, υπενθυμίζοντας και προστατεύοντας τον εκάστοτε χρήστη.

Τα τελευταία χρόνια, για την αποφυγή αυτών των προβλημάτων και την ανάδειξη των σημαντικών χαρακτηριστικών μιας MCS εφαρμογής έχουν δημιουργηθεί κατάλληλες πλατφόρμες όπως η Medusa και η Vita, οι οποίες ως σημεία επαφής αναλαμβάνουν να

ενεργούν μεταξύ των χρηστών και των εφαρμογών Crowdsourcing. Έτσι, βασικά χαρακτηριστικά των εν λόγω πλατφορμών είναι:

1. Ο εντοπισμός των απαραίτητων χρηστών και η ανάθεση της συλλογής των μετρήσεων που χρειάζεται η εφαρμογή
2. Η διαθεσιμότητα των ενδιαφερομένων χρηστών που ελέγχεται σε τακτά χρονικά διαστήματα από την ίδια την πλατφόρμα
3. Η ευχέρεια στις εφαρμογές MCS να την χρησιμοποιούν ώστε να αποκτούν έλεγχο των ενεργών χρηστών για την παροχή των μετρήσεων που χρειάζονται
4. Η εγκατάσταση των απαραίτητων παραμέτρων και των πολιτικών στους διαθέσιμους χρήστες που διαθέτουν την MCS εφαρμογή
5. Η προσφορά ικανοποιητικού επιπέδου ποιότητας παρεχόμενων υπηρεσιών που έχει συμφωνηθεί με τις εφαρμογές MCS .

ΚΕΦΑΛΑΙΟ 3: ΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ

3.1 ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ CLOUD

Σε όλη την ιστορία της επιστήμης των υπολογιστών, πολλές προσπάθειες έχουν γίνει για να αποδεσμευτούν οι χρήστες από τις υλικές ανάγκες ενός υπολογιστικού συστήματος. Το υπολογιστικό νέφος, είναι ένα καινοτόμο πληροφοριακό σύστημα, το οποίο φαίνεται πλέον να είναι το μέλλον του computing, ανατρέπει όσα γνωρίζαμε για τα λειτουργικά συστήματα, τα πελάτη/εξυπηρετητή συστήματα και τους περιηγητές διαδικτύου.

Απελευθερώνοντας τους χρήστες από τις ανάγκες του υλικού, το cloud μειώνει τις λοιπές ανάγκες των πελατών και τις πολυπλοκότητες των συστημάτων. Το cloud computing, πήρε το όνομα του από το σύμβολο του σύννεφου που χρησιμοποιείται συχνά για να αντιπροσωπεύει το διαδίκτυο σε διάφορων ειδών διαγράμματα. Μια εμφανής στροφή προς τα συστήματα cloud λάμβανε χώρα τα τελευταία χρόνια με τους τελευταίους χρήστες, σταδιακά διατηρούν ένα αυξανόμενο αριθμό προσωπικών δεδομένων, που περιλαμβάνει σελιδοδείκτες, φωτογραφίες, αρχεία μουσικής και πολλά άλλα σε απομακρυσμένους servers προσπελάσιμους μέσω του διαδικτύου.

Το cloud computing έχει ενισχυθεί από την τεχνολογία υλοποίησης, αλλά για πολύ καιρό ήταν διαθέσιμη μόνο σε συστήματα mainframe. Στην πεμπουσία του, ο υπολογιστής εκτελεί μια εφαρμογή γνωστή ως υπερεπόπτης. Έτσι, δημιουργούνται μία ή περισσότερες εικονικές μηχανές, οι οποίες προσομοιώνουν φυσικούς υπολογιστές τόσο πιστά, που οι προσομοιώσεις μπορεί να τρέξουν οποιοδήποτε λογισμικό, από λειτουργικά συστήματα, σε εφαρμογές των τελικών χρηστών.

Σε επίπεδο υλικού, ένας αριθμός φυσικών συσκευών, συμπεριλαμβανομένων των επεξεργαστών, σκληρών δίσκων και συσκευών δικτύου, βρίσκονται σε κέντρα δεδομένων, ανεξάρτητα από τη γεωγραφική θέση, τα οποία είναι υπεύθυνα για την αποθήκευση και τις επεξεργαστικές ανάγκες. Πέρα από αυτό, ο συνδυασμός των στρωμάτων λογισμικού, το στρώμα εικονικοποίησης και το στρώμα διαχείρισης, επιτρέπουν την αποτελεσματική διαχείριση των servers.

Η εικονικοποίηση είναι ένα βασικό στοιχείο στις υλοποιήσεις του cloud και χρησιμοποιείται για να παρέχει ζωτικά χαρακτηριστικά της ανεξαρτησίας θέσης, της συγκέντρωσης πόρων και γρήγορης ελαστικότητας. Διαφέρει από τις παραδοσιακές τοπολογίες δικτύου όπως client-server, το cloud computing είναι σε θέση να προσφέρει αντοχή και να αποτρέψει προβλήματα της κυκλοφοριακής συμφόρησης. Το στρώμα διαχείρισης είναι σε θέση να παρακολουθεί την κυκλοφορία και να ανταποκρίνεται σε κορυφές ή καμπύλες με τη δημιουργία νέων εξυπηρετητών ή την καταστροφή των μη απαραίτητων. Το στρώμα διαχείρισης έχει την πρόσθετη ικανότητα να μπορεί να εφαρμόσει την παρακολούθηση και τους κανόνες ασφάλειας σε όλο το cloud. Σύμφωνα με τον Merrill Lynch, αυτό που κάνει το cloud computing ξεχωριστό, είναι η εικονικοποίηση.

3.2 ΙΣΤΟΡΙΚΗ ΕΞΕΛΙΞΗ

Γενικά αναφέρουμε ότι η εξέλιξη και η ανάπτυξη της έννοιας Cloud χωρίζεται σε τρεις βασικές χρονικές περιόδους.

1. 70'-80' εποχή η οποία χαρακτηρίζεται σαν mainframe
2. 80'-2000 εποχή η οποία αναφέρεται ως client-server
3. 2000-2010 – έως και σήμερα η εποχή η οποία χαρακτηρίζεται

Τα χαρακτηριστικά της πρώτης χρονικής περιόδου είναι:

- ο Εικονικοποίηση (virtualization)
- ο Δεν υπάρχει επικοινωνία προς τα έξω c) Έχουμε συγκεντρωτική πληροφορική

Τα χαρακτηριστικά της δεύτερης χρονικής περιόδου είναι:

- ο Κατανεμημένη επεξεργασία
- ο Το hardware γίνεται πιο ισχυρό
- ο Συνειδητοποίηση ότι υπάρχουν αρκετοί υπολογιστές στον κόσμο

Τα χαρακτηριστικά της τρίτης χρονικής περιόδου είναι:

- ο Μεγάλη διαθεσιμότητα υπηρεσιών
- ο Μεγάλο εύρος ζώνης
- ο Μεγάλη διαθεσιμότητα
- ο Μεγάλα κέντρα δεδομένων
- ο Το υλικό παρέχεται σε λογικές τιμές

3.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ CLOUD COMPUTING

Με τον όρο αρχιτεκτονική του Cloud Computing αναφερόμαστε στα στοιχεία και στα συστατικά εκείνα τα οποία είναι ικανά να συνθέσουν/αποτελέσουν ένα Cloud Computing. Τα συστατικά εκείνα τα οποία συνθέτουν ένα Cloud Computing είναι μία front-end πλατφόρμα (υπολογιστής, φορητή συσκευή, τερματικό), μία back-end πλατφόρμα (server), μία διανομή βασισμένη σε Cloud υπηρεσίες, ένα δίκτυο (internet, intranet). Όλα τα παραπάνω συνθέτουν την αρχιτεκτονική ενός Cloud Computing. Με τον όρο ενδοδιαδίκτυο αναφερόμαστε σε ένα δίκτυο υπολογιστών το οποίο κάνει χρήση διαφόρων ιντερνετικών πρωτοκόλλων προκειμένου να μοιράζονται πληροφορίες, λειτουργικά συστήματα ή computer services μέσα σε έναν οργανισμό.

Το cloud computing αξιοποιεί την εικονικοποίηση ώστε να μεγιστοποιείται η υπολογιστική ισχύς. Η εικονικοποίηση, διαχωρίζοντας το λογικό από το φυσικό, επιλύει μερικές από τις προκλήσεις που αντιμετωπίζει η τεχνολογία του computing. Ενώ το Grid Computing επιτυγχάνει υψηλή χρησιμοποίηση μέσω της κατανομής πολλαπλών servers σε ένα ενιαίο έργο ή εργασία, η εικονικοποίηση των servers στο cloud computing επιτυγχάνει υψηλή χρήση, επιτρέποντας σε έναν εξυπηρετητή να εκτελεί πολλές εργασίες ταυτόχρονα.

Ενώ οι περισσότεροι συγγραφείς αναγνωρίζουν ομοιότητες μεταξύ των δύο παραδειγμάτων, οι απόψεις φαίνεται να συγκεντρώνονται γύρω από το διαπίστωση ότι το cloud computing έχει εξελιχθεί από το Grid Computing και ότι το Grid Computing είναι η βάση για το cloud computing.

Στο cloud computing, τα διαθέσιμα μοντέλα υπηρεσιών είναι :

Infrastructure as a Service (IaaS): Είναι ένα μοντέλο διάταξης στο οποίο ένας οργανισμός (organization) διαθέτει τον εξοπλισμό του προκειμένου να καλύψει ανάγκες χρηστών και εταιριών όπως, στήριξη ενεργειών, αποθήκευση, hardware, servers και εξαρτήματα δικτύωσης. Ο πάροχος αυτών των υπηρεσιών είναι υπεύθυνος για την λειτουργία, την συντήρηση-διατήρηση και την στέγαση αυτού του εξοπλισμού. Ο χρήστης συνήθως πληρώνει αυτές τις υπηρεσίες που λαμβάνει σε σχέση με μία ανά-χρήστη βάση.

Βασικά χαρακτηριστικά και συνιστώσες του IaaS είναι:

1. Utility computing service και μοντέλο χρέωσης
2. Αυτοματοποίηση των διοικητικών καθηκόντων
3. Δυναμική κλιμάκωση
4. Desktop virtualization
5. Εφαρμογές που δίνουν ιδιαίτερη βάση στην πολιτική
6. Σύνδεση στο Διαδίκτυο
7. Σημαντικό πλεονέκτημα του IaaS είναι η δυνατότητα μεταφοράς εικονικών μηχανημάτων από το ιδιόκτητο περιβάλλον της εταιρίας ή του ιδιώτη στο cloud με απλές διαδικασίες.

Platform as a Service (PaaS): Είναι ένας τρόπος για να νοικιάζεται υλικό, λειτουργικά συστήματα, αποθήκευση και χωρητικότητα δικτύου πάνω στο internet. Αυτό το μοντέλο παροχής υπηρεσιών επιτρέπει τον πελάτη να νοικιάσει virtualized servers και συναφείς υπηρεσίες για την εκτέλεση υπαρχόντων εφαρμογών ή την ανάπτυξη και έλεγχο νέων. Επίσης, είναι αποτέλεσμα του Software as a Service (SaaS), ένα μοντέλο διανομής λογισμικού στο οποίο φιλοξενούνται εφαρμογές λογισμικού που διατίθενται στους πελάτες μέσω διαδικτύου. Το PaaS έχει πολλά πλεονεκτήματα για τους προγραμματιστές. Με το PaaS, τα χαρακτηριστικά του λειτουργικού συστήματος μπορούν να αλλάζουν και να αναβαθμίζονται συχνά. Επιπλέον, γεωγραφικά κατανομημένες ομάδες ανάπτυξης λογισμικού μπορούν να συνεργαστούν σε διάφορα έργα ανάπτυξης λογισμικού. Επιπρόσθετα, το κόστος χρήσης μπορεί να μειωθεί με τη χρήση των υπηρεσιών υποδομής από έναν μόνο προμηθευτή, αντί να διατηρούνται πολλαπλές εγκαταστάσεις του υλικού που συχνά εκτελούν διπλές λειτουργίες ή υποφέρουν από προβλήματα ασυμβατότητας.

Στον αντίποδα, το PaaS επιφέρει κάποιο κίνδυνο "lock-in", αν οι προσφορές απαιτούν κατάλληλες υπηρεσίες interface ή ανάπτυξη γλωσσών. Ένα άλλο μειονέκτημα είναι ότι η ευελιξία των προσφορών που παρέχονται δεν είναι αρκετές για να καλύψουν τις ανάγκες ορισμένων χρηστών οι οποίες εξελίσσονται ταχύτατα. Το PaaS βασίζεται στο μοντέλο 'Pay-Per-Use' με τέτοιο τρόπο, ώστε να επιτυγχάνεται η πλήρης αξιοποίηση των υπολογιστικών πόρων που χρησιμοποιούνται σε σχέση με το κόστος χρήσης.

Software as a Service (SaaS): Το SaaS βασίζεται στη λογική της υπενοικίασης λογισμικού από έναν πάροχο, αντί της αγοράς της άδειας χρήσης. Αυτό σημαίνει ότι το λογισμικό βρίσκεται σε ένα δίκτυο servers προκειμένου να διατίθεται ως υπηρεσία από το Διαδίκτυο. Επίσης καλείται και 'software on demand' και αποτελεί τον πιο γνωστό τύπο cloud computing. Δηλαδή, ο

πάροχος της υπηρεσίας φιλοξενεί και την εφαρμογή και τα δεδομένα, οπότε με οποιοδήποτε δυνατό τρόπο οι χρήστες μπορούν να τη χρησιμοποιήσουν.

Το μεγαλύτερο πλεονέκτημα αυτής της υπηρεσίας και ίσως το πιο χρήσιμο προς τον χρήστη, είναι η μείωση του κόστους. Είναι πολύ πιο οικονομική η ενοικίαση μιας εφαρμογής για κάποιο χρονικό διάστημα παρά η αγορά της. Επίσης, ένα άλλο μεγάλο πλεονέκτημα είναι ότι δεν απαιτείται από την πλευρά των χρηστών συντήρηση και αναβάθμιση του λογισμικού, καθώς γι' αυτές τις λειτουργίες υπεύθυνος είναι ο πάροχος. Σαν αποτέλεσμα, δε θα είναι απαραίτητη η αγορά αυθεντικού λογισμικού, γιατί το Cloud Computing παρέχει τη δυνατότητα ενοικιάσής του για όσο χρόνο το χρησιμοποιούν οι χρήστες.

3.4 ΒΑΣΙΚΑ ΜΟΝΤΕΛΑ ΑΝΑΠΤΥΞΗΣ

Τα τέσσερα μοντέλα ανάπτυξης που έχουν προσδιοριστεί για επίλυση προβλημάτων αρχιτεκτονικής cloud, περιγράφονται παρακάτω :

- **Ιδιωτικό cloud:** Το cloud λειτουργεί για μια ιδιωτική οργάνωση. Μπορεί να διαχειρίζεται από την οργάνωση ή από μια ομάδα τρίτων, και μπορεί να ακολουθεί ή όχι τις προδιαγραφές.
- **Community cloud:** Το cloud μοιράζεται από διάφορους οργανισμούς και υποστηρίζει μια συγκεκριμένη κοινότητα που έχει κοινές ανησυχίες (π.χ., αποστολή, απαιτήσεις ασφάλειας, πολιτική, και κανόνες συμμόρφωσης). Μπορεί να διαχειρίζεται από τις οργανώσεις ή από μια ομάδα τρίτων, εντός κι εκτός προδιαγραφών.
- **Public cloud:** Το cloud είναι στη διάθεση του ευρέος κοινού ή μιας μεγάλης βιομηχανία και ανήκει σ' ένα οργανισμό που πουλά τις υπηρεσίες cloud.
- **Hybrid cloud:** Το υβριδικό cloud είναι μια σύνθεση από δύο ή περισσότερα cloud (ιδιωτικών, κοινοτικών ή δημόσιων) που παραμένουν μοναδικές οντότητες, αλλά συνδέονται μεταξύ τους με τυποποιημένη ή ιδιόκτητη τεχνολογία, που επιτρέπει φορητότητα δεδομένων κι εφαρμογή (π.χ., το cloud να καταρρέει από την εξισορρόπηση φορτίου μεταξύ των clouds).

Το cloud computing θεωρείται ως μία από τις πλέον υποσχόμενες τεχνολογίες στην πληροφορική σήμερα, εγγενώς σε θέση να αντιμετωπίσει μια σειρά από ζητήματα. Έχουν ταυτοποιηθεί μια σειρά από βασικά χαρακτηριστικά του cloud computing:

Ευελιξία / Ελαστικότητα: οι χρήστες μπορούν γρήγορα να έχουν πρόσβαση στους πόρους υπολογιστών, όσο χρειάζονται χωρίς την ανθρώπινη αλληλεπίδραση.

Επεκτασιμότητα του cloud: νέοι κόμβοι μπορεί να προστεθούν ή να αφαιρεθούν από το δίκτυο όπως οι φυσικοί εξυπηρετητές, με περιορισμένες τροποποιήσεις στην εγκατάσταση της αρχιτεκτονικής και στο λογισμικό. Το Cloud μπορεί να τεθεί σε οριζόντια ή κάθετη κλίμακα, αναλόγως.

Ευρεία πρόσβαση στο δίκτυο : Οι δυνατότητες είναι διαθέσιμες μέσω του δικτύου και είναι προσπελάσιμες μέσω τυποποιημένων μηχανισμών που προωθούν την χρήση από ετερογενείς πλατφόρμες (π.χ. κινητά τηλέφωνα, φορητούς υπολογιστές, και PDAs) .

Ανεξαρτησία Τοποθεσίας: Υπάρχει μια αίσθηση ανεξαρτησίας της θέσης, στο ότι ο πελάτης δεν έχει γενικά κανένα έλεγχο ή γνώση σχετικά με την ακριβή τοποθεσία των παρεχόμενων πόρων, αλλά μπορεί να είναι σε θέση να προσδιορίζει τη θέση σε υψηλότερο επίπεδο αφαίρεσης (π.χ. χώρα, κράτος ή datacenter) .

Αξιοπιστία: βελτιώνεται μέσω της χρήσης πολλαπλών ιστότοπων, που κάνουν το cloud computing κατάλληλο για την ομαλή λειτουργία των εταιριών και ανάκτηση δεδομένων από τυχόν απώλειες δεδομένων από καταστροφές.

Οικονομία κλίμακας κι αποτελεσματικότητα του κόστους: Οι Cloud εφαρμογές, ανεξάρτητα από το μοντέλο ανάπτυξης, τείνουν να είναι όσο πιο μεγάλες γίνεται, προκειμένου να επωφεληθούν από τις οικονομίες κλίμακας. Μεγάλα συστήματα cloud μπορεί συχνά να βρίσκονται κοντά σε φτηνούς σταθμούς παροχής ενέργειας και να εγκαθίστανται σε φθηνά ακίνητα, για να μειωθεί το κόστος .

Βιωσιμότητα: προκύπτει μέσω της βελτιωμένης χρήσης των πόρων, πιο αποτελεσματικά συστήματα.

3.5 ΑΣΦΑΛΕΙΑ ΣΤΟ ΥΠΟΛΟΓΙΣΤΙΚΟ ΝΕΦΟΣ

Οι εφαρμογές που φιλοξενούνται στο Υπολογιστικό Νέφος συχνά περιέχουν προηγμένες τεχνολογίες ασφάλειας, κι είναι κυρίως διαθέσιμες λόγω της συγκέντρωσης των δεδομένων και της καθολικής αρχιτεκτονικής. Η ομογένειες του πόρους που χαρακτηρίζει τη φύση του cloud, επιτρέπει στους παρόχους cloud, να εστιάσουν όλους τους πόρους της ασφάλειάς πάνω στην ασφάλεια του cloud. Την ίδια στιγμή, οι δυνατότητες αυτοματοποίησης του cloud σε συνδυασμό με τους βαθύτερα εστιασμένους πόρους ασφάλειας, συνήθως οδηγούν σε προηγμένες ικανότητες ασφάλειας. Η διορατικότητα είναι απαραίτητη σε έναν τομέα που εξελίσσεται ραγδαία. Το cloud computing δεν είναι πανάκεια και πολλοί πιστεύουν ότι είναι μια καμπάνια αποκλειστικά καθοδηγούμενη από εμπορικούς παράγοντες. Η επιφυλακτικότητα είναι αναγκαία, έτσι για να μην παρασυρθούμε από την ιδιοτροπία της στιγμής. Το cloud computing κάθε αυτό έχει τη δυνατότητα να αντιμετωπίσει μια σειρά ελλείψεων των παραδοσιακών αρχιτεκτονικών χάρη στα μοναδικά χαρακτηριστικά του, ωστόσο η υιοθέτηση αυτής της πρωτοποριακής αρχιτεκτονικής του μπορεί να συνεπάγεται μια σειρά από επιπλέον άγνωστες απειλές.

ΚΕΦΑΛΑΙΟ 4: ΤΕΧΝΟΛΟΓΙΕΣ ΥΛΟΠΟΙΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΒΑΣΙΣΜΕΝΗΣ ΣΤΟΝ ΙΣΤΟ ΚΑΙ ΤΗΣ ANDROID ΕΦΑΡΜΟΓΗΣ

Για τη υλοποίηση της εφαρμογής στον ιστό επιλέξαμε να εργαστούμε σε λειτουργικό Linux, Debian όπου έχουμε και στο Server στον οποίο ανεβάσαμε το πρόγραμμα. Εκτός από τις απαραίτητες τεχνολογίες για την κατασκευή της ιστοσελίδας όπως την HTML θα αναφέρουμε συγκεκριμένα όλα τα απαραίτητα τεχνολογικά μέσα και εργαλεία που χρησιμοποιήσαμε κατά τη διάρκεια της εργασίας.

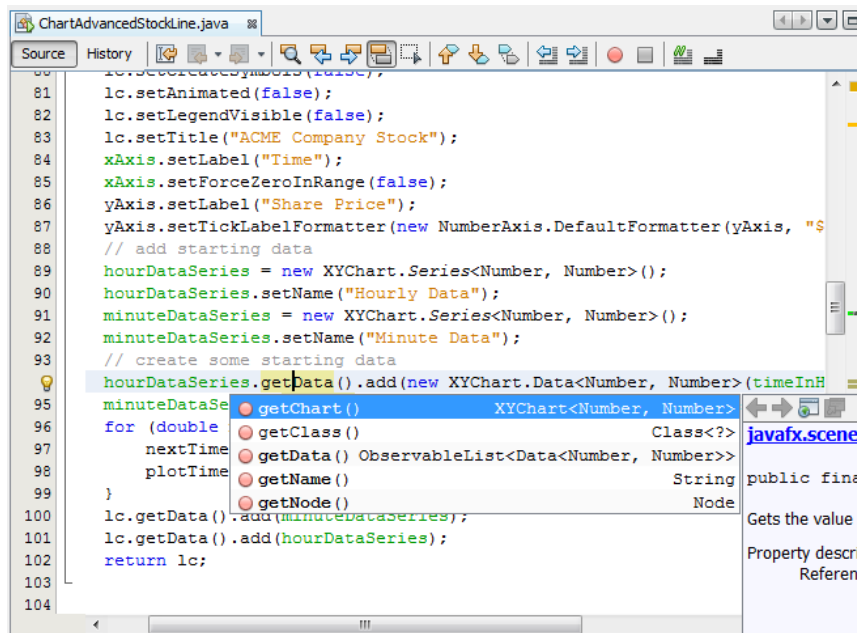
4.1 NETBEANS IDE

Εργαστήκαμε στο NetBeans IDE (Integrated Development Environment) το οποίο προσφέρεται σε συνδυασμό με τον Glassfish Server. Τα δύο αυτά προϊόντα ανήκουν στην Oracle, κάτι που διευκολύνει τον προγραμματισμό με κώδικα Java. Οι περισσότεροι προγραμματιστές αναγνωρίζουν το NetBeans IDE ως το γνήσιο δωρεάν IDE για προγραμματισμό με Java. Ωστόσο, το NetBeans IDE υποστηρίζει και πολλές άλλες γλώσσες, όπως PHP, C / C ++ και JavaScript και πλαίσια (frameworks).

Το NetBeans αποτελεί προϊόν ανάπτυξης λογισμικού (NetBeans IDE και πλατφόρμα NetBeans) το οποίο καλύπτει τις ανάγκες των προγραμματιστών, των χρηστών και των επιχειρήσεων που βασίζονται σε αυτό ως βάση για τα προϊόντα τους. Δίνει τη δυνατότητα εύκολης, γρήγορης και αποτελεσματικής ανάπτυξης προϊόντων. Το NetBeans χορηγείται από την Sun Microsystems και προσφέρεται με ανοιχτό κώδικα, από τον Ιούνιο του 2000 μέχρι τον Ιανουάριο του 2010 που η εταιρεία αγοράστηκε από την Oracle, η οποία εξακολούθησε να το προσφέρει με ανοιχτό κώδικα στο κοινό. Το γεγονός αυτό δίνει το πλεονέκτημα ότι μπορεί να τροποποιηθεί και να διαμοιραστεί αφού ο σχεδιασμός του είναι προσιτός στο κοινό. Τα δύο προϊόντα βάσης, το NetBeans IDE και το NetBeans Platform, διατίθενται δωρεάν για εμπορική και μη εμπορική χρήση. Ο πηγαίος κώδικας και των δύο είναι διαθέσιμος για επαναχρησιμοποίηση από οποιονδήποτε, φτάνει να συμφωνεί με τους όρους χρήσης. Η κοινότητα του NetBeans αποτελείται από ανθρώπους από όλο τον κόσμο οι οποίοι μπορούν να κάνουν ερωτήσεις, να δώσουν συμβουλές, να συμβάλλουν και εν τέλει να διαδώσουν τα επιτυχημένα προϊόντα της εταιρείας. Σήμερα τα downloads του NetBeans IDE φτάνουν τα 18 εκατομμύρια και χρησιμοποιείται για την ανάπτυξη προγραμμάτων και εφαρμογών, από πέραν των 800.000 προγραμματιστών κατατάσσοντας το έργο της NetBeans σε μια σημαντική θέση στο χώρο της ανάπτυξης λογισμικού.

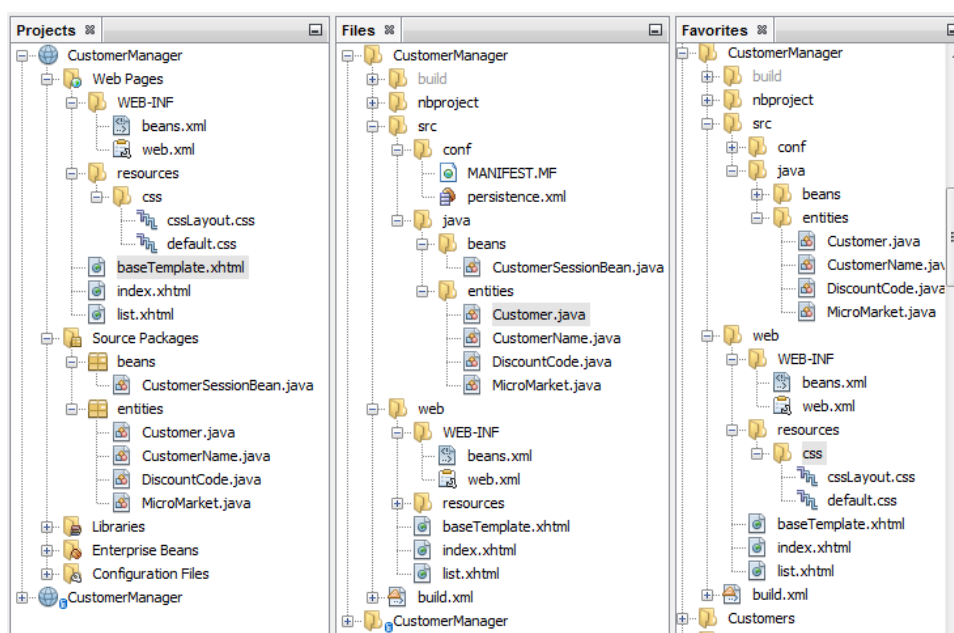
Το NetBeans IDE προσφέρει ακριβώς όσα χρειαστήκαμε για τη δημιουργία του web application. Όπως ειπώθηκε πιο πάνω, επιτρέπει τη γρήγορη και εύκολη ανάπτυξη εφαρμογών Java σε desktop, κινητά (mobile) και web, καθώς και εφαρμογές HTML5 με την HTML, JavaScript και CSS. Το IDE είναι δωρεάν και open source και έχει μια μεγάλη κοινότητα χρηστών και προγραμματιστών από όλο τον κόσμο. Προσφέρει την καλύτερη υποστήριξη για τις τελευταίες τεχνολογίες Java. Με τους συντάκτες (editors), τους αναλυτές κώδικα (code analyzers) και τους μετατροπείς (converters) που συμπεριλαμβάνει, επιτρέπει την αναβάθμιση της εφαρμογής έτσι ώστε να καθίσταται δυνατή η χρησιμοποίηση των νέων γλωσσικών δομών Java 8, λειτουργικών

δραστηριοτήτων, και αναφορών μεθόδου. Όπως και τα περισσότερα IDE έτσι και το NetBeans IDE, αφήνει αυτόματα γραμμές μεταξύ εντολών εκεί που πρέπει, ταιριάζει λέξεις με παρενθέσεις, και χρωματίζει εντολές πηγαίου κώδικα συντακτικά και σημασιολογικά. Παρέχει επίσης πρότυπα κώδικα, συμβουλές κωδικοποίησης και γεννήτριες κώδικα.



Εικόνα 6 Εμφάνιση όλων των συναρτήσεων μιας κλάσης στο περιβάλλον NetBeans

Ο σχηματισμός μιας σαφούς εικόνας σε μεγάλες εφαρμογές που περιέχουν χιλιάδες φακέλους και αρχεία, και εκατομμύρια γραμμές κώδικα, είναι ένα δύσκολο έργο. Το NetBeans IDE παρέχει κατάλληλες προβολές των δεδομένων μιας εφαρμογής, από πολλαπλά παράθυρα του project με χρήσιμα εργαλεία για πιο αποτελεσματική διαχείριση, επιτρέποντας στο χρήστη να ανατρέξει στα δεδομένα του εύκολα και γρήγορα.



Εικόνα 7 Εργαλεία Εμφάνισης των στοιχείων μιας Εφαρμογής στο περιβάλλον NetBeans

Η δυνατότητα αυτή διευκολύνει νέους προγραμματιστές να συμμετάσχουν στη δημιουργία μιας εφαρμογής, αφού μπορούν να κατανοήσουν τη δομή του προγράμματος, εξαιτίας του καλά οργανωμένου κώδικα. Πολύ σημαντικό στην κατασκευή μιας εφαρμογής είναι το Γραφικό περιβάλλον χρήστη ή γραφική διασύνδεση/διεπαφή χρήστη (Graphical User Interface - GUI). GUI καλείται το σύνολο των γραφικών στοιχείων, τα οποία εμφανίζονται στην οθόνη κάποιας ψηφιακής συσκευής και χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με τη συσκευή αυτή. Το NetBeans IDE είναι κατάλληλο για το σχεδιασμό Γραφικού Περιβάλλοντος Χρήστη (GUI) για εφαρμογές σε Java SE, Java EE, αλλά και άλλες γλώσσες προγραμματισμού, γρήγορα και ομαλά με τη χρήση των editors και εργαλείων. Για εφαρμογές Java SE, η NetBeans GUI Builder φροντίζει αυτόματα τη σωστή απόσταση και ευθυγράμμιση κώδικα.

Για τη δημιουργία της εφαρμογής μας, χρησιμοποιήσαμε την τεχνολογία **Servlet** η οποία δίνει τη δυνατότητα να ανταλλάσσονται δεδομένα με τον πελάτη (client) της εφαρμογής ιστού, κάτι το οποίο μας είναι απαραίτητο. Η λειτουργία του Servlet θα περιγραφεί στη συνέχεια αυτού του κεφαλαίου.

4.2 GLASSFISH SERVER

Ο Glassfish Server Open Source Edition 4.1.1 είναι ένας ανοικτού κώδικα διακομιστής εφαρμογών που ξεκίνησε από την Sun Microsystems για την πλατφόρμα της Java EE και χορηγείται πλέον από την Oracle. Η υποστηριζόμενη έκδοση ονομάζεται Oracle GlassFish Server.

Ο GlassFish είναι ελεύθερο λογισμικό με διπλή άδεια: την Common Development and Distribution License (CDDL) και την GNU General Public License (GPL). Ο GlassFish υλοποιεί αναφορές της Java EE και υποστηρίζει Servlets, web services, Enterprise JavaBeans, JPA, JavaServer Faces, JMS, RMI, JavaServer Pages, κλπ. Αυτό επιτρέπει στους developers να δημιουργούν εφαρμογές φορητές και επεκτάσιμες, οι οποίες μπορούν να ενσωματώσουν τεχνολογίες. Πολλοί προγραμματιστές οι οποίοι έχουν εξοικειωθεί στη Java SE/Java EE επιλέγουν GlassFish για την υλοποίηση εφαρμογών Java. Σημαντικό πλεονέκτημα είναι ότι ο GlassFish Server υποστηρίζει της ενημερώσεις των εκδόσεων Java EE/Java SE, πολύ πιο σύντομα από τους υπόλοιπους.

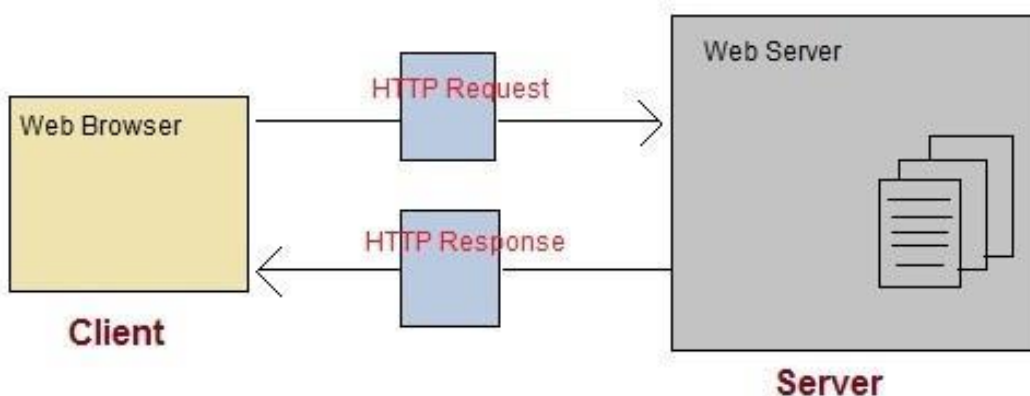
4.3. JAVA SE 8

Αρχικά, για να μπορέσουμε να τρέξουμε προγράμματα Java, πρέπει να έχουμε εγκαταστήσει στον υπολογιστή μας (και εν ακολουθία στο Server μας) ένα περιβάλλον εκτέλεσης Java (Java Runtime Environment - JRE), το οποίο μετατρέπει όλα τα αρχεία με την κατάληξη .java σε εκτελέσιμα. Το JDK περιέχει όλα τα απαραίτητα εργαλεία της Java, που χρειάζεται ένας προγραμματιστής, έτσι ώστε να αναπτύξει ένα web application. Το JRE είναι ένα υποσύνολο του JDK. Το υλικό για την εγκατάσταση του JDK βρίσκεται στην επίσημη ιστοσελίδα της Oracle. Εμείς κατεβάσαμε και εγκαταστήσαμε την τελευταία έκδοση η οποία ήταν η Java SE 8.

4.4 SERVLET

Για την δημιουργία της εφαρμογής ιστού χρησιμοποιήσαμε την τεχνολογία των Servlets. Το διαδίκτυο αποτελείται από δισεκατομμύρια πελάτες και εξυπηρετητές οι οποίοι επικοινωνούν μεταξύ τους μέσω ενσύρματων και ασύρματων δικτύων. Οι πελάτες κάνουν αιτήσεις στους Web Servers, ο Web Server λαμβάνει το αίτημα, βρίσκει τους πόρους και επιστρέφει την απάντηση στον πελάτη. Πιο συγκεκριμένα, ο πελάτης χρησιμοποιώντας ένα browser στέλνει την αίτηση στο Server, στη συνέχεια ο Server ανταποκρίνεται στο browser επιστρέφοντας ένα σύνολο οδηγιών γραμμένο σε HTML (HyperText Markup Language). Ο browser ερμηνεύει τις οδηγίες αυτές και εμφανίζει το περιεχόμενο στο πελάτη.

Προτού εξηγήσουμε το ρόλο ενός Servlet στη διαδικασία που περιγράψαμε πιο πάνω, πρέπει πρώτα να εξηγήσουμε το ρόλο ενός σημαντικού παράγοντα της διαδικασίας ανταλλαγής δεδομένων μεταξύ πελάτη και διακομιστή. Αυτός είναι, το πρωτόκολλο HTTP (Hypertext Transfer Protocol) που ακολουθούν οι πελάτες και οι διακομιστές για να επικοινωνούν μεταξύ τους. Το πρωτόκολλο HTTP είναι παρόμοιο με άλλα πρωτόκολλα του Διαδικτύου όπως με τα SMTP (Simple Mail Transfer Protocol) και FTP (File Transfer Protocol), αλλά υπάρχει μια θεμελιώδης διαφορά. Σε αντίθεση με τα υπόλοιπα, το πρωτόκολλο HTTP χαρακτηρίζεται ως stateless, αυτό σημαίνει ότι υποστηρίζει μόνο μία αίτηση ανά σύνδεση, δηλαδή οι πελάτες συνδέονται με το διακομιστή για να στείλουν ένα αίτημα και αμέσως μετά αποσυνδέονται. Ο τρόπος λειτουργίας του, επιτρέπει τη σύνδεση περισσότερων χρηστών με ένα διακομιστή σε ένα δεδομένο χρονικό διάστημα. Όπως φαίνεται στην εικόνα 8, ο πελάτης στέλνει ένα αίτημα HTTP και ο διακομιστής απαντά με μια σελίδα HTML στον πελάτη, χρησιμοποιώντας HTTP.



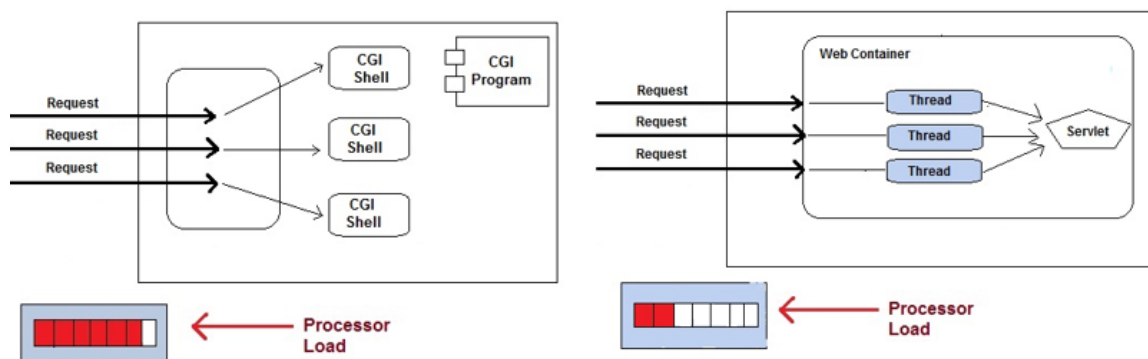
Εικόνα 8 Πρωτόκολλο Client- Server

Η τεχνολογία των Servlet χρησιμοποιείται για τη δημιουργία web application και χρησιμοποιεί τη γλώσσα προγραμματισμού Java. Πριν από τα Servlets, επικρατούσε μια άλλη τεχνολογία στον τομέα της κατασκευής των εφαρμογών ιστού και ονομαζόταν CGI (Common Gateway Interface). Με την τεχνολογία των CGI, ο χρήστης κάνει κλικ σε μια διεύθυνση (link) της οποίας το URL (Uniform Resource Locator) παραπέμπει σε μια δυναμική ιστοσελίδα αντί σε μια στατική. Η διεύθυνση URL αποφασίζει ποιο πρόγραμμα CGI να εκτελέσει όπου το πρόγραμμα τρέχει σε κέλυφος του λειτουργικού συστήματος (OS shell). Τα αποτελέσματα του CGI αποστέλλονται

πίσω στο Web Server, ο οποίος τα πακετάρει σε μια απόκριση HTTP η οποία στέλνεται πίσω στον web browser. Η λειτουργία αυτή ήταν χρονοβόρα αφού τα προγράμματα CGI εκτελούνταν σε δικό τους κέλυφος OS. Επιπλέον δεν είχαν τη δυνατότητα επέκτασης και τα προγράμματα CGI δεν ήταν ούτε ασφαλή ούτε αντικειμενοστραφή.

Εξαιτίας αυτών των μειονεκτημάτων, οι προγραμματιστές άρχισαν να ψάχνουν για καλύτερες λύσεις. Στη συνέχεια, η Sun Microsystems ανέπτυξε την τεχνολογία των Servlet την οποία προώθησε ως εναλλακτική λύση. Η μετάβαση στη τεχνολογία αυτή έγινε πολύ γρήγορα εξαιτίας των βασικών πλεονεκτημάτων που παρείχε, όπως:

- Λιγότερο χρόνο απόκρισης διότι κάθε αίτημα τρέχει σε ξεχωριστό νήμα.
- Η τεχνολογία των Servlets είναι επεκτάσιμη.
- Τα Servlets είναι object-oriented.
- Δεν εξαρτώνται από την πλατφόρμα.



Εικόνα 9 Μείωση του φόρτου του επεξεργαστή με Servlet

Λειτουργία Servlet

Το αίτημα HTTP από τον πελάτη μπορεί να γίνει με μια ποικιλία μεθόδων, αλλά οι δύο πιο δημοφιλείς είναι η Get και η Post. Το όνομα της μεθόδου προσδιορίζει το είδος της αίτησης που θα πραγματοποιηθεί και τα στοιχεία που πρέπει να συμπεριλαμβάνει. Το Servlet API (Application Programming Interface) αποτελείται από δύο σημαντικά πακέτα που περιέχουν όλες τις σημαντικές κατηγορίες διασύνδεσης του. Αυτά είναι:

- javax.servlet
- javax.servlet.http

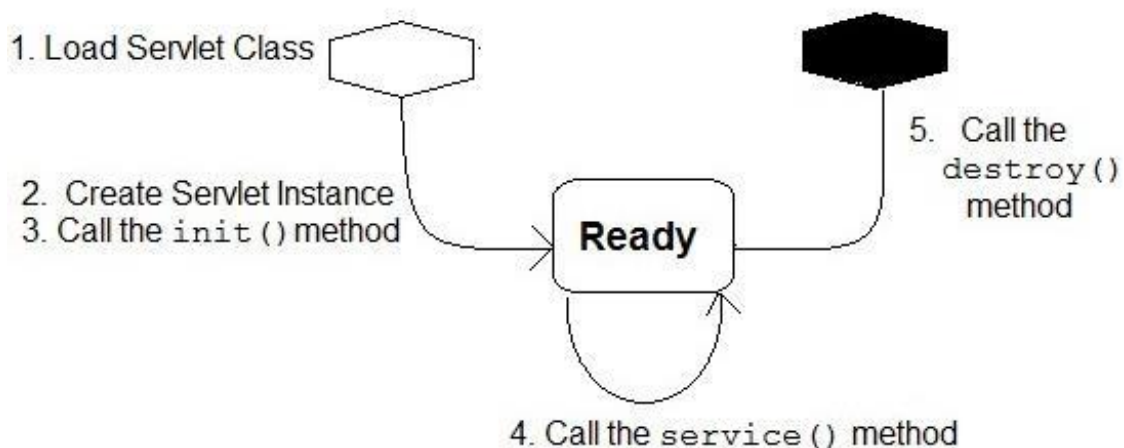
Η κατηγορία HttpServlet είναι μια κλάση η οποία ερμηνεύει την υλοποίηση των διαφόρων υπηρεσιών/μεθόδων του Servlet. Για τη δημιουργία ενός Servlet, πρέπει να δημιουργήσουμε μια κλάση που υπάγεται στη κατηγορία HttpServlet. Η κλάση αυτή, δεν υπερσχύει της μεθόδου υπηρεσίας, service() method, η οποία διαχειρίζεται το είδος του αιτήματος αλλά αναλαμβάνει τις ευθύνες της μεθόδου doGet() και/ή doPost(). Για να μπορέσει να χειριστεί το αίτημα ενός πελάτη παρέχει αρχικά δύο σημαντικές διασυνδέσεις. Αυτές είναι η javax.servlet.ServletRequest και η javax.servlet.http.HttpServletRequest οι οποίες ερμηνεύουν το αίτημα του πελάτη και παρέχουν σημαντική πληροφορία σχετικά με αυτό. Στη συνέχεια αφού εκτελεστεί το αίτημα του πελάτη ακολουθούν άλλες δύο σημαντικές διασυνδέσεις που παρέχει το Servlet API. Αυτές είναι η

ServletResponse και η HttpServletResponse και έχουν ως στόχο την αποστολή απάντησης στον πελάτη.

Οι πιο πάνω διεργασίες πραγματοποιούνται με τη βοήθεια των Διαδικτυακών δοχείων (Web containers) τα οποία είναι υπεύθυνα για τη διαχείριση και την εκτέλεση των εφαρμογών Java EE που χρησιμοποιούν Servlets ή JSP. Όταν πραγματοποιείται ένα αίτημα στο Servlet, ο διακομιστής παραπέμπει το αίτημα αυτό σε ένα Web container. Το Web container αναλαμβάνει την ευθύνη να προσφέρει τις υπηρεσίες ενός Servlet ή να δημιουργήσει ένα νέο νήμα έτσι ώστε να εξυπηρετήσει την αίτηση. Το Web container ουσιαστικά μεσολαβεί, έτσι ώστε να στείλει την αίτηση στο Servlet και να λάβει απάντηση από αυτό. Στην περίπτωση πολλών αιτήσεων, το Web container δημιουργεί πολλαπλά νήματα τα οποία αναλαμβάνει ένα μόνο Servlet και τα επεξεργάζεται.

Ο κύκλος ζωής της συνολικής διαδικασίας που αναλαμβάνει ένα Servlet δεν εκτελείται σε μια κύρια μέθοδο, main method(), αλλά ακολουθεί τα εξής στάδια. Αρχικά, ο χρήστης στέλνει ένα αίτημα για το Servlet κάνοντας κλικ σε ένα link του οποίου η διεύθυνση URL παραπέμπει σε ένα Servlet. Το Web container βρίσκει το Servlet χρησιμοποιώντας έναν περιγραφητή ανάπτυξης (deployment descriptor) και δημιουργεί ένα αντικείμενο αιτήματος (HttpServletRequest) και ένα αντικείμενο απόκρισης (HttpServletResponse). Στη συνέχεια, το Web container δημιουργεί ή διαθέτει ένα νήμα για το αίτημα αυτό και καλεί τη μέθοδο υπηρεσίας, service method(), του Servlet, δίνοντας της το περιεχόμενο της αίτησης. Η service() method, ανήκει στην κλάση (class) HttpServlet και εκτελείται στην περίπτωση που ενεργοποιηθούν οι Http μέθοδοι GET ή POST. Τότε καλεί τις μεθόδους doGet () ή doPost () οι οποίες είναι βασισμένες στον τύπο Http και θα επιτελέσουν το ρόλο των αντίστοιχων υπηρεσιών. Ακολούθως, το Servlet στέλνει την απάντηση πίσω στον πελάτη. Τέλος, αφού ολοκληρωθεί η service method(), το νήμα πεθαίνει και οι σχετικές πληροφορίες αιτήματος και απόκρισης καταστρέφονται.

Ο κύκλος ζωής παρουσιάζεται στην εικόνα 10, όπου η init method(), χρησιμοποιείται για την εκκίνηση και αρχικοποίηση του Servlet:



Εικόνα 10 Κύκλος ζωής ενός Servlet

4.5 WEB SERVICES

Μια υπηρεσία ιστού (Web service) είναι κάθε κομμάτι του λογισμικού που διατίθεται μέσω του διαδικτύου και χρησιμοποιεί ένα τυποποιημένο σύστημα XML για την ανταλλαγή μηνυμάτων. Η XML χρησιμοποιείται για την κωδικοποίηση των επικοινωνιών του Web service. Για παράδειγμα, ένας πελάτης επικαλείται ένα Web service με την αποστολή ενός μηνύματος XML, και περιμένει για μια αντίστοιχη απάντηση σε XML. Αφού όλες οι επικοινωνίες γίνονται σε XML, τα Web services δεν περιορίζονται από κανένα λειτουργικό σύστημα ή γλώσσα προγραμματισμού. Το γεγονός αυτό κάνει δυνατή την επικοινωνία μεταξύ Java και Perl ή εφαρμογές των Windows με εφαρμογές Unix.

Τα Web services μπορεί να χαρακτηριστούν ως αυτόνομα και δυναμικά applications τα οποία μπορεί να περιγραφούν, να δημοσιευθούν, να ευρεθούν, ή να κληθούν μέσω του Διαδικτύου για τη δημιουργία προϊόντων και διαδικασιών. Οι εφαρμογές αυτές μπορούν να είναι τοπικές ή και web-based και κατασκευάζονται βάση καθιερωμένων προτύπων, όπως το TCP / IP, HTTP, Java, HTML και XML. Χρησιμοποιούνται επίσης και για την ανταλλαγή μηνυμάτων μεταξύ εφαρμογών τα οποία μπορεί να περιλαμβάνουν προγράμματα, αντικείμενα, μηνύματα ή ακόμα και έγγραφα καθώς επίσης και στην ανταλλαγή δεδομένων μέσω δικτύων υπολογιστών με παρόμοιο τρόπο όπως γίνεται η επικοινωνία μεταξύ διεργασιών σε έναν υπολογιστή.

Συνοπτικά, υπηρεσίες ιστού μπορούμε να ονομάσουμε οποιαδήποτε υπηρεσία που:

- Είναι διαθέσιμη μέσω του Διαδικτύου ή κάποιου ιδιωτικού δικτύου
- Χρησιμοποιεί ένα τυποποιημένο σύστημα ανταλλαγής μηνυμάτων XML
- Δεν συνδέεται με οποιαδήποτε λειτουργικό σύστημα ή γλώσσα προγραμματισμού
- Αυτοπεριγράφεται μέσω μιας κοινής γραμματικής XML
- Εντοπίζεται μέσω ενός απλού μηχανισμού αναζήτησης

Τα Web Services βασίζονται σε XML και σε HTTP και η λειτουργικότητά τους στηρίζεται στα εξής συστατικά:

- SOAP (Simple Object Access Protocol)
- UDDI (Universal Description, Discovery and Integration)
- WSDL (Web Services Description Language)

Λειτουργία ενός Web service

Το Web service επιτρέπει την επικοινωνία μεταξύ των διαφόρων εφαρμογών με τη χρήση ανοικτών προτύπων όπως των HTML, XML, WSDL και SOAP. Με τη βοήθεια της XML επισημαίνει τα δεδομένα, με τη βοήθεια της SOAP μπορεί να μεταφέρει ένα μήνυμα, και με τη χρήση WSDL μπορεί να καθορίσει τη διαθεσιμότητα της υπηρεσίας.

Τα Web services που παρέχει το NetBeans IDE αναλαμβάνουν την εκτέλεση ενός κομματιού της συνολικής επεξεργασίας ενός Web application και μπορούν να χρησιμοποιηθούν για την ενσωμάτωση εφαρμογών οι οποίες είναι γραμμένες σε διαφορετικές γλώσσες προγραμματισμού και οι οποίες τρέχουν σε διαφορετικές πλατφόρμες. Το γεγονός ότι μπορούν να αναπτυχθούν σε διακομιστή εφαρμογών Java EE, φάνηκε χρήσιμο αφού τις κάνει κατάλληλες για τον GlassFish Server που επιλέξαμε.

Υπάρχουν δύο κατηγορίες Web services:

- Τα REST-based. Τα REpresentational State Transfer (Παραστατική Κατάσταση Μεταφοράς) βασίζονται σε μια νέα μέθοδο δημιουργίας Web services, όπου χρησιμοποιούνται Uniform Resource Identifiers (URIs) τα οποία επεξεργάζονται μέσω HTTP λειτουργιών και
- τα SOAP/WSDL-based. Αποτελούν τα παραδοσιακά Web services, όπου οι διεπαφές αποτελούν URL διευθύνσεις και παρουσιάζονται με τη βοήθεια WSDL εγγράφων (όπως XML). Η ανταλλαγή μηνυμάτων στη κατηγορία αυτή γίνεται με τη βοήθεια της SOAP. Οι δύο κατηγορίες προσφέρουν παρόμοιες δυνατότητες και χρησιμοποιούνται συχνά σε υλοποιήσεις Web application.

4.6 JavaScript και jQuery

Η κύρια χρήση της JavaScript βασίζεται σε client-side web scripting, όπου ασχολείται με την αλληλεπίδραση στο πεδίο της εφαρμογής με τον πελάτη. Τα περισσότερα από τα δυναμικά χαρακτηριστικά μιας ιστοσελίδας οφείλονται στον ενσωματωμένο κώδικα JavaScript στην HTML. Είναι υπεύθυνη για οποιαδήποτε αλλαγή και προσθήκη δυναμικών εφέ εφόσον φορτωθεί η ιστοσελίδα και έχει την δυνατότητα, με τη χρήση διαδικασιών (functions), να προκαλέσει μεταβολές στη διεπαφή.

Ακόμα ένα χρήσιμο εργαλείο που εκμεταλλευτήκαμε είναι η βιβλιοθήκη jQuery. Η jQuery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη έτσι ώστε να απλοποιήσει την υλοποίηση σεναρίων (scripting) στην πλευρά του πελάτη (client-side) της HTML και υποστηρίζεται από πολλούς Web browsers. Κυκλοφόρησε τον Ιανουάριο του 2006 από τον John Resig και η χρήση της ξεπερνά το 65% των πρώτων 10.000 ιστοσελίδων με τη μεγαλύτερη επισκεψιμότητα.

4.7 Android Studio

Το λειτουργικό Android τρέχει σε εκατοντάδες εκατομμύρια συσκευές σε περισσότερες από 190 χώρες σε όλο τον κόσμο. Είναι η μεγαλύτερη εγκατεστημένη βάση από οποιαδήποτε πλατφόρμα κινητών συσκευών και αυξάνεται ταχέως. Το λειτουργικό σύστημα Android της Google προσφέρει μια πλατφόρμα υψηλής ποιότητας για τη δημιουργία εφαρμογών και παιχνιδιών για τους χρήστες του, καθώς και μια ανοιχτή αγορά για την άμεση διανομή σε αυτούς. Κάθε μέρα ενεργοποιούνται σε όλο τον κόσμο πάνω από ένα εκατομμύριο νέες συσκευές Android. Το λειτουργικό Android είναι open-source, κάτι το οποίο προτιμάται από πολλούς καταναλωτές αλλά κυρίως από προγραμματιστές και το οποίο συνεισφέρει στη συνεχή προσφορά και ζήτηση εφαρμογών οι οποίες διατίθενται στο Google Play.

Συνεχώς προσπαθεί να ξεπεράσει τα όρια της συνεργασίας υλικού και λογισμικού και να φέρει νέες δυνατότητες για τους χρήστες και τους προγραμματιστές. Η πλατφόρμα Android Studio, σε συνδυασμό με τα εργαλεία SDK (Software Development Kit) που παρέχει, επιτρέπουν στους προγραμματιστές να υλοποιήσουν ισχυρές καινούριες εφαρμογές που εκμεταλλεύονται τις τελευταίες τεχνολογίες κινητών επικοινωνιών. Δίνει πρόσβαση στους developers στις πιο πρόσφατες τεχνολογίες και καινοτομίες όπως επεξεργασία πολλαπλών πυρήνων, γραφικά υψηλής απόδοσης, state-of-the-art αισθητήρες, οθόνες αφής και πολλά άλλα. Προσφέρει επίσης τη δυνατότητα μοντελοποίησης ενιαίων εφαρμογών οι οποίες μπορεί να τρέχουν αποτελεσματικά σε ένα ευρύ φάσμα συσκευών όπως τηλέφωνα και tablets, ανεξαρτήτως του μεγέθους της οθόνης τους. Παρέχει εργαλεία που επωφελούνται από τις δυνατότητες του υλικού που διαθέτει κάθε συσκευή και συμπεριλαμβάνει αυτόματη προσαρμογή της διεπαφής του χρήστη για τη βέλτιστη εμφάνιση σε κάθε συσκευή. Ταυτόχρονα προσφέρει τεράστιο ποσοστό

ελέγχου στο χρήστη, για να προσαρμόζει το UI ακριβώς όπως επιθυμεί. Κατά τη διάρκεια της εκτέλεσης εφαρμόζει αυτόματα τα σωστά σύνολα πόρων με βάση το μέγεθος της οθόνης, την πυκνότητα, της ρυθμίσεις γλώσσας, την τοποθεσία, κ.ο.κ.

Για την αποτελεσματική ανάπτυξη εφαρμογών, τα Android Developer Tools προσφέρουν ένα πλήρες Java IDE που περιλαμβάνει πακέτα, βιβλιοθήκες και λειτουργία αποσφαλμάτωσης (debugging). Προσφέρεται επίσης πρόγραμμα το οποίο δημιουργεί εικονικές συσκευές έτσι ώστε ο κατασκευαστής να μπορεί να δει πώς θα φαίνεται η εφαρμογή σε οποιαδήποτε συσκευή. Ως μια ανοιχτή αγορά, το Google Play δίνει τη ευχέρεια στους κατασκευαστές να δημοσιεύουν όποτε θέλουν, όσο συχνά θέλουν, και για τους πελάτες που θέλουν. Επομένως, ο κατασκευαστής μπορεί να επιλέξει να διανέμει ευρέως σε όλες τις αγορές και συσκευές την εφαρμογή του ή να επικεντρωθεί σε συγκεκριμένες κατηγορίες, συσκευές ή περιοχές. Η επέκταση της δυνατότητας του Android Studio στην υλοποίηση μιας εφαρμογής Android θα συμβάλει άμεσα στην προώθηση της διαδικασίας ευθυγράμμισης που κατασκευάστηκε.

ΚΕΦΑΛΑΙΟ 5: ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ANDROID

5.1 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΦΙΛΟΞΕΝΟΥΜΕΝΗ ΣΤΟ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ANDROID.

Η ιδέα ανάπτυξης μίας εφαρμογής που θα μπορεί να κάνει πιο εύκολη την ζωή εκείνων που έχουν στην κατοχή τους συσκευές Android και δεν είναι αρκετά εξοικειωμένοι με αυτήν την πλατφόρμα, ήρθε ως αποτέλεσμα τριβής με άτομα που ανήκουν στην παραπάνω κατηγορία και που βρίσκονται στον στενό οικογενειακό αλλά και φιλικό κύκλο.

Οι δυσκολίες που έπρεπε να προσπεραστούν ανήκουν σε δύο κατηγορίες:

1. Το γεγονός ότι η συσκευή έχει πληθώρα προ-εγκατεστημένων εφαρμογών, και επομένως αρκετά εικονίδια που κάνουν πιο δύσκολο τον εντοπισμό των βασικών λειτουργιών.
2. Το μέγεθος των κουμπιών αλλά και της γραμματοσειράς της εκάστοτε εφαρμογής δεν είναι ικανό να δείξει στο χρήστη τι ακριβώς πρέπει να κάνει και σε ποιο ακριβώς σημείο.

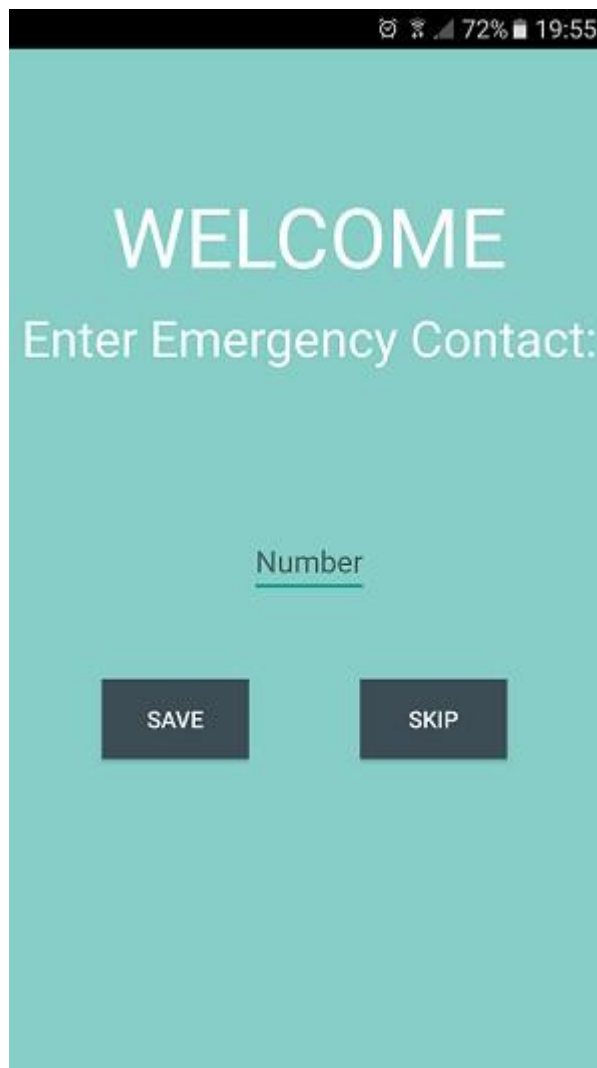
Ύστερα από μία μικρή έρευνα που πραγματοποιήσαμε καταλήξαμε στις 6 πιο σημαντικές εφαρμογές που έπρεπε να περιλαμβάνει η εφαρμογή μας.

- Τηλέφωνο
- Επαφές
- Μηνύματα
- E-Mail
- Κάμερα
- Έκθεση Φωτογραφιών

Για κάθε μία από αυτές χρησιμοποιήσαμε ειδικό εικονίδιο που δηλώνει την κάθε λειτουργία, πριν ο χρήστης επιλέξει αυτήν που επιθυμεί. Ταυτόχρονα κάναμε χρήση του ηχείου της συσκευής προκειμένου το ερέθισμα προς το χρήστη να είναι οπτικοακουστικό κάνοντας ακόμη πιο εύκολη την κατανόηση του τι ακριβώς αντιπροσωπεύει το κουμπί-λειτουργία που επέλεξε. Ένα παράδειγμα για το παραπάνω είναι: ο χρήστης πατά το εικονίδιο της κάμερας, και αμέσως θα ακουστεί ηχητικό μήνυμα(π.χ. Camera).

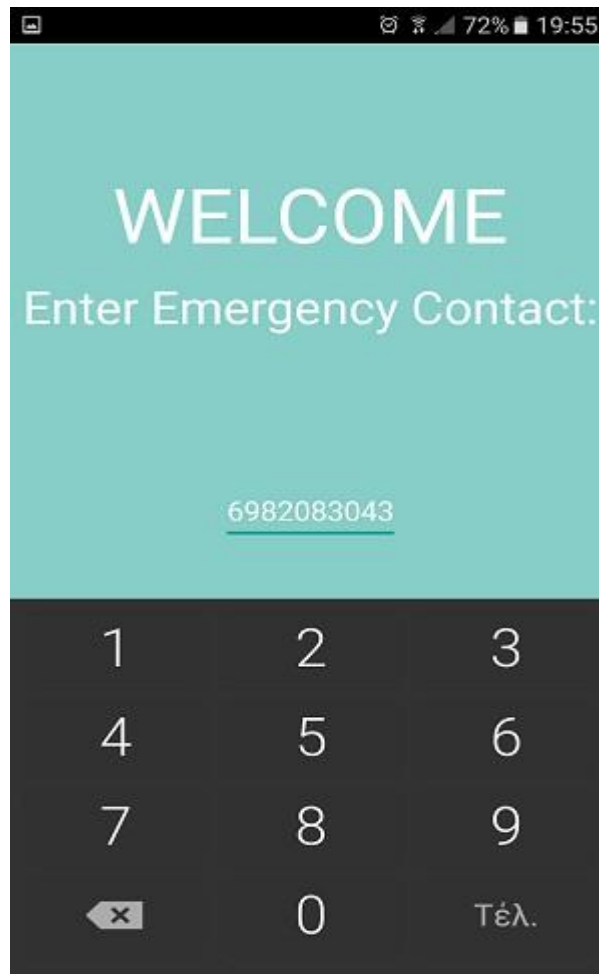
5.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ANDROID ΣΕ ΕΙΚΟΝΕΣ

Ξεκινώντας από την αρχική οθόνη, ο χρήστης καλείται να εισάγει τον αριθμό τηλεφώνου έκτακτης ανάγκης. Ο συγκεκριμένος αριθμός, όπως θα δούμε σε επόμενη ενότητα, θα χρησιμεύσει προκειμένου να ειδοποιηθεί εάν οι πληροφορίες που παίρνουμε από τους αισθητήρες υποδηλώνουν πτώση ή άγνοια θέσης.



Εικόνα 11 Αρχική Οθόνη "Help The Old Ones"

Αφού πληκτρολογήσει τον αριθμό επιλέγει SAVE για την αποθήκευση του, ή SKIP σε περίπτωση που δεν θέλει να δηλώσει την επαφή έκτακτης ανάγκης τη δεδομένη χρονική στιγμή.



Εικόνα 12 Εισαγωγή Επαφής Έκτακτης Ανάγκης

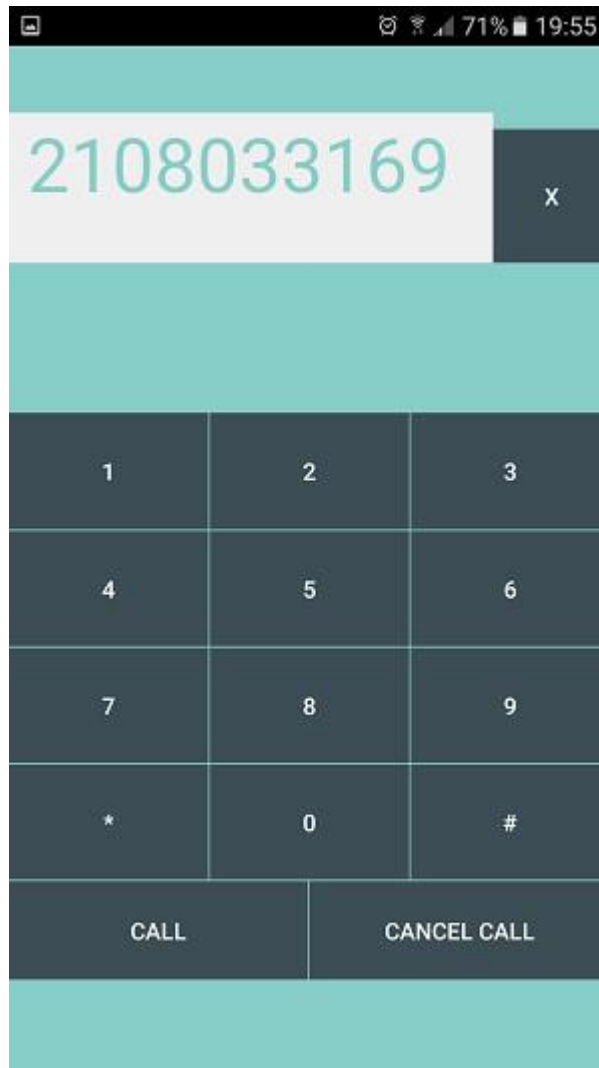
Αμέσως μετά εμφανίζεται εμφανίζεται η δεύτερη οθόνη Εικόνα 13, που αποτελεί και το κεντρικό μενού και που δείχνει όλες τις δυνατές βασικές λειτουργίες που περιλαμβάνει η εφαρμογή Android.



Εικόνα 13 Βασικές Λειτουργίες της Εφαρμογής (Κεντρικό Μενού)

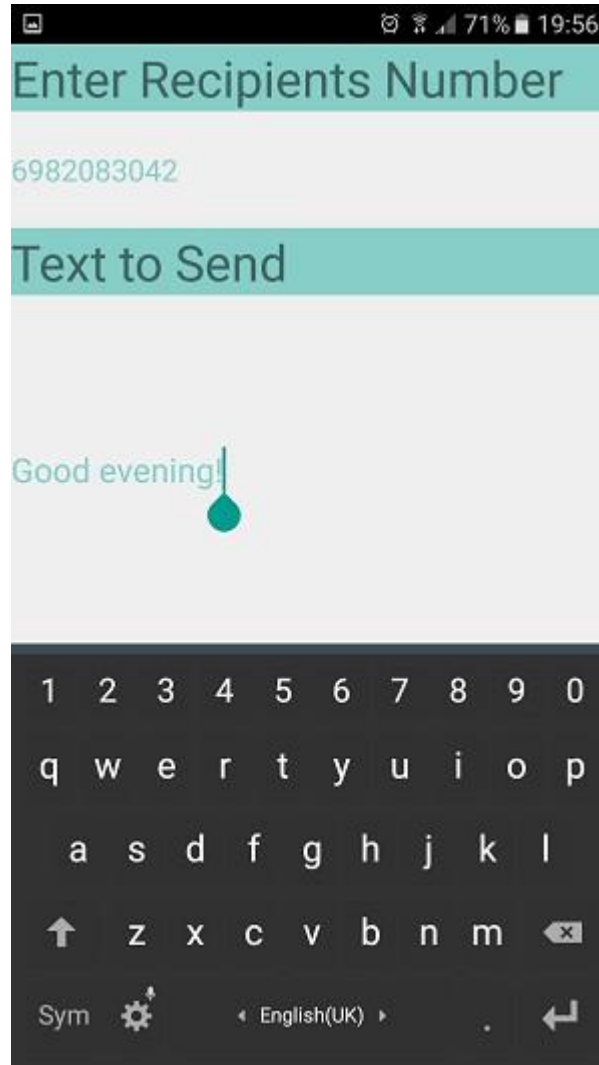
Όπως έχουμε ήδη αναφέρει, έχουμε χρησιμοποιήσει εικονίδια που υποδεικνύουν άμεσα τι ακριβώς αντιπροσωπεύει κάθε κουμπί της συγκεκριμένης οθόνης (Εικόνα 13). Ταυτόχρονα πάνω από κάθε κουμπί έχουμε τοποθετήσει και κείμενο που περιγράφει την εκάστοτε λειτουργία. Γεγονός που μειώνει στο ελάχιστο την πιθανότητα ο χρήστης να επιλέξει κάποια λειτουργία και να μην κατανοεί τι ακριβώς κάνει.

Πιο συγκεκριμένα, ο χρήστης μπορεί να πραγματοποιήσει μια κλήση, έχοντας στην οθόνη του μόνο τα βασικά στοιχεία ώστε να κάνει χρήση της λειτουργία του τηλεφώνου. Ουσιαστικά αποκρύφτηκαν όλα τα περιττά κουμπιά που μπορεί οδηγήσουν το χρήστη σε σύγχυση.



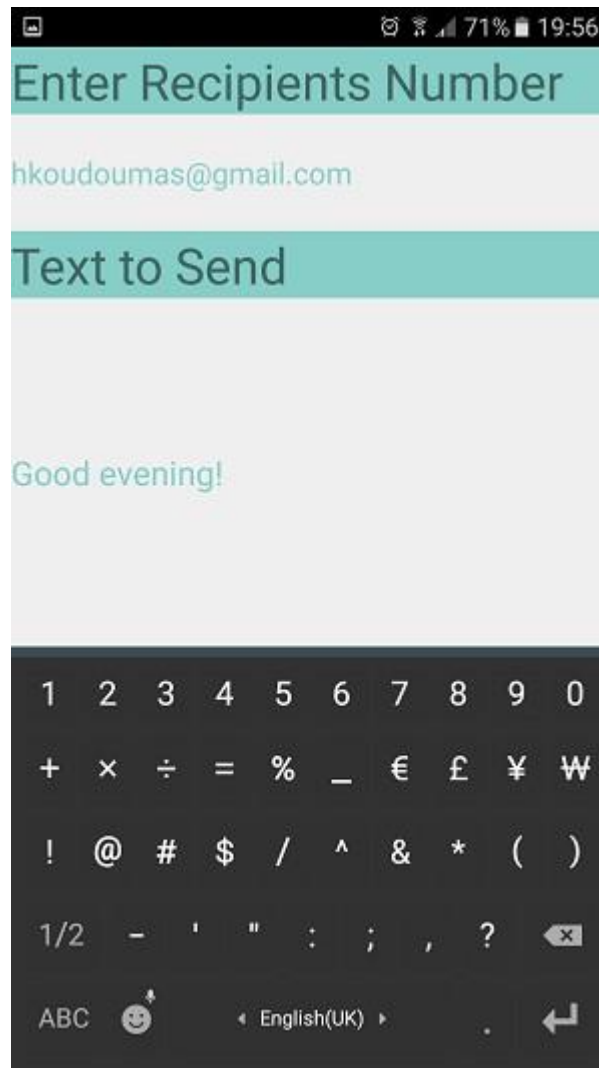
Εικόνα 14 Λειτουργία «Τηλέφωνο» της Εφαρμογής

Δίνεται η δυνατότητα να στείλει ένα μήνυμα (SMS) (Εικόνα15). Και σε αυτή την οθόνη αφαιρέσαμε κάθε περιττό στοιχείο και μεγαλώσαμε τις λέξεις κλειδιά που δηλώνουν με τι θα πρέπει να συμπληρωθεί το αντίστοιχο κείμενο.



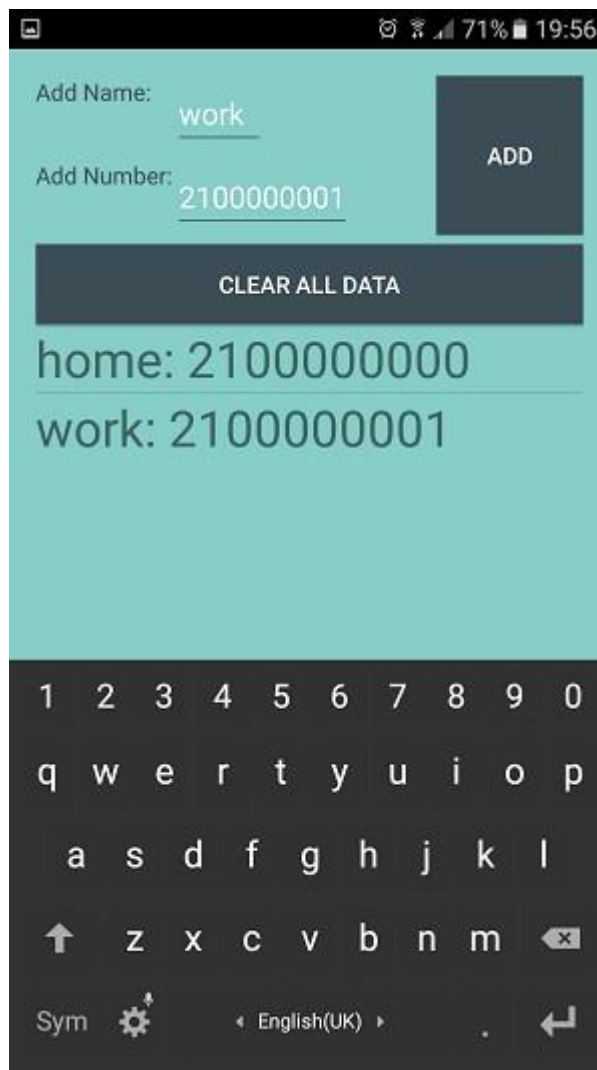
Εικόνα 15 Λειτουργία «Μήνυμα» της Εφαρμογής

Κινούμενοι στο ίδιο μοτίβο με το εκείνο της αποστολής μηνύματος κάναμε τις ίδιες αλλαγές και στη λειτουργία του e-mail (Εικόνα 14), ώστε να καθιστά την αποστολή ενός e-mail εύκολη ακόμη και για τον πιο μη εξοικειωμένο χρήστη.



Εικόνα 16 Λειτουργία «e-Mail» της εφαρμογής

Για την εφαρμογή των επαφών, δημιουργήσαμε ένα activity, το οποίο θα δείχνει με σαφήνεια τι ακριβώς πρέπει να πατήσει για να κάνει μια εισαγωγή. Προσθέσαμε ένα κουμπί διαγραφής, και τη δυνατότητα να κάνει κλήση μόλις πατηθεί κάποια επαφή από την εμφανιζόμενη λίστα. Το λειτουργικό Android δυσκολεύει αρκετά την εισαγωγή μίας επαφής, καθώς στο βωμό της αποθήκευσης στο λογαριασμό Google, του συγχρονισμού των επαφών και της κοινής χρήσης αυτών, θυσιάστηκε η απλότητα της εισαγωγής μιας επαφής.



Εικόνα 17 Λειτουργία «Επαφές» της Εφαρμογής

Πηγαίνοντας ένα μικρό βήμα παρακάτω με την συγκεκριμένη εφαρμογή, θελήσαμε να βάλουμε και κάποιες λειτουργίες που θα γίνονται διαθέσιμες με την φωνή του χρήστη. Έτσι λοιπόν μπορεί να κάνει και χρήση του φλας της συσκευής σαν φακό, εφόσον διαθέτει η συσκευή με τη φωνή. Σε αυτή τη λειτουργία προσθέσαμε το άνοιγμα του φακού μέσω φωνητικής εντολής χρησιμοποιώντας API της GOOGLE. Μέσα σε αυτό το activity υπάρχει και η επιλογή αλλαγής της επαφής έκτακτης ανάγκης, η προσθήκη αυτής αν ο χρήστης είχε επιλέξει το κουμπί SKIP, κατά την πρώτη χρήση της εφαρμογής στην πρώτη οθόνη που εμφανίζεται (Εικόνα 11).



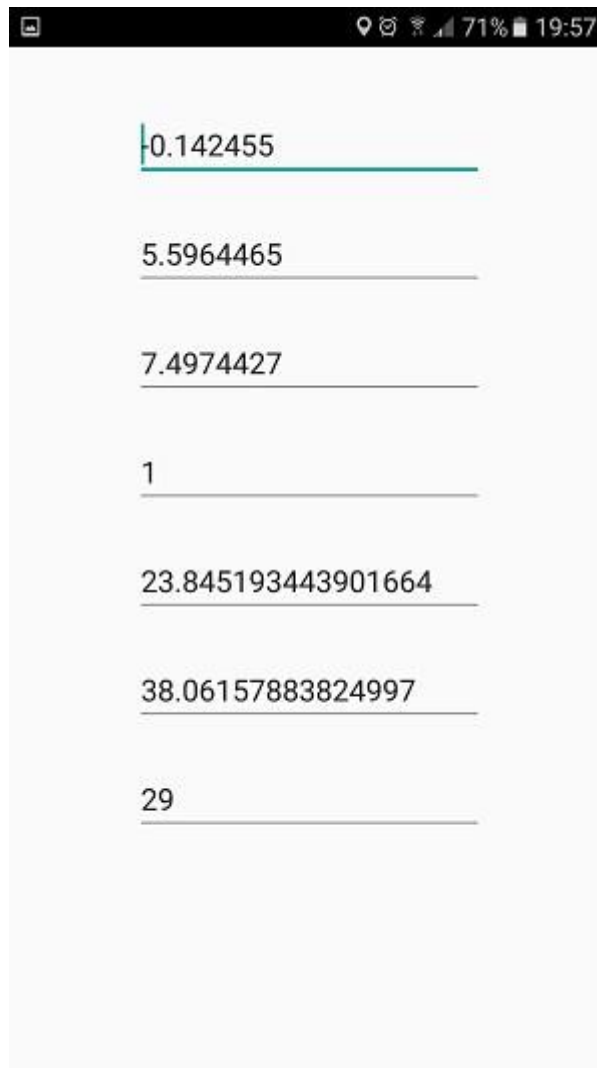
Εικόνα 18 Λειτουργία «Φωνητικές Εντολές-Αλλαγή αρ. Έκτακτης Ανάγκης» της Εφαρμογής

Εκτός όμως από το να κάνουμε πιο εύχρηστη τη συσκευή Android, θέσαμε και έναν ακόμη στόχο: να αξιοποιήσουμε αισθητήρες του κινητού τηλεφώνου που δεν είναι σε πολλούς γνωστοί, με σκοπό να αποτρέψουμε δύσκολες για τον χρήστη καταστάσεις (άγνοια θέσης, πτώση).

5.3 ΠΑΡΟΥΣΙΑΣΗ ΤΩΝ ΕΙΔΙΚΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

5.3.1 ΛΕΙΤΟΥΡΓΙΑ EMERGENCY

Όπως δείξαμε σε προηγούμενη ενότητα, ο χρήστης όταν ανοίξει για πρώτη φορά την εφαρμογή, καλείται να εισάγει έναν αριθμό έκτακτης ανάγκης (ο αριθμός αυτός μπορεί να αλλάξει οποιαδήποτε στιγμή). Ο σκοπός της ύπαρξης μιας τέτοιας επαφής έχει να κάνει με την επιπρόσθετη λειτουργία EMERGENCY. Η συγκεκριμένη λειτουργία δίνει τη δυνατότητα στην εφαρμογή να μπορεί να αναγνωρίσει πτώση ή απότομη κίνηση της συσκευής.



Εικόνα 19 Τιμές Αισθητήρων

Οι τιμές που εμφανίζονται στην Εικόνα 19 δείχνουν τις μετρήσεις του επιταχυνσιόμετρο στον άξονα X,Y και Z. Η επόμενη τιμή αναφέρεται στο πόσες φορές έχει ξεπεραστεί η τιμή του κατωφλίου για την λειτουργία Άγνοια-Θέσης. Στη συνέχεια, οι επόμενες δύο τιμές δείχνουν το γεωγραφικό μήκος και γεωγραφικό πλάτος σύμφωνα με τον αισθητήρα GPS. Η τελευταία τιμή μας δείχνει το μετρητή του κατωφλίου για την αναγνώριση πτώσης.

Πιο συγκεκριμένα με τα δεδομένα που παίρνει από τους αισθητήρες είναι σε θέση να καταλάβει αν γίνεται

1. Περιστροφή της συσκευής (Λειτουργία- Άγνοια θέσης)
2. Απότομη αλλαγή ύψους (Λειτουργία-Ανίχνευση πτώσης)

και εφόσον δεν γίνει καμία άλλη ενέργεια από το χρήστη, ενημερώνεται κατάλληλα η επαφή έκτακτης ανάγκης.

5.3.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Η λειτουργία EMERGENCY, όπως φαίνεται και στην εικόνα 20, λαμβάνει τιμές από τους αισθητήρες της εφαρμογής (έχουν μπει καθαρά για λόγους ελέγχου στη μεταπτυχιακή διατριβή).



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.emergency_layout);

    sensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    accelerometer = sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    sensorManager.registerListener((SensorEventListener) this, accelerometer, SensorManager.SENSOR_DELAY_NORMAL);

    grav = sensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    sensorManager.registerListener((SensorEventListener) this, grav, SensorManager.SENSOR_DELAY_NORMAL);

    xAxisLabel = (TextView) findViewById(R.id.editText1);
    yAxisLabel = (TextView) findViewById(R.id.editText2);
    zAxisLabel = (TextView) findViewById(R.id.editText3);
    textTimer = (TextView) findViewById(R.id.editText4);
    lat = (TextView) findViewById(R.id.editText5);
    longt = (TextView) findViewById(R.id.editText6);
    timer = (TextView) findViewById(R.id.editText7);
    SharedPreferences loadContact = getSharedPreferences("myData", MODE_PRIVATE);
    final String emerContact = loadContact.getString("Number", "199");
}

```

Εικόνα 20 onCreate της λειτουργίας Emergency

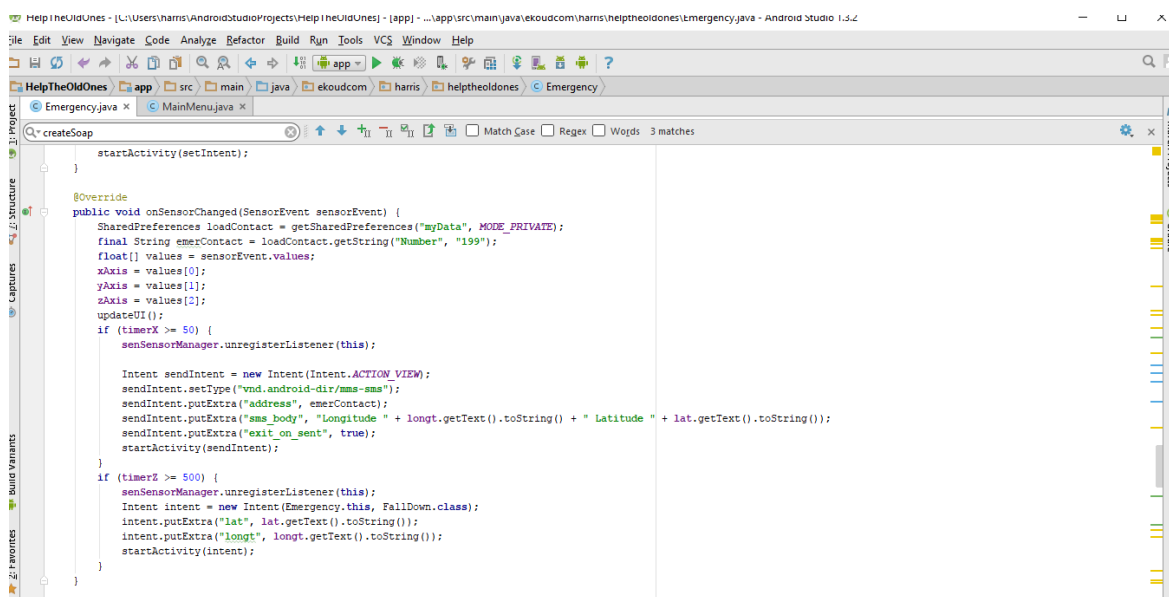
Όταν ξεκινήσει η συγκεκριμένη λειτουργία, όπως φαίνεται στην εικόνα 21, αρχικοποιούνται οι διαχειριστές των εμπλεκόμενων αισθητήρων. Είναι πολύ σημαντικό για την καλή λειτουργία της συσκευής τόσο σε απόδοση CPU όσο και σε κατανάλωση μπαταρίας, οι μετρήσεις που λαμβάνονται να μην είναι πάρα πολύ κοντινές χρονικά η μία στην άλλη. Γι αυτό το λόγο, όπως φαίνεται και στο κώδικα στη μέθοδο registerListener, έχουμε ορίσει το δεύτερο όρισμα να έχει την τιμή SensorManager.SENSOR_DELAY_NORMAL. Πρακτικά σημαίνει ότι θα «ακούμε» τον αισθητήρα κάθε 200ms.

Αφού δηλώσουμε τους διαχειριστές των αισθητήρων, θα πρέπει να ορίσουμε και τα κομμάτια του UI που θα εμφανίζουν τι τιμές αυτών. Για τις τιμές του επιταχυνσιόμετρο (άξονες x,y,z) χρησιμοποιούμε τα labels xAxisLabel, yAxisLabel και zAxisLabel. Αποτελούν μέρος της οθόνης που έχουμε ορίσει για τη συγκεκριμένη λειτουργία και είναι τα TextViews που μας παρέχει η πλατφόρμα Android. Χρησιμοποιούμε άλλα δύο TextViews για να εμφανίσουμε το γεωγραφικό μήκος και πλάτος που παίρνουμε από το GPS της συσκευής. Τέλος οι τιμές του 4^{ου} και 7^{ου} πεδίου αποτυπώνουν τους μετρητές που έχουμε θέσει ώστε όταν περάσουν μια ορισθείσα τιμή

να ενεργοποιήσουν με τη σειρά τους άλλες λειτουργικότητες που θα περιγράψουμε σε επόμενες ενότητες.

5.3.1.1 ΛΕΙΤΟΥΡΓΙΑ-ΑΓΝΟΙΑ ΘΕΣΗΣ

Στις περιπτώσεις που χρήστης δεν γνωρίζει τη ακριβή του θέση (έχει χαθεί), κουνώντας την συσκευή για περισσότερα από 5 δευτερόλεπτα μπορεί να ειδοποιηθεί την επαφή έκτακτης ανάγκης που έχει οριστεί στην πρώτη οθόνη της εφαρμογής.



```

startActivity(setIntent);
}

@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    SharedPreferences loadContact = getSharedPreferences("myData", MODE_PRIVATE);
    final String emerContact = loadContact.getString("Number", "199");
    float[] values = sensorEvent.values;
    xAxis = values[0];
    yAxis = values[1];
    zAxis = values[2];
    updateUI();
    if (timerX >= 50) {
        senSensorManager.unregisterListener(this);

        Intent sendIntent = new Intent(Intent.ACTION_VIEW);
        sendIntent.setType("vnd.android-dir/sms-sms");
        sendIntent.putExtra("address", emerContact);
        sendIntent.putExtra("sms_body", "Longitude " + longt.getText().toString() + " Latitude " + lat.getText().toString());
        sendIntent.putExtra("exit_on_sent", true);
        startActivity(sendIntent);
    }
    if (timerZ >= 500) {
        senSensorManager.unregisterListener(this);
        Intent intent = new Intent(Emergency.this, FallDown.class);
        intent.putExtra("lat", lat.getText().toString());
        intent.putExtra("longt", longt.getText().toString());
        startActivity(intent);
    }
}

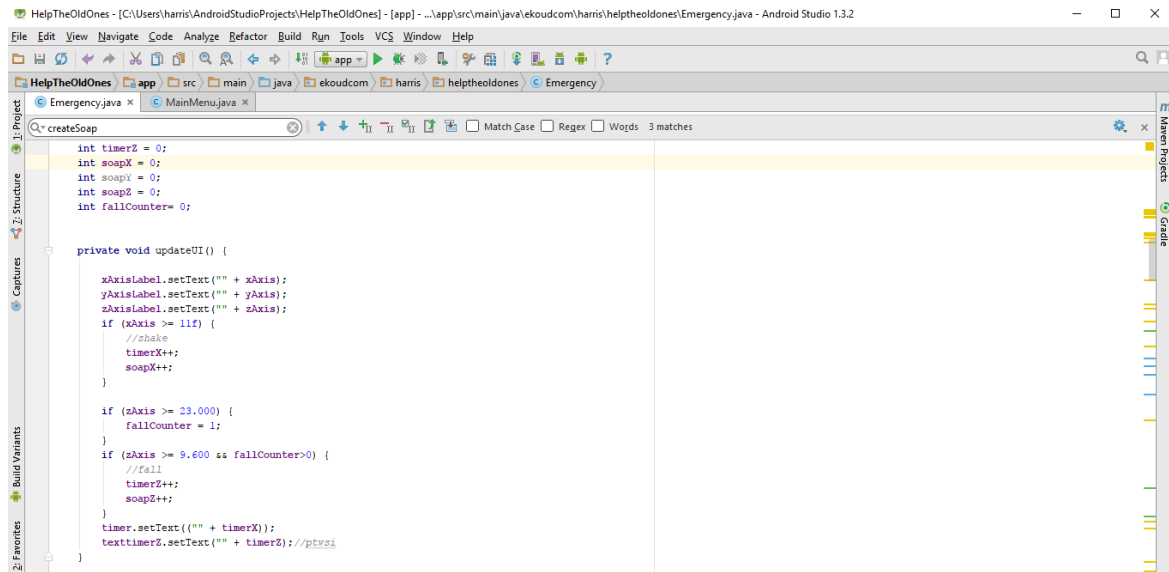
```

Εικόνα 21 OnSensorChanged

Έχουμε κάνει override μια μέθοδο τη onSensorChanged, που αφορά γενικά όλους τους αισθητήρες. Αυτό γιατί έπρεπε μέσα σε αυτή τη μέθοδο να προσθέσουμε κώδικα που θα κάνει δύο πολύ σημαντικές δουλειές:

1. Να καλεί τη μέθοδο updateUi
2. Να ελέγχει τις τιμές των μετρητών.

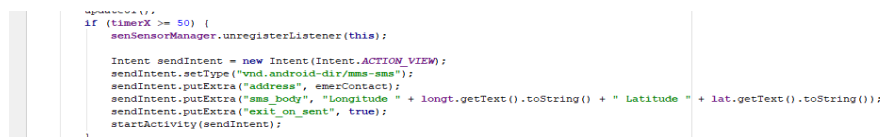
Η μέθοδος updateUi αποτελεί τη «ραχοκοκαλιά της εφαρμογής». Είναι εκείνη που θα εμφανίσει στο UI τις τιμές των μετρητών, αλλά και εκείνη που θα αυξήσει την τιμή του μετρητή όταν ξεπεραστεί το κατώφλι που έχει οριστεί.



Εικόνα 22 Ενημέρωση των τιμών στο GUI μέσω της UpdateUi

Όπως δείχνει η Εικόνα 22 αρχικά ορίζει τις τιμές των labels που αναφέραμε προηγουμένως, ώστε να είναι διακριτές στο «μπροστινό» μέρος της εφαρμογής. Στη συνέχεια όμως υπάρχουν κάποιοι έλεγχοι που καθορίζουν τις τιμές των μετρητών. Ο πρώτος έλεγχος έχει να κάνει με τη λειτουργία της Άγνοιας-Θέσης. Οι απότομες κινήσεις της συσκευής από τα αριστερά προς τα δεξιά και το αντίστροφο, οδηγούν σε μεταβολή του xAxis. Εμείς καταγράφουμε αυτές τις μεταβολές αυξάνοντας το μετρητή κάθε φορά που θα ξεπεράσουν την τιμή 11f.

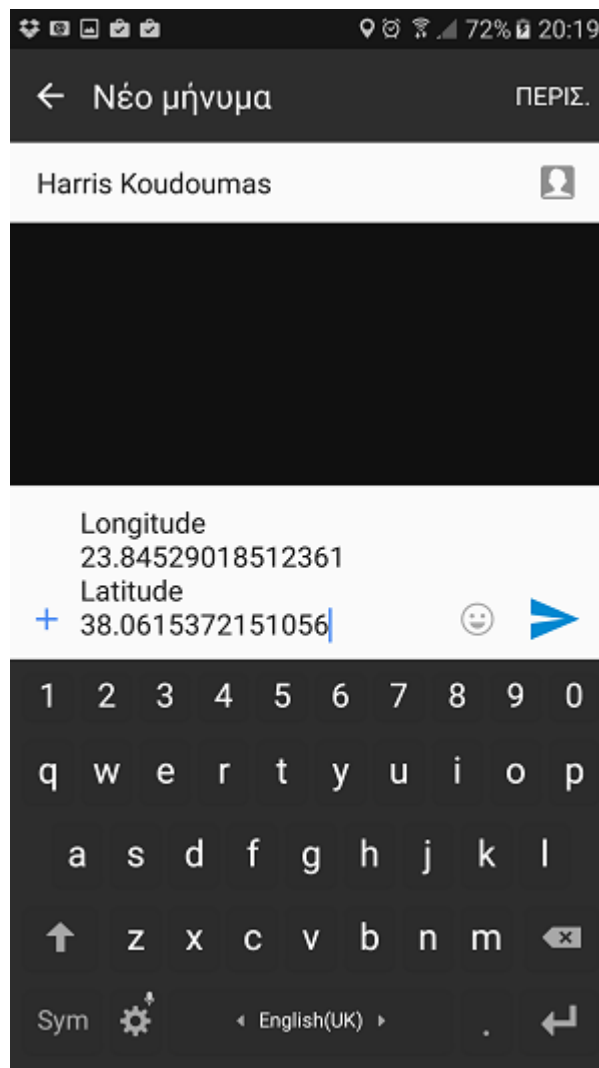
Οι δοκιμές που κάναμε προκειμένου να κατανοήσουμε τη λειτουργία του συγκεκριμένου αισθητήρα μας έδειξαν πως η τιμή (11f), θα αποτελέσει το κατώφλι που πρέπει να περάσει το xAxis από το επιταχυνσιόμετρο προκειμένου να αποφύγουμε άσκοπες ειδοποιήσεις, σε περιπτώσεις που δεν αποτελούν έκτακτη ανάγκη. Έχοντας λοιπόν έναν μετρητή, καταγράφουμε και αξιολογούμε την κίνηση που κάνει η κινητή συσκευή. Εφόσον ο μετρητής περάσει τον αριθμό των επαναλήψεων που αντιστοιχούν σε πέντε δευτερόλεπτα (50 φορές), γίνεται κλήση του Activity για την αποστολή μηνύματος. Σε αντίθετη περίπτωση θα μηδενίζει όταν οι διακυμάνσεις των δεδομένων του αισθητήρα ανήκουν στο σύνολο (-4f,4f).



Εικόνα 23 Κλήση του Activity για την Αποστολή Μηνύματος

Έτσι λοιπόν, όταν η τιμή στον timerX γίνει μεγαλύτερη από την τιμή 50, έχουμε ορίσει να καλείται το Activity αποστολής μηνύματος. Θα πρέπει σε αυτό το σημείο να πούμε ότι τόσο το τηλέφωνο του παραλήπτη όσο και το κείμενο του μηνύματος θα είναι προκαθορισμένα από την εφαρμογή, με την εξής διαφορά. Το τηλέφωνο θα έχει την τιμή που έχει θέσει ο χρήστης, και το κείμενο θα περιέχει τις τιμές του γεωγραφικού μήκους και πλάτους που είχε τη δεδομένη χρονική στιγμή η συσκευή.

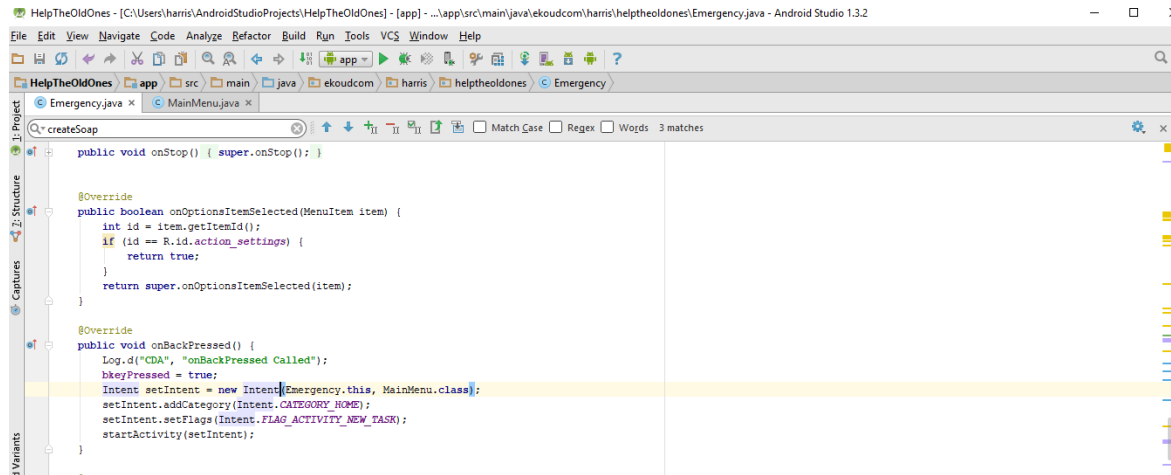
Η εφαρμογή θα έχει αναγνωρίσει αυτή την παρατεταμένη κίνηση και θα έχει συντάξει ένα μήνυμα με το γεωγραφικό μήκος και πλάτος "longitude:Value, latitude: Value" που έχει πάρει από το GPS αισθητήρα, όπως φαίνεται στην Εικόνα 24.



Εικόνα 24 Μήνυμα Έκτακτης Ανάγκης

Αξίζει να αναφερθεί ότι η αποστολή μπορεί να αποτραπεί από το χρήστη στην περίπτωση που δεν έχει γίνει ηθελημένα, πατώντας το κουμπί της επιστροφής της συσκευής. Αυτό θα έχει σαν συνέπεια να οδηγηθεί ο χρήστης στο αρχικό μενού της εφαρμογής.

Και σε αυτήν την περίπτωση έχουμε κάνει override τη μέθοδο `onBackPressed`. Έχουμε θέσει σαν ενέργεια το άνοιγμα του αρχικού μενού. Το λειτουργικό μας παρέχει αυτή τη δυνατότητα χρησιμοποιώντας τις εντολές που εμφανίζονται στην Εικόνα 25.



```
public void onStop() { super.onStop(); }

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

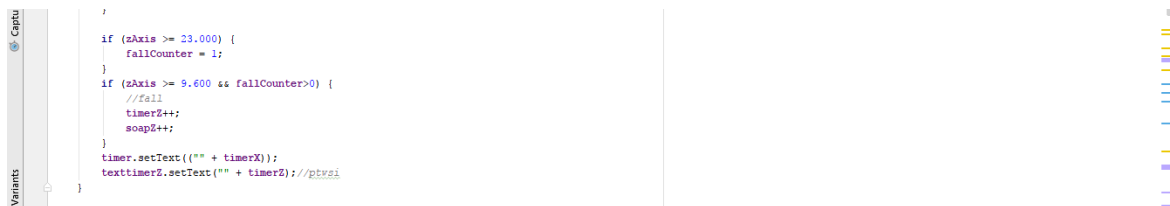
@Override
public void onBackPressed() {
    Log.d("CDM", "onBackPressed Called");
    KeyEvent keyEvent = new KeyEvent(KeyEvent.ACTION_UP, null);
    Intent setIntent = new Intent(Emergency.this, MainMenu.class);
    setIntent.addCategory(Intent.CATEGORY_HOME);
    setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(setIntent);
}
```

Εικόνα 25 `onBackPressed` του Activity της λειτουργία *Emergency*

5.3.1.2 ΛΕΙΤΟΥΡΓΙΑ ΑΝΙΧΝΕΥΣΗΣ ΠΤΩΣΗΣ

Πτώση σημαίνει σύμφωνα με το λεξικό την προς τα κάτω κίνηση λόγω της βαρύτητας. Πάνω σε αυτόν τον ορισμό πατήσαμε προκειμένου να υλοποιήσουμε τη δεύτερη αναγνώριση κίνησης της λειτουργίας Emergency.

Όπως αναφέραμε και στην προηγούμενη ενότητα, η μέθοδος `onSensorChanged` έχει δύο σημαντικές εργασίες να κάνει, να εμφανίσει τις τιμές των αισθητήρων στο UI και να ελέγχει την τιμή των μετρητών. Για τη συγκεκριμένη λειτουργία έχουμε ορίσει έναν νέο μετρητή, τον `timerZ`.



```
},
    if (zAxis >= 23.000) {
        fallCounter = 1;
    }
    if (zAxis >= 9.600 && fallCounter>0) {
        //fall
        timerZ++;
        soapZ++;
    }
    timer.setText(("" + timerX));
    texttimerZ.setText(("" + timerZ); //πίτσι
}
```

Εικόνα 26 Μετρητές για την Αναγνώριση Πτώσης

Θα πρέπει σε αυτό το σημείο να κάνουμε μια αναφορά και σε μία ακόμη μεταβλητή που ο ρόλος της είναι πολύ σημαντικός. Στην Εικόνα 26 υπάρχει ένας νέος έλεγχος ο οποίος θέτει την τιμή της μεταβλητής `fallCounter` ίση με τη μονάδα. Πριν ξεκινήσει η υλοποίηση της συγκεκριμένης λειτουργίας, έπρεπε να σκεφτούμε και να αποκλείσουμε τις περιπτώσεις όπου η κατακόρυφη κίνηση της συσκευής να μην συνδέεται με πτώση από ύψος. Παραδείγματα τέτοιων περιπτώσεων είναι

- Να αφήσουμε από ύψος τη συσκευή πάνω στο γραφείο ή στο κρεβάτι.
- Να κατεβούμε από το λεωφορείο.
- Να κατεβούμε με ταχύτητα απότομη σκάλα.

Πραγματοποιήσαμε αρκετούς ελέγχους και πειραματικές διαδικασίες, και καταλήξαμε ότι στις παραπάνω περιπτώσεις η τιμή που παίρνει η μεταβλητή `zAxis`, είναι μικρότερη της τιμής 23f. Φτάσαμε λοιπόν στο συμπέρασμα ότι έναυσμα για να ενεργοποιηθεί η παρακολούθηση του αισθητήρα βαρύτητας και του επιταχυνσιόμετρου είναι να ξεπεραστεί αυτή η τιμή. Όπως φαίνεται και στο κώδικα, υπάρχει ο έλεγχος **if (zAxis >= 23.000)**, όπου αν ισχύει θα αλλάξει τη τιμή της παραπάνω μεταβλητής.

Η απότομη ή μη αλλαγή ύψους, έχει ως αποτέλεσμα την αλλαγή τιμής στη μεταβλητή `Z axis` του επιταχυνσιόμετρου. Έχουμε θέσει ένα κατώφλι ύστερα από αρκετές δοκιμές, όπου όταν η τιμή της **zAxis** ξεπεράσει το συγκεκριμένο αυτο κατώφλι (9,6f) τότε να αυξάνεται ο μετρητής `counterZ`. Εφόσον αμέσως μετά σταθεροποιηθεί η τιμή `Zaxis`, σημαίνει ότι η κινητή συσκευή βρίσκεται ακινητοποιημένη.

Επομένως, ο συνδυασμός, της τιμής zAxis να είναι μεγαλύτερη του 9.6f και της fallcounter να είναι ίση με τη μονάδα, είναι ικανή και αναγκαία συνθήκη ενεργοποίησης της λειτουργίας Emergency.

Αναλύοντας περαιτέρω την μέθοδο onSensorChanged, θα πρέπει να αναφερθούμε και στο 2^ο έλεγχο που πραγματοποιεί.

```
if (timerZ >= 500) {
    senSensorManager.unregisterListener(this);
    Intent intent = new Intent(Emergency.this, FallDown.class);
    intent.putExtra("lat", lat.getText().toString());
    intent.putExtra("longt", longt.getText().toString());
    startActivity(intent);
}
```

Εικόνα 27 Έλεγχος των Μετρητών Πτώσης

Τα 30 δευτερόλεπτα ύστερα από ελέγχους που κάναμε αντιστοιχούν η τιμή του zAxis να είναι μεγαλύτερη από το 9.6f 500 φορές. Θέσαμε λοιπόν το 2^ο έλεγχο να κοιτάει την τιμή της timerZ και εφόσον αυτή είναι μεγαλύτερη είτε ίση με 500 να κάνει χρήση των εντολών του λειτουργικού Android προκειμένου να ανοίξει μια επιπλέον οθόνη.

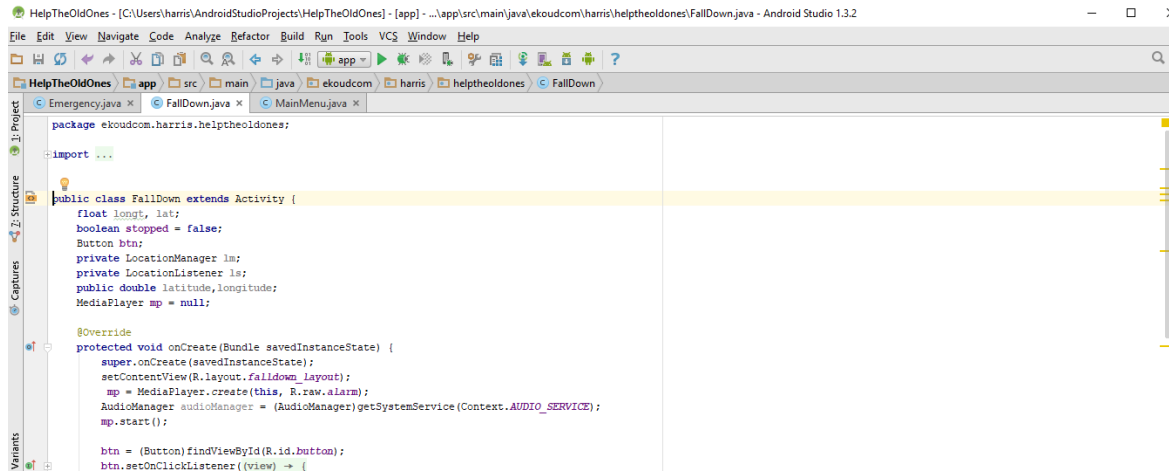
Στην Εικόνα 27 εκτός από τις εντολές για να ανοίξει το επόμενο activity έχουμε προσθέσει και κάποιες ακόμη που είναι πολύ σημαντικές όπως **senSensorManager.unregisterListener(this);** που κλείνει ουσιαστικά το κανάλι επικοινωνίας με τους αισθητήρες ώστε να μην υπάρχει κατανάλωση πόρων τους συστήματος και ενέργειας χωρίς λόγο, καθώς ο έλεγχος για τις τιμές που παρατηρούμε έχει ολοκληρωθεί. Τέλος με τις εντολές

- **intent.putExtra("lat", lat.getText().toString());** και
- **intent.putExtra("longt", longt.getText().toString());** περνάμε στο επόμενο Activity τις τιμές που έχουμε από το GPS.



Εικόνα 28 Activity-Button/Οθόνη που εμφανίζεται μετά από την παρέλευση τριάντα δευτερολέπτων, ύστερα από πτώση

Όταν ανοίξει η παραπάνω εικόνα από την ενεργοποίηση της Emergency για την πτώση, έχουμε θέσει ένα ηχητικό μήνυμα να ξεκινήσει να αναπαράγεται αμέσως. Το ηχητικό αυτό μήνυμα είναι διάρκειας 30 δευτερολέπτων και έχει ως στόχο να προκαλέσει το ενδιαφέρον των ατόμων που βρίσκονται κοντά στο χρήστη ώστε αν θεωρηθεί απαραίτητο να του παρασχεθεί βοήθεια.



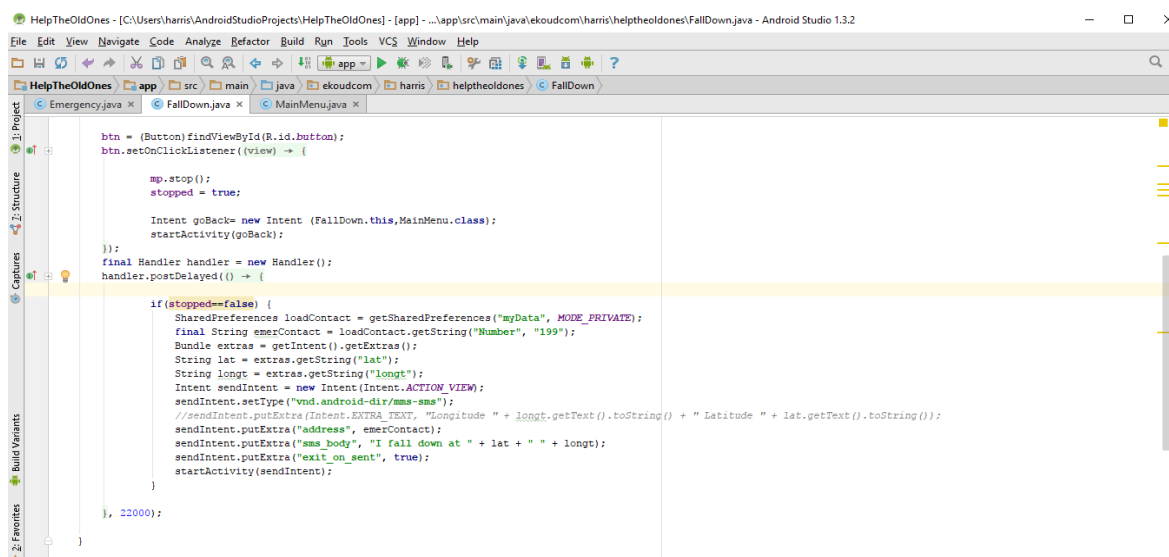
Εικόνα 29 Αναπαραγωγή Ηχητικού Μηνύματος μετά από Πτώση

Και σε αυτήν την περίπτωση όπως δείχνει η Εικόνα 29, κάναμε override την μέθοδο onCreate. Η πλατφόρμα Android μας παρέχει τη δυνατότητα να αποθηκεύσουμε στο project ηχητικά μηνύματα και να τα χρησιμοποιούμε όπου κρίνουμε απαραίτητο. Με τις εντολές:

1. **mp = MediaPlayer.create(this, R.raw.alarm);**
2. **AudioManager audioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);**
3. **mp.start();**

ορίζουμε ποιος από τους αποθηκευμένους ήχους θα αναπαραχθεί. Αρχικοποιούμε έναν διαχειριστή AudioManager ώστε να είναι προσπελάσιμο από το κώδικα το ηχείο της συσκευής και στη συνέχεια ξεκινά η αναπαραγωγή του ήχου.

Σε αυτό το Activity έχουμε προσθέσει κώδικα ώστε όταν ο χρήστης έχει πέσει αλλά δεν διατρέχει κάποιο κίνδυνο ή είναι καλά στη υγεία του, να σταματήσει την αναπαραγωγή του ήχου μέσα στο διάστημα των 30 δευτερολέπτων.



Εικόνα 30 Παύση Ηχητικού Μηνύματος

Η εικόνα που εμφανίζεται στην οθόνη της συσκευής έχει οριστεί να είναι ένα μεγάλο κουμπί που καλύπτει όλη την επιφάνεια της συσκευής. Πατώντας πάνω στην οθόνη ο χρήστης θα σταματήσει το ηχητικό μήνυμα αλλά ταυτόχρονα θα αποτραπεί και η αποστολή μηνύματος στην επαφή έκτακτης ανάγκης με αντίστοιχο μήνυμα.

Οι εντολές:

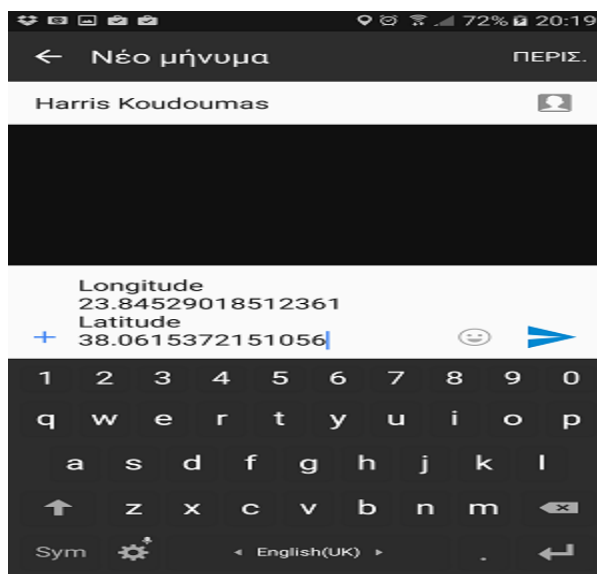
```
btn = (Button)findViewById(R.id.button);
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        mp.stop();
        stopped = true;

        Intent goBack= new Intent (FallDown.this,MainMenu.class);
        startActivity(goBack);
    }
});
```

ορίζουν τι θα συμβεί όταν ο χρήστης πατήσει πάνω στην οθόνη. Θα γίνει παύση του ήχου και θα ορίσει τη τιμή της μεταβλητής stopped να είναι true. Η μεταβλητή αυτή όπως βλέπουμε και στην εικόνα, είναι το κριτήριο με βάση το οποίο θα γίνει ή όχι η αποστολή του μηνύματος στην επαφή που έχει ορίσει ο χρήστης.

Στην περίπτωση που έχουν περάσει τα 30 δευτερόλεπτα και η τιμή της stopped παραμένει false, όπως και στη λειτουργία της Άγνοιας-Θέσης η εφαρμογή έχει συντάξει ένα μήνυμα με το γεωγραφικό μήκος και γεωγραφικό πλάτος "I fell down at longitude:Value, lattitude: Value" που έχει πάρει από το GPS αισθητήρα, όπως φαίνεται στην Εικόνα 31.



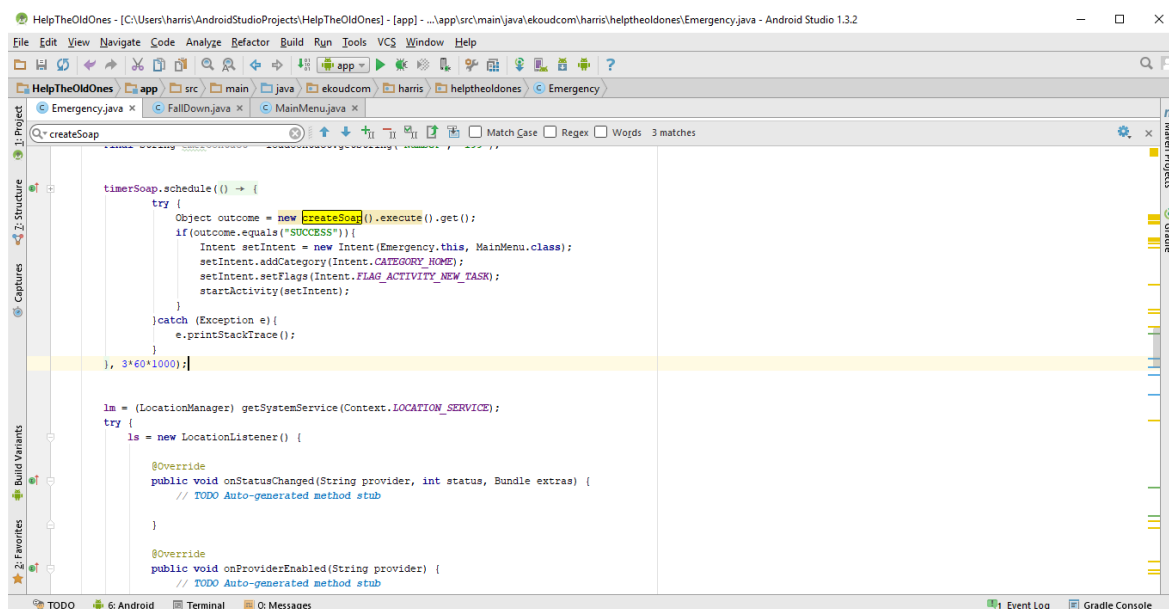
Εικόνα 31 Αποστολή Μηνύματος μετά από Πτώση

5.3.1.3 ΕΝΗΜΕΡΩΣΗ ΕΦΑΡΜΟΓΗΣ ΦΙΛΟΞΕΝΟΥΜΕΝΗΣ ΣΤΟ ΝΕΦΟΣ

Μία επιπλέον λειτουργία που προσθέσαμε στην εφαρμογή αυτή, έχει να κάνει με την σύγχρονη ενημέρωση μια απομακρυσμένης web εφαρμογής

Για την εφαρμογή αυτή θα πούμε περισσότερα σε επόμενο κεφάλαιο. Είναι αρκετό να επισημάνουμε ότι θα φιλοξενηθεί σε αφιερωμένο διακομιστή, ώστε να είναι σε θέση να λαμβάνει τις τιμές των μετρήσεων από τους αισθητήρες της συσκευής κάθε στιγμή.

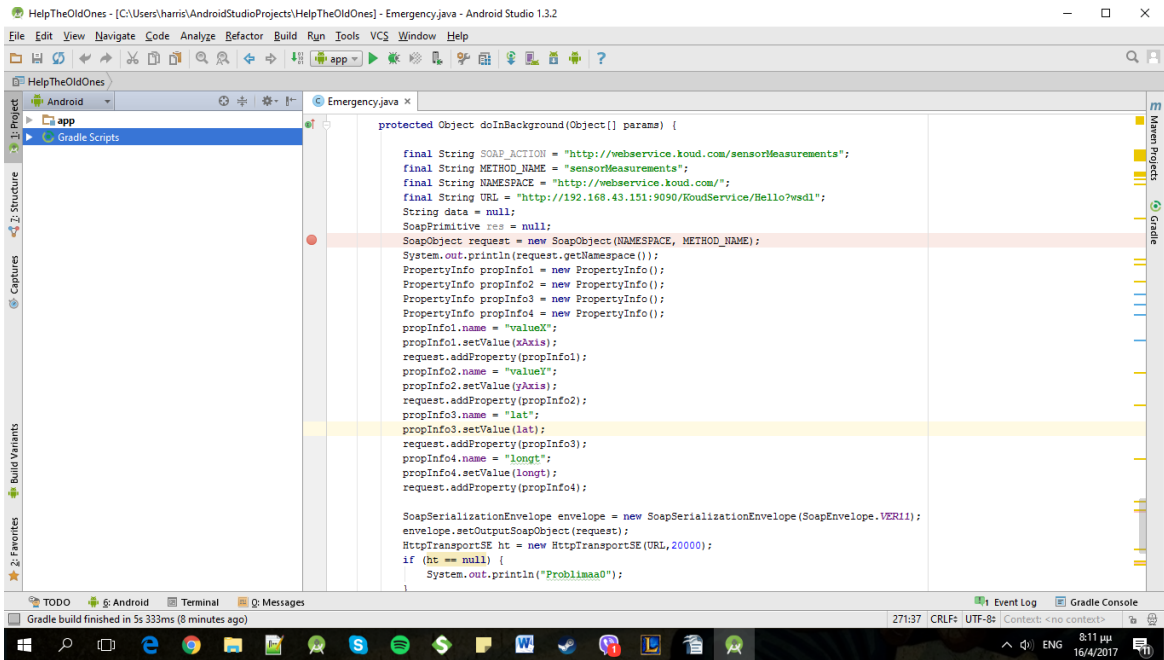
Κινούμενοι πάντα στο ίδιο μοτίβο με τις προηγούμενες λειτουργίες, χρησιμοποιήσαμε έναν νέο χρονοδιακόπτη 3 λεπτών. Κάθε διάστημα τριών λεπτών πραγματοποιείται μια κλήση του web-service της παραπάνω εφαρμογής και ενημερώνει έναν πίνακα για τις μετρήσεις που έχει πάρει από τους αισθητήρες.



Εικόνα 32 Timer τριών λεπτών για κλήση προς το Web Service

Όπως φαίνεται στην Εικόνα 32 χρησιμοποιήθηκε το πακέτο Timer της Java, ώστε να επιτύχουμε την κλήση αυτή όποτε επιθυμούμε. Αφού περάσουν τα τρία λεπτά γίνεται η κλήση μιας εσωτερικής κλάσης μέσα στο Activity Emergency, η οποία είναι επιφορτισμένη με το να πάρει τα δεδομένα και να δημιουργήσει ένα μήνυμα τύπου SOAP ώστε να το στείλει στην εφαρμογή του νέφους.

Κάνοντας χρήση της κλάσης AsyncTask της πλατφόρμας Android, δημιουργήσαμε μια κλάση που φτιάχνει ένα SOAPMessage με όλες τις τιμές που έχουν οι αισθητήρες εκείνη τη χρονική στιγμή. Στη συνέχεια, χρησιμοποιώντας το Endpoint (Static IP) της web εφαρμογής, αποστέλλονται οι μετρήσεις με σκοπό να αποθηκευτούν και να γίνουν προσπελάσιμες από άμεσα ενδιαφερόμενα με το χρήστη άτομα.



Εικόνα 33 Δημιουργία XML για αποστολή προς Webservice

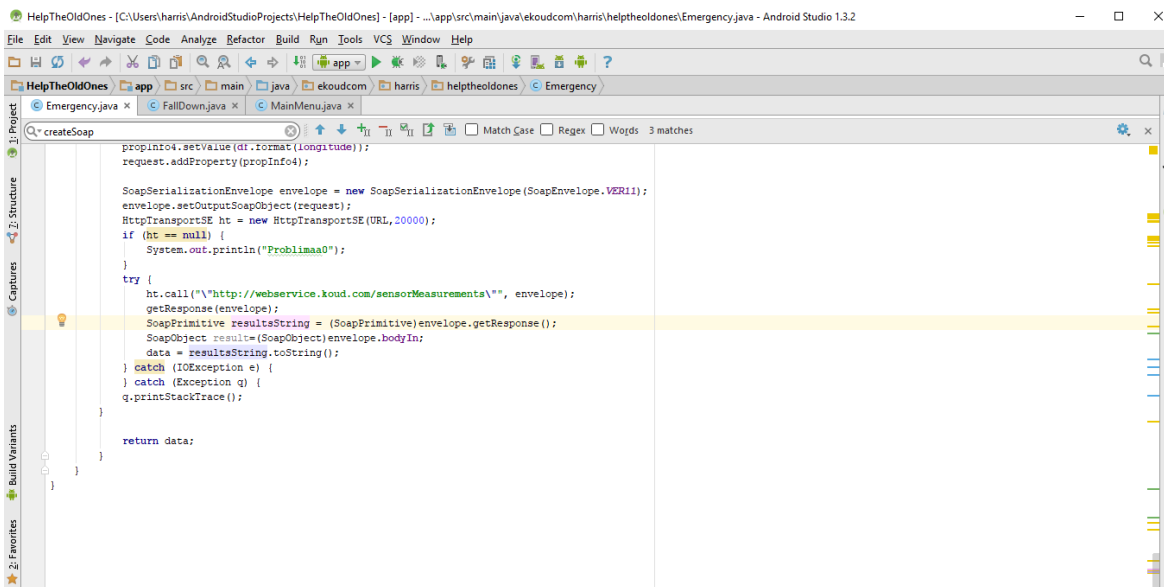
Πριν όμως δημιουργήσουμε το μήνυμα πρέπει να αρχικοποιήσουμε κάποιες μεταβλητές του που είναι απαραίτητες ώστε να φτάσει τελικά το μήνυμα στο Web Service. Παρακάτω αναφέρουμε τις εντολές με τις οποίες γίνεται η αρχικοποίηση.

```
final String SOAP_ACTION = "http://webservice.koud.com/sensorMeasurements";
final String METHOD_NAME = "sensorMeasurements";
final String NAMESPACE = "http://webservice.koud.com/";
final String URL = "http://192.168.43.151:9090/KoudService/Hello?wsdl";
```

Η πρώτη μαζί με τη δεύτερη δηλώνει ποια είναι η ενέργεια/μέθοδος που θα καλέσουμε. Όπως είναι γνωστό ένα Web Service μπορεί να περιέχει περισσότερα του ενός operations. Είναι λοιπόν σημαντικό να δηλώσουμε ποιο θέλουμε, στην προκειμένη περίπτωση είναι το sensorMeasurements. Η τρίτη εντολή δηλώνει το namespace του Web Service θα χρησιμοποιηθεί στη συνέχεια για το κτίσιμο του xml. Τέλος η τέταρτη εντολή αποτελεί το σύνδεσμο όπου βρίσκεται το WSDL.

Στη συνέχεια δημιουργούμε ένα αντικείμενο της βιβλιοθήκης SoapObject(request) στο οποίο βάζουμε ως ορίσματα τις παραπάνω μεταβλητές που ορίσαμε. Το αντικείμενο αυτό θα αποτελέσει τελικά και το μήνυμα που θα σταλθεί, επομένως πρέπει να έχει και τις μεταβλητές των αισθητήρων. Για αυτό το λόγο κάναμε χρήση της κλάσης PropertyInfo τέσσερις φορές όσες και οι τιμές που περνάμε στο μήνυμα(xAxis, yAxis, Latitude, Longitude). Τα instances αυτής της κλάσης μας δίνουν τη δυνατότητα να ορίσουμε ζευγάρια μεταβλητή/τιμή, τα οποία θα χρησιμοποιηθούν για να διαμορφωθεί το τελικό XML αρχείο.

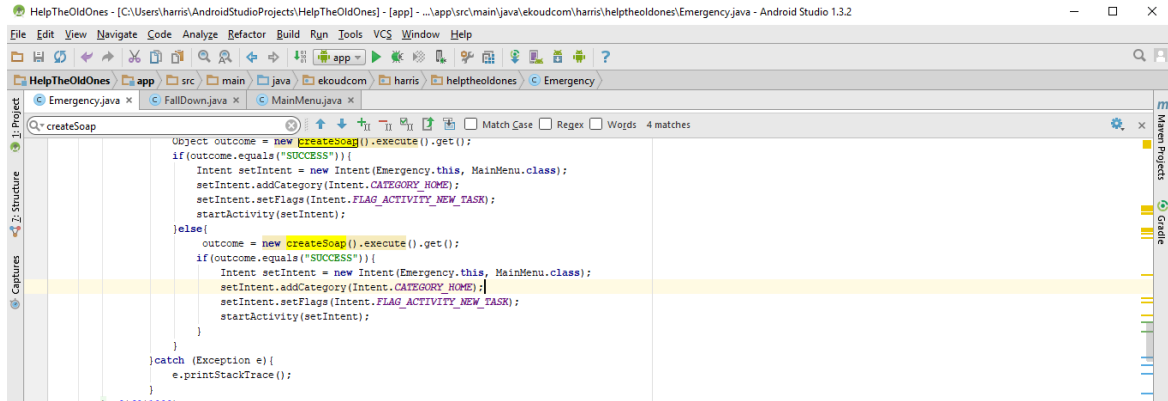
```
PropertyInfo propInfo1 = new PropertyInfo();
propInfo1.name = "valueX";
propInfo1.setValue(1);
request.addProperty(propInfo1);
```



Εικόνα 34 Δημιουργία Αντικείμενου SoapSerializationEnvelope

Ως επόμενο βήμα πρέπει να δημιουργηθεί ένα αντικείμενο της κλάσης SoapSerializationEnvelope, στο οποίο θα εισάγουμε το request μαζί με όλες τις υπόλοιπες πληροφορίες. Αφού λοιπόν γίνει η κλήση του Web Service λαμβάνουμε την απάντηση την οποία αποθηκεύουμε στη μεταβλητή data. Η απάντηση μπορεί να πάρει τιμές **SUCCESS**, εάν όλα πήγαν καλά ή **FAIL**, αν εμφανίστηκε κάποιο σφάλμα τόσο στη κλήση όσο και στην αποθήκευση των τιμών στη βάση.

Είναι πολύ σημαντικό να ξέρουμε αν όντως έγινε με επιτυχία η κλήση και η αποστολή μηνύματος καθώς σε αντίθετη περίπτωση θα πρέπει να κάνουμε εκ νέου κλήση. Όπως φαίνεται στην Εικόνα 35 έχουμε προσθέσει επιπλέον κώδικα, όπου γίνεται έλεγχος της απάντησης που έχουμε πάρει από την εφαρμογή ιστού. Εάν η διαδικασία έγινε με επιτυχία οδηγούμαστε στο αρχικό μενού της εφαρμογής Android, σε διαφορετική περίπτωση γίνεται επανάληψη της κλήσης. Θα πρέπει να αναφέρουμε ότι θα πραγματοποιηθεί μια μόνο επανάληψη προκειμένου να μην γίνεται άσκοπη κατανάλωση πόρων του συστήματος ή δημιουργία «κίνησης» στην εφαρμογή που φιλοξενείται στο νέφος.



Εικόνα 35 Έλεγχος του Αποτελέσματος της 1^{ης} κλήσης προς το Web Service

Οι μετρήσεις αυτές, όπως έχουμε αναφέρει αποτελούν τιμές του γυροσκοπίου, του επιταχυνσιόμετρου, του μετρητή της βαρύτητας και του GPS. Ο συνδυασμός τους είναι ικανός να μας δείξει τις κινήσεις που έχει κάνει το άτομο που χρησιμοποιεί την κινητή συσκευή, αλλά και την ακριβή θέση του (Για την παραβίαση ή όχι της προσωπικής ελευθερίας θα κάνουμε αναφορά σε επόμενη ενότητα).

ΚΕΦΑΛΑΙΟ 6: ΠΑΡΟΥΣΙΑΣΗ ΤΗΝ ΕΦΑΡΜΟΓΗΣ ΦΙΛΟΞΕΝΟΥΜΕΝΗΣ ΣΤΟ ΝΕΦΟΣ

Όπως έχουμε περιγράψει σε προηγούμενα κεφάλαια, κύριος στόχος της εφαρμογής Android δεν είναι μόνο να διευκολύνει εκείνους που χρησιμοποιούν την κινητή συσκευή, αλλά να αποτελέσει ένα ισχυρό και χρήσιμο εργαλείο για άτομα που ενδιαφέρονται για την κατάσταση των χρηστών της εφαρμογής.

Για το λόγο αυτό δημιουργήσαμε μία εφαρμογή βασισμένη στον ιστό ώστε να επικοινωνεί απευθείας με την κινητή συσκευή, η οποία δέχεται κλήσεις από την εκάστοτε συσκευή και στη συνέχεια θα αποτυπώνει την τελευταία θέση που είχε με βάση πάντα τις τιμές από τον GPS αισθητήρα. Στόχος της εφαρμογής είναι αποθηκεύει σε πραγματικό χρόνο τις τιμές που παίρνει και στη συνέχεια να τις αποτυπώνει πάνω στο χάρτη προσδιορίζοντας την ακριβή τοποθεσία του χρήστη.

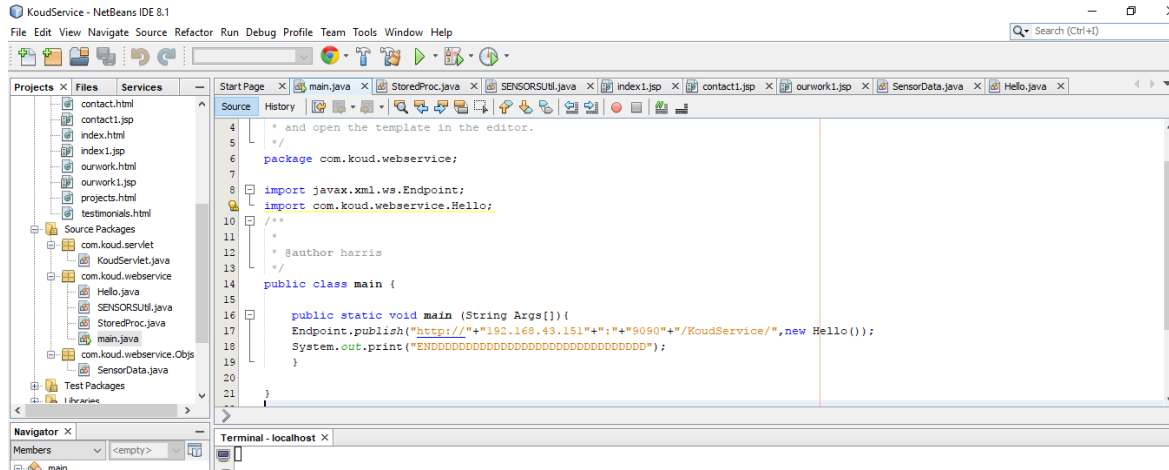
Στο πλαίσιο της συγκεκριμένης μεταπτυχιακής διατριβής περιορίσαμε την λειτουργικότητα μόνο στην εμφάνιση της θέσης της συσκευής και όχι των κινήσεων αυτής. Υπήρχαν πολλοί σημαντικοί λόγοι, ένα εξ αυτών έχει να κάνει με την παραβίαση της ιδιωτικότητας του χρήστη, γεγονός που έκανε δύσκολη την εύρεση ατόμων να συναινέσουν στη λήψη δεδομένων από τις κινητές συσκευές τους.

6.1 ΣΧΕΔΙΑΣΜΟΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εν λόγω εφαρμογή, έχει υλοποιηθεί σε κώδικα Java, HTML5, ORACLE PL/SQL, Javascript(JQUERY) και περιλαμβάνει τα εξής δομικά στοιχεία:

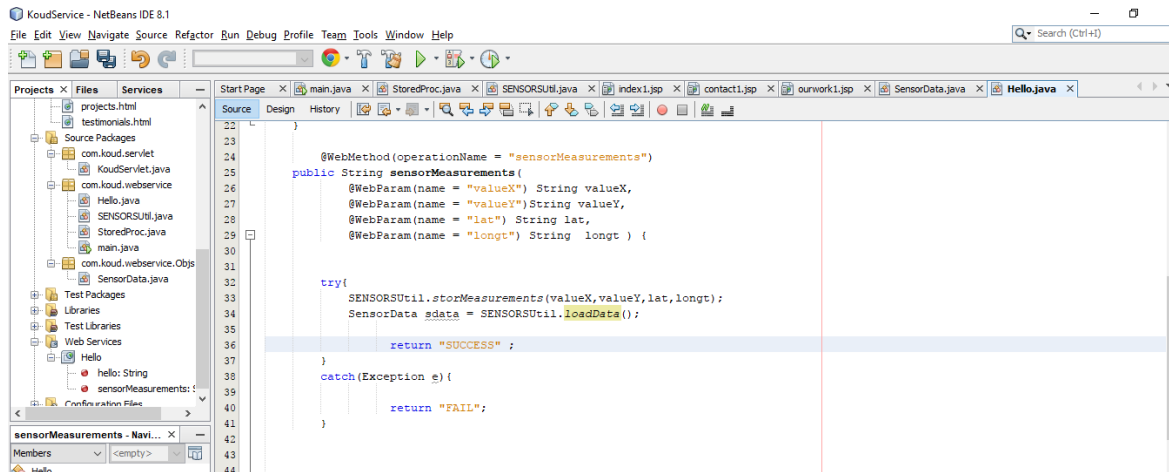
1. Υπηρεσία Ιστού για να δέχεται κλήσεις και να στέλνει απαντήσεις στην κινητή συσκευή.
2. Βάση (Client Oracle) για να αποθηκεύεται η πληροφορία των αισθητήρων.
3. GUI για να μπορεί να είναι προσπελάσιμη αυτή η πληροφορία ανά πάσα στιγμή.

Για να γίνει εφικτή η κλήση του Webservice από την κινητή συσκευή έπρεπε να την κάνουμε publish σε μια συγκεκριμένη στατική ip. Γι αυτό και επιλέξαμε να φιλοξενηθεί η εφαρμογή σε εξυπηρετητή στο cloud, ώστε να είναι διαθέσιμη 24 ώρες το 24ωρο και 7 ημέρες τη βδομάδα. Την στατική ip την έχουμε ορίσει hardcoded μέσα στην εφαρμογή Android, και έχουμε αποφασίσει να αλλάζει μόνο σε επερχόμενα update που θα γίνονται επί αυτής. Επιλύουμε έτσι το πρόβλημα να έχει γίνει κάποια αλλαγή στο server της εφαρμογής ιστού και να μην είναι εφικτό να επικοινωνήσουν οι κινητές συσκευές μαζί του.



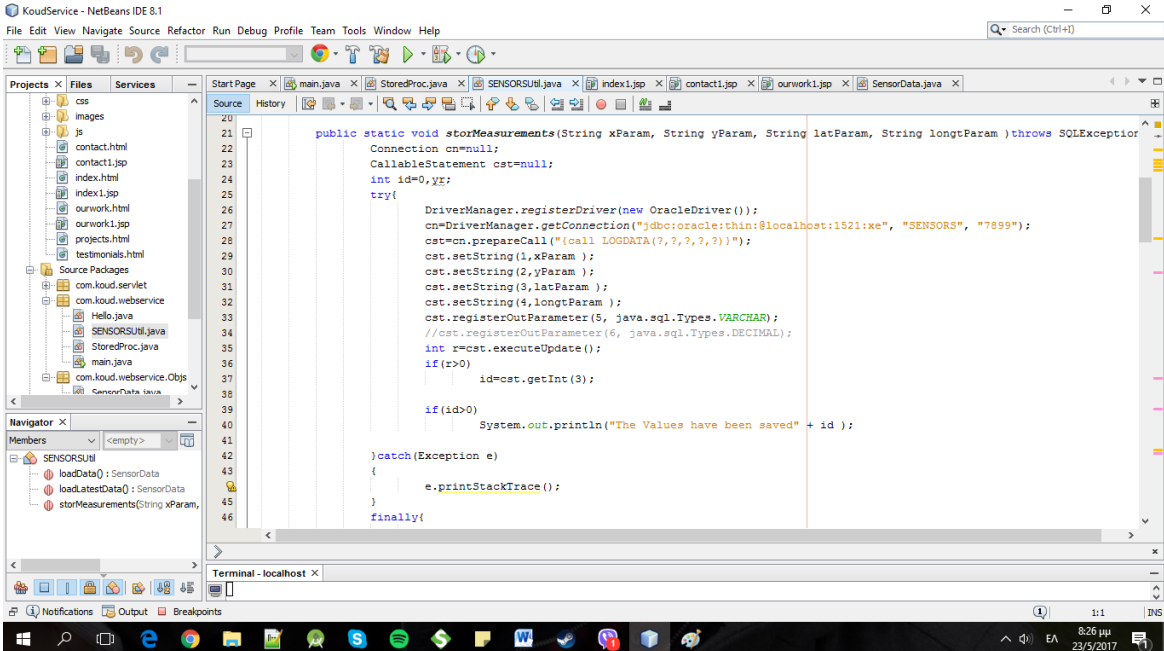
Εικόνα 36 Publish to Web Service σε στατική IP

Στην εφαρμογή Android (Κεφ. 4) χρησιμοποίησαμε τη βιβλιοθήκη SOAP προκειμένου να στείλουμε μηνύματα στο WebService. Τα μηνύματα αυτά είναι γραμμένα σε xml, σύμφωνα με το wsdl που έχει δημιουργηθεί για το συγκεκριμένο Web Service. Τα μηνύματα αυτά “καταναλώνονται” από το operation: sensorsMeasurements. Έχουμε δημιουργήσει μια κλάση την SENSORSUtil, η οποία περιέχει όλες τις μεθόδους που καλούνται για την επεξεργασία και αποθήκευση των μηνυμάτων. Από την παραπάνω κλάση, σε κάθε κλήση που γίνεται στο WebService καλείται η μέθοδος loadData.



Εικόνα 37 Κλήση του operation sensorMeasurements

Όπως φαίνεται στο κώδικα της Εικόνας 38, πραγματοποιείται μια σύνδεση στη βάση, ώστε να κληθεί η procedure (storeMeasurements) που είναι υπεύθυνη για την αποθήκευση των μετρήσεων. Πιο συγκεκριμένα ανοίγει μία σύνδεση στη βάση, και γίνεται κλήση της procedure LOGDATA με input τις τιμές που πήραμε από το xml. Η αποθήκευση των τιμών γίνεται κάθε τρία λεπτά, σύμφωνα με τον timer που έχουμε θέσει στην εφαρμογή Android. Εφόσον όλα πάνε καλά και δεν υπάρξει κάποιο exception είτε στη Java είτε στην ORACLE PL/SQL, απελευθερώνεται η σύνδεση για να μην καταναλώνονται πόροι στο σύστημα.



```
20
21
22     public static void storeMeasurements(String xParam, String yParam, String latParam, String longtParam )throws SQLException
23     {
24         Connection cn=null;
25         CallableStatement cst=null;
26         int id=0,yΣ;
27         try{
28             DriverManager.registerDriver(new OracleDriver());
29             cn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "SENSORS", "7699");
30             cst=cn.prepareCall("{call LOGDATA(?,?,?,?,?)}");
31             cst.setString(1,xParam );
32             cst.setString(2,yParam );
33             cst.setString(3,latParam );
34             cst.setString(4,longtParam );
35             cst.registerOutParameter(5, java.sql.Types.VARCHAR);
36             //cst.registerOutParameter(6, java.sql.Types.DECIMAL);
37             int r=cst.executeUpdate();
38             if(r>0)
39                 id=cst.getInt(3);
40
41             if(id>0)
42                 System.out.println("The Values have been saved" + id );
43         }catch(Exception e)
44         {
45             e.printStackTrace();
46         }
47     }
48     finally{
```

Εικόνα 38 Αποθήκευση των τιμών από τους Αισθητήρες στη βάση

6.2 ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΔΙΕΠΑΦΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΙΣΤΟΥ ΜΕ ΤΟ ΧΡΗΣΤΗ

Η εφαρμογή που θα φιλοξενηθεί στο σύννεφο, περιλαμβάνει, όπως περιγράψαμε σε προηγούμενη ενότητα την υπηρεσία ιστού και την βάση ORACLE. Ωστόσο για να είναι λειτουργική και προς το άτομο που επιθυμεί να γνωρίζει τη θέση του χρήστη που έχει εγκατεστημένη την Android εφαρμογή, απαραίτητο κρίθηκε να υπάρξει διεπαφή που να εμφανίζει αυτή ακριβώς την πληροφορία.



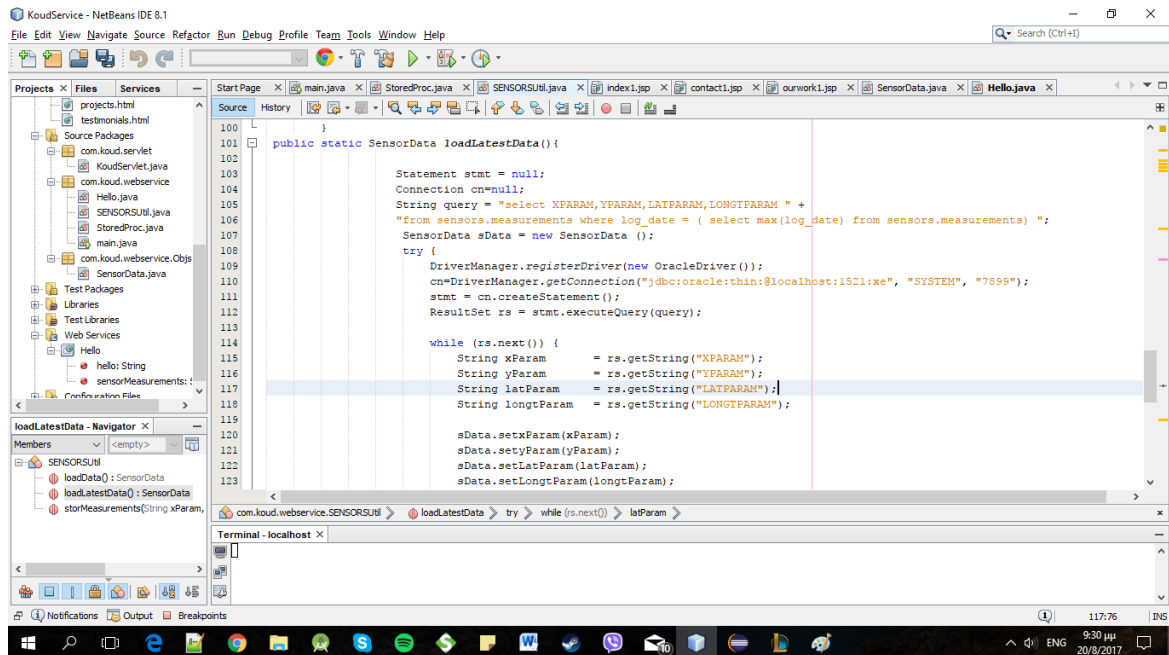
Εικόνα 39 Αρχική σελίδα της Web Εφαρμογής

Η Εικόνα 39 αποτελεί την αρχική σελίδα που θα δει ο χρήστης όταν θα μπει για πρώτη φορά στην εφαρμογή. Από αυτή τη σελίδα, επιλέγοντας το σύνδεσμο «Εντοπισμός», θα μεταφερθεί σε εκείνη που δείχνει το τελευταίο στίγμα θέσης που έχει στείλει η κινητή συσκευή.



Εικόνα 40 Εμφάνιση της τελευταίας χρονικά θέσης της κινητής συσκευής

Έχουμε προσθέσει μια νέα λειτουργικότητα, με το κουμπί «Εκ νέου Εντοπισμός», όπου πατώντας το ο χρήστης κάνει κλήση στη βάση προκειμένου να ληφθούν οι τελευταίες τιμές του GPS αισθητήρα, και στη συνέχεια τα εμφανίζει στο χάρτη. Ουσιαστικά έχουμε δημιουργήσει μια νέα μέθοδο, η οποία παίρνει την τελευταία χρονικά εισαχθείσα εγγραφή στον πίνακα και επιστρέφει ένα αντικείμενο της κλάσης `SensorData`, το οποίο περιέχει τις τιμές που αντιστοιχούν στο γεωγραφικό μήκος και πλάτος.

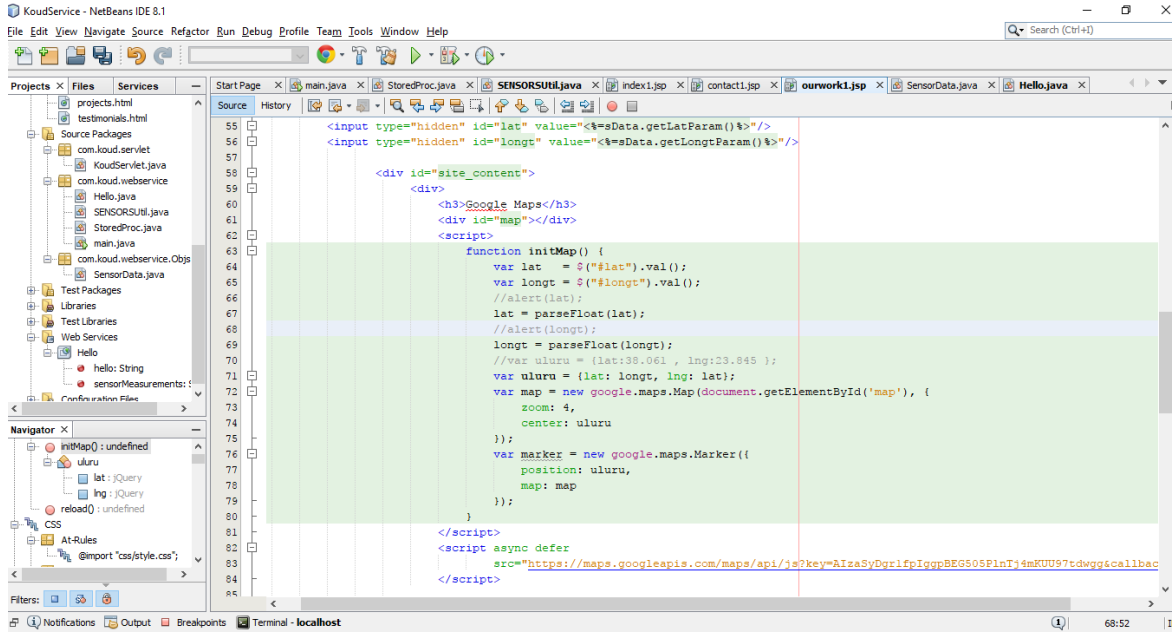


Εικόνα 41 Κλήση της Μεθόδου loadLatestData

Για τη εμφάνιση του χάρτη, χρησιμοποιήσαμε την βιβλιοθήκη που μας παρέχει η GOOGLE για τα API GOOGLE MAPS. Παραμετροποιήσαμε το API αυτό ώστε να μπορούμε να εισάγουμε την πληροφορία από τη βάση, και να εμφανίζουμε σε πραγματικό χρόνο τη θέση της συσκευής. Αυτή η αλλαγή έγινε μέσα από κώδικα JQuery. Μέσα στη jsp θέσαμε δύο μεταβλητές ως hidden:

1. `<input type="hidden" id="lat" value="<%=sData.getLatParam()%>"/>`
2. `<input type="hidden" id="longt" value="<%=sData.getLongtParam()%>"/>`

Ο κώδικας της JQuery, «βλέπει» τις τιμές που έχουν οι παραπάνω μεταβλητές, (όπως φαίνεται παίρνουν τις τιμές από το αντικείμενο `sData` που μας επέστρεψε η `loadLatest`), και τις εισάγει σε δύο αντικείμενα JSON, ώστε να εμφανιστούν στη σελίδα.



The screenshot shows the NetBeans IDE interface with the following components:

- Project Explorer:** Shows a project structure with files like `projects.html`, `testimonials.html`, and source packages for `com.koud.servlet`, `com.koud.webservice`, and `com.koud.webservice.Obsj`.
- Source Editor:** Displays the code for `ourwork1.jsp`. The code includes:
 - Two hidden input fields for `lat` and `longt` with values from `sData`.
 - A `<div id="site_content">` block containing:
 - `<h3>Google Maps</h3>`
 - `<div id="map"></div>`
 - A `<script>` block with a `function initMap()` that:
 - Retrieves `lat` and `longt` from the inputs.
 - Parses the values to floats.
 - Calculates `uluru` based on the coordinates.
 - Initializes a `google.maps.Map` with `uluru` as the center.
 - Creates a `google.maps.Marker` at `uluru`.
 - A `<script async defer src="https://maps.googleapis.com/maps/api/js?key=AizaSyDgrlfpIggpBEGS05FlntJ4mKU9?cdwgg&callbac">` tag.
- Navigator:** Shows the `initMap()` function and its variables: `uluru`, `lat`, and `lng`.
- Terminal:** Shows `Terminal - localhost`.

Εικόνα 42 Ενημέρωση του API της Google για την εμφάνιση της τελευταίας Θέσης

ΚΕΦΑΛΑΙΟ 7: ΙΔΙΩΤΙΚΟΤΗΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η χρήση των Smartphones έχει γίνει αναπόσπαστο κομμάτι της ζωής εκατομμυρίων ανθρώπων σε όλο τον κόσμο. Οι περισσότερες από αυτές τις συσκευές έχουν τουλάχιστον πέντε αισθητήρες από τους οποίους ένας προγραμματιστής μπορεί να εξάγει πληροφορίες για τη συγκεκριμένη συσκευή. Γίνεται λοιπόν κατανοητό ότι πρέπει να γίνει μια αναφορά για το αν η Android εφαρμογή που παρουσιάστηκε σε αυτή τη μεταπτυχιακή διατριβή «προσβάλει» την ιδιωτικότητα του χρήστη και αν μπορεί να εκμεταλλευτεί κάποιος την πληροφορία που παρέχει.

Από τη αρχή της δημιουργίας της η “Help The Old Ones” (Κεφάλαιο 4) είναι μια εφαρμογή που κάνει χρήση δεδομένων από του αισθητήρες της συσκευής, με μοναδικό στόχο να κάνει πιο εύκολη την αποτροπή περιπτώσεων όπου ο χρήστης βρίσκεται σε καταστάσεις σημαντικής ανάγκης όπως να άγνοια της θέσης του ή πτώση. Καταστάσεις δύσκολες όταν τα άτομα αυτά ανήκουν στις ευαίσθητες κοινωνικές κατηγορίες όπως παιδιά και ηλικιωμένοι. Αυτός ήταν άλλωστε και ο λόγος που ξεκινήσαμε να υλοποιήσουμε μια εφαρμογή σαν αυτή.

Από τη στιγμή που μπαίνει όμως στο παιχνίδι η πληροφορία της ακριβούς θέσης της συσκευής αλλά και των κινήσεων που κάνει αυτή είναι εύλογο να δημιουργηθούν ερωτήματα για το πόσο δίνει ελευθερία κινήσεων σε εκείνον που τη χρησιμοποιεί, ή πως μπορεί να τη χρησιμοποιήσει αυτή τη πληροφορία το άτομο που έχει πρόσβαση προκειμένου να βλάψει το χρήστη.

Η εφαρμογή έχει δημιουργηθεί καθαρά για να εξυπηρετήσει τα άτομα που έχουν δυσκολίες στη χρήση του λειτουργικού Android αλλά και να δώσει τη δυνατότητα στα άτομα που πραγματικά ενδιαφέρονται για τον εκάστοτε χρήστη να γνωρίζει κάθε στιγμή τη θέση του ώστε να μπορεί να το βοηθήσει αν υπάρξει πτώση ή άγνοια θέσης.

Μπορεί κανείς να φανταστεί πόσο χρήσιμη δύναται να φανεί η συγκεκριμένη εφαρμογή σε άτομα που έχουν σοβαρά προβλήματα υγείας και συχνά αντιμετωπίζουν δύσκολες καταστάσεις. Ενδεικτικά παρουσιάζουμε δύο από τις πολλές αυτές κατηγορίες ώστε να δείξουμε τη σημαντικότητα της:

1. Άτομο με Άνοια
2. Άτομο με Επιληψία

Είναι λοιπόν σημαντικό να κατανοήσουμε ότι η συγκεκριμένη εφαρμογή αποτελεί ένα σημαντικό εργαλείο για να βοηθήσουμε τους συνανθρώπους μας και όχι απλά να «κατασκοπεύουμε» κάθε κίνησή τους.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Καθ' όλη τη διάρκεια υλοποίησης της συγκεκριμένης εφαρμογής βρεθήκαμε αρκετές φορές μπροστά στη σημαντικότητα που έχει αποκτήσει όλα αυτά τα χρόνια το κινητό τηλέφωνο. Οι δυνατότητες που έχουν το 2017 οι κινητές συσκευές είναι τεράστιες και η χρήση τους μπορεί να κάνει πολύ πιο εύκολη τη ζωή μας, από το να αποθηκεύσουμε κάποιες επαφές που δεν μπορούμε να θυμηθούμε ή φοβόμαστε να γράψουμε σε ένα κομμάτι χαρτί σε περίπτωση που χαθεί, μέχρι την ανίχνευση κίνησης και αυτόματης ειδοποίησης επαφής έκτακτης ανάγκης.

Οι αισθητήρες που φέρουν όλες οι κινητές συσκευές είναι ικανές για έναν έμπειρο προγραμματιστή να μπορέσει να χρησιμοποιήσει τα δεδομένα που στέλνουν, ώστε να κάνει ακόμη μεγαλύτερη τη χρηστική ικανότητα τους και να παρέχει λειτουργίες που θα αποτρέπουν απρόβλεπτες καταστάσεις.

Μεταφέροντας ένα κομμάτι της εφαρμογής στο υπολογιστικό νέφος κατανοήσαμε πως νοικιάζοντας υποδομή γλιτώσαμε χρόνο, χρήμα αλλά και κόπο για να σετάρουμε ένα λειτουργικό σύστημα, μόνο και μόνο για να τρέξει μια εφαρμογή. Μάλιστα ανάλογα με το πόσο εξελίσαμε την εφαρμογή, μπορούσαμε να αυξήσουμε ή να μειώσουμε την υπολογιστική ισχύ του μηχανήματος που νοικιάζαμε.

Πλέον καθίσταται «παλαιομοδίτικη» η πολιτική να υπάρχει ένας server (μηχάνημα) για την Web εφαρμογή, το οποίο θα συντηρείται από τρίτους αμφιβόλου κατάστασης και ασφάλειας, όταν υπάρχει η δυνατότητα ενοικίασης dedicated server από την Amazon(EC2) για παράδειγμα σε πολύ ανταγωνιστικές τιμές.

ΒΕΛΤΙΩΣΕΙΣ

Η εφαρμογή αυτή καθ' αυτή δίνει τις βάσεις να μπορέσει να βελτιωθεί ακόμη περισσότερο τόσο σε επίπεδο εμφάνισης όσο και λειτουργιών. Ως προς το επίπεδο εμφάνισης θα μπορούσε να γίνει σχεδιασμός των εικονιδίων από σχεδιαστική ομάδα ώστε να κάνει ακόμη πιο κατανοητό το τι αντιπροσωπεύει κάθε κουμπί της εφαρμογής. Ως προς το επίπεδο λειτουργιών της Android εφαρμογής υπάρχουν και μπορούν να γίνουν αρκετές και σημαντικές βελτιώσεις

- Φωνητικές λειτουργίες σε πολλές γλώσσες.
- Προσθήκη επιπλέον βασικών λειτουργιών όπως ραδιόφωνο ή αριθμομηχανή.
- Χρήση της κάμερας για αποστολή φωτογραφίας σε καταστάσεις έκτακτης ανάγκης.
- Περισσότερα πειράματα ώστε τα κατώφλια να είναι όσο το δυνατόν πιο ακριβή.

Βελτιώσεις θα μπορούσαν να γίνουν και στην φιλοξενούμενη στο νέφος εφαρμογή. Σε αυτή την έκδοση έχουμε βάλει να εμφανίζεται μόνο η θέση της συσκευής. Επόμενη έκδοση θα μπορούσε να δείχνει τις περιστροφές που κάνει η κινητή συσκευή ή της αλλαγές ύψους που μπορεί να εμφανίζονται, επίσης να στέλνει email με τις τιμές των αισθητήρων, να γίνεται επιλογή ακτίνας κίνησης για τη συγκεκριμένη συσκευή και ειδοποίηση σε περίπτωση που ξεπεραστεί το όριο. Επίσης να μπορεί να χρησιμοποιηθεί από πολλούς χρήστες ταυτόχρονα, καθώς τη δεδομένη στιγμή λειτουργεί για έναν χρήστη και μία συσκευή Android.

Γίνεται λοιπόν κατανοητό ότι μπορούμε να συνεχίσουμε αυτή τη προσπάθεια ώστε να δημιουργήσουμε μια πολύ ισχυρή εφαρμογή που παρέχει πολλές δυνατότητες είτε αυτές έχουν να κάνουν με την κινητή συσκευή είτε με την Desktop Εφαρμογή. Προσπάθεια που ξεφεύγει από τα όρια μιας μεταπτυχιακής διατριβής.

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΡΘΡΟΓΡΑΦΙΑ

- Neil Smyth. Android Studio Essentials.
- Jonathan Simon. Head First Android Development.
- Reto Meier. Professional Android 4 Application Development.
- Paul Deitel, Harvey Deitel, Abbey Deitel, Michael Morgano. Android for Programmers, an App-Driven Approach. 1st Edition for the New Generation of Mobile Devices.
- Jason Morris. Android User Interface Development: Beginner's Guide.
- CSCC. Cloud Customer Architecture for Web Application Hosting.
- Hurwitz J, Bloor R, Kaufman M, Halper F. Cloud computing for dummies.
- Plummer DC, Bittman TJ, Austin T, Clearly D, Smith DM. Cloud computing: Defining and describing and emerging phenomenon.
- Lau W. An introduction to Cloud Computing Characteristics and Service/Deployment Models.

ΔΙΑΔΙΚΤΥΑΚΟΙ ΤΟΠΟΙ

- Techterms.com
- www.Stackoverflow.com
- www.developer.android.com/training
- www.Android-developers.blogspot.gr
- www.soapui.org
- www.w3Schools.com
- www.cloud.google.com/tools/android-studio/docs
- www.aws.amazon.com
- Scholar.google.com
- www.sitepoint.com/using-android-sensors-application/

Παράρτημα Α

Κώδικας εφαρμογής Android

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ekoudcom.harris.helptheoldones" >

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="com.KaKou.project.permission.MAPS_RECEIVE" />
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <permission android:name="android.permission.FLASHLIGHT"
        android:permissionGroup="android.permission-group.HARDWARE_CONTROLS"
        android:protectionLevel="normal" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.MICROPHONE" />
    <uses-feature android:name="android.hardware.camera" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/htoo"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".StarterActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".Phone"
            android:label="" >
        </activity>
        <activity
            android:name=".MainMenu"
            android:label="" >
        </activity>
        <activity
            android:name=".Contact"
            android:label="" >
        </activity>
        <activity
            android:name=".Message"
            android:label="" >
```

```

</activity>
<activity
  android:name=".Email"
  android:label="" >
</activity>
<activity
  android:name=".Emergency"
  android:label="" >
</activity>
<activity
  android:name=".FallDown"
  android:label="" >
</activity>
<activity
  android:name=".Flash"
  android:label="" >
</activity>
</application>

</manifest>

```

ContactLayout.xml

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
  android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity"
  android:background="@color/accent_material_dark">

```

```

<TextView
  android:layout_width="wrap_content"
  android:layout_height="50dp"
  android:id="@+id/viewName"
  android:layout_alignParentTop="true"
  android:layout_alignParentLeft="true"
  android:layout_alignParentStart="true"
  android:text="Add Name:"
  android:textColor="@color/material_blue_grey_800"
/>

```

```

<TextView
  android:layout_width="wrap_content"
  android:layout_height="50dp"
  android:id="@+id/viewNumber"
  android:layout_below="@+id/viewName"
  android:layout_alignParentLeft="true"
  android:text="Add Number:"
  android:textColor="@color/material_blue_grey_800"/>

```

```

<Button
  android:layout_width="wrap_content"
  android:layout_height="match_parent"

```

```
android:text="Add"
android:id="@+id/Add"
android:layout_alignTop="@+id/viewName"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:onClick="onClick_AddContact"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"
android:layout_alignBottom="@+id/contactNumber" />
```

```
<Button
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Clear All Data"
android:id="@+id/button2"
android:layout_below="@+id/viewNumber"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true"
android:layout_marginTop="1 dp"
android:onClick="clearAll"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"/>
```

```
<ListView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/listView"
android:layout_below="@+id/button2"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
/>
```

```
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/contactName"
android:layout_alignParentTop="true"
android:layout_toRightOf="@+id/viewNumber"
android:hint="Name"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
/>
```

```
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/contactNumber"
android:layout_alignTop="@+id/viewNumber"
android:layout_toRightOf="@+id/viewNumber"
android:layout_toEndOf="@+id/viewNumber"
android:hint="Number"
android:inputType="number"
```

```
    android:textColor="@color/abc_primary_text_disable_only_material_dark"/>
</RelativeLayout>
```

Email_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/accent_material_dark">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="@string/Recipient"/>
    <EditText
        android:layout_width="match_parent"
        android:id="@+id/recipientEmail"
        android:layout_height="70dp"
        android:hint="@string/Here"
        android:background="@color/background_floating_material_light"
        android:textColor="@color/accent_material_dark"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="@string/TextToSend"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="209dp"
        android:id="@+id/emailText"
        android:textSize="20dp"
        android:background="@color/background_floating_material_light"
        android:textColor="@color/accent_material_dark"
        android:hint="@string/Here"
        android:textAlignment="gravity"
    />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/messageSendButton"
        android:text="@string/SendMessage"
        android:layout_gravity="center"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:background="@color/material_blue_grey_800"/>

</LinearLayout>
```

Emergency layout.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"

    tools:context="edu.uci.ics.orientationproj.MainActivity">

    <EditText android:id="@+id/editText1" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
        android:ems="10" android:inputType="numberDecimal" android:labelFor="@+id/editText1">

        <requestFocus />
    </EditText>

    <EditText android:id="@+id/editText2" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_below="@+id/editText1"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
        android:ems="10" android:inputType="numberDecimal" android:labelFor="@+id/editText2" />

    <EditText android:id="@+id/editText3" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_below="@+id/editText2"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
        android:ems="10" android:inputType="numberDecimal" android:labelFor="@+id/editText3" />
    <EditText android:id="@+id/editText4" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_below="@+id/editText3"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
        android:ems="10" android:inputType="numberDecimal" android:labelFor="@+id/editText4" />

    <EditText android:id="@+id/textLong" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_below="@+id/editText4"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
        android:ems="10" android:inputType="numberDecimal" android:labelFor="@+id/Long" />

    <EditText android:id="@+id/textLat" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_below="@+id/textLong"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
        android:ems="10" android:inputType="numberDecimal" android:labelFor="@+id/Lat" />

    <EditText android:id="@+id/textZ" android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_below="@+id/textLat"
        android:layout_centerHorizontal="true" android:layout_marginTop="20dp"
```



```
android:ems="10" android:inputType="numberDecimal" />
```

```
</RelativeLayout>
```

FallDown layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:text="New Button"
        android:id="@+id/button"
        android:background="@drawable/falldown"/>
</LinearLayout>
```

Flash layout.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity"
    android:background="@color/accent_material_dark">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="80dp"
        android:text="Speak"
        android:id="@+id/btn_speak"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true" />

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/list"
        android:layout_below="@+id/btn_speak"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_marginTop="87dp"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:background="@color/material_blue_grey_800">

</ListView>
```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:text="Open"
    android:id="@+id/open"
    android:layout_toLeftOf="@+id/btn_speak"
    android:layout_toStartOf="@+id/btn_speak"
    android:onClick="openFlash"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"
    android:layout_above="@+id/list" />

```

```

<Button
    android:layout_width="wrap_content"
    android:layout_height="80dp"
    android:text="Close"
    android:id="@+id/close"
    android:layout_alignTop="@+id/btn_speak"
    android:layout_toRightOf="@+id/btn_speak"
    android:layout_toEndOf="@+id/btn_speak"
    android:onClick="closeFlash"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"/>

```

```

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Change EMERGENCY CONTACT"
    android:id="@+id/button10"
    android:layout_marginTop="2dp"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"
    android:onClick="openStarter"
    android:layout_below="@+id/btn_speak"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
</RelativeLayout>

```

Item layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/text1"
        android:textSize="30dp"/>

    <TextView
        android:layout_width="wrap_content"

```

```

    android:layout_height="wrap_content"
    android:text=": "
    android:textSize="30dp"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/text2"
    android:textSize="30dp"/>
</LinearLayout>

```

Main menu.xml

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity"
    android:orientation="vertical"
    android:background="@color/accent_material_dark">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginRight="0dp"
    android:layout_marginLeft="0dp"
    android:layout_weight="1">
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_marginRight="1 dp"
        android:id="@+id/phone"
        android:text="Phone"
        android:background="@drawable/phone"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:id="@+id/message"
        android:text="Message"
        android:background="@drawable/message"/>
</LinearLayout>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1">
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"

```

```
        android:id="@+id/contacts"
        android:text="Contacts"
        android:layout_marginRight="1 dp"
        android:layout_marginTop="1 dp"
        android:background="@drawable/contacts"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:id="@+id/email"
        android:layout_marginTop="1 dp"
        android:text="Email"
        android:background="@drawable/mail"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1">
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_marginTop="1 dp"
        android:id="@+id/gallery"
        android:layout_marginRight="1 dp"
        android:text="Gallery"
        android:background="@drawable/gallery"/>
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_marginTop="1 dp"
        android:id="@+id/camera"
        android:text="Camera"
        android:background="@drawable/camera"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1">
    <Button
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:layout_marginTop="1 dp"
        android:id="@+id/emergency"
        android:text="Emergency"
        android:layout_marginRight="1 dp"
        android:background="@drawable/emergency"/>
    <Button
```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:layout_marginTop="1 dp"
    android:id="@+id/flash"
    android:text="Flash"
    android:background="@drawable/flash"/>
</LinearLayout>

</LinearLayout>

```

Message layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/accent_material_dark">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="@string/Recipient"/>
    <EditText
        android:layout_width="match_parent"
        android:id="@+id/recipientNumber"
        android:layout_height="70dp"
        android:hint="@string/Here"
        android:inputType="number"
        android:background="@color/background_floating_material_light"
        android:textColor="@color/accent_material_dark"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="30dp"
        android:text="@string/TextToSend"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="209dp"
        android:id="@+id/messageText"
        android:background="@color/background_floating_material_light"
        android:textColor="@color/accent_material_dark"
        android:textSize="20dp"
        android:hint="@string/Here"
        android:textAlignment="gravity"
    />
    <Button
        android:layout_width="match_parent"
        android:layout_height="50dp"

```

```
    android:id="@+id/messageSendButton"
    android:text="@string/SendMessage"
    android:layout_gravity="center"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"/>
```

```
</LinearLayout>
```

Phone layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/accent_material_dark">
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_height="90dp"
            android:layout_width="match_parent"
            android:layout_weight="3"
            android:id="@+id/textViewCallNumber"
            android:layout_marginTop="40dp"
            android:background="@color/background_floating_material_light"
            android:textColor="@color/accent_material_dark"
            android:textSize="45dp"
        />
        <Button
            android:layout_width="70dp"
            android:layout_height="match_parent"
            android:text="X"
            android:layout_marginTop="50dp"
            android:id="@+id/buttonDelete"
            android:textColor="@color/abc_primary_text_disable_only_material_dark"
            android:background="@color/material_blue_grey_800"/>
    </LinearLayout>
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:layout_marginTop="90dp">
```

```
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="70dp">
        <Button
```

```
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:text="1"
android:layout_weight="1"
android:id="@+id/button1"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"
android:layout_marginRight="1dp"
/>
<Button
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:text="2"
android:layout_weight="1"
android:id="@+id/button2"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"
android:layout_marginRight="1dp"/>

<Button
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:text="3"
android:layout_weight="1"
android:id="@+id/button3"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"/>
</LinearLayout>//123

<LinearLayout
android:layout_width="match_parent"
android:layout_height="70dp">
<Button
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:text="4"
android:layout_marginTop="1dp"
android:layout_marginRight="1dp"
android:layout_weight="1"
android:id="@+id/button4"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"/>
<Button
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:text="5"
android:layout_marginTop="1dp"
android:layout_marginRight="1dp"
android:layout_weight="1"
android:id="@+id/button5"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"/>
```

```
<Button
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:text="6"
  android:layout_marginTop="1dp"
  android:layout_weight="1"
  android:id="@+id/button6"
  android:textColor="@color/abc_primary_text_disable_only_material_dark"
  android:background="@color/material_blue_grey_800"/>
</LinearLayout>//456
```

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="70dp">
```

```
<Button
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:text="7"
  android:layout_marginTop="1dp"
  android:layout_marginRight="1dp"
  android:layout_weight="1"
  android:id="@+id/button7"
  android:textColor="@color/abc_primary_text_disable_only_material_dark"
  android:background="@color/material_blue_grey_800"/>
```

```
<Button
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:text="8"
  android:layout_marginTop="1dp"
  android:layout_marginRight="1dp"
  android:layout_weight="1"
  android:id="@+id/button8"
  android:textColor="@color/abc_primary_text_disable_only_material_dark"
  android:background="@color/material_blue_grey_800"/>
```

```
<Button
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:text="9"
  android:layout_marginTop="1dp"
  android:layout_weight="1"
  android:id="@+id/button9"
  android:textColor="@color/abc_primary_text_disable_only_material_dark"
  android:background="@color/material_blue_grey_800"/>
</LinearLayout>//789
```

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="70dp">
  <Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
```



```
        android:text=""
        android:layout_marginTop="1dp"
        android:layout_marginRight="1dp"
        android:layout_weight="1"
        android:id="@+id/buttonStar"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:background="@color/material_blue_grey_800"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="0"
    android:layout_marginTop="1dp"
    android:layout_marginRight="1dp"
    android:layout_weight="1"
    android:id="@+id/button0"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"/>
<Button
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="#"
    android:layout_marginTop="1dp"
    android:layout_weight="1"
    android:id="@+id/buttonSemi"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"/>
</LinearLayout>/*0#

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="60dp">
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="@string/Call"
    android:layout_marginTop="1dp"
    android:layout_marginRight="1dp"
    android:layout_weight="1"
    android:id="@+id/buttonCall"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="@string/Cancel"
    android:layout_weight="1"
    android:layout_marginTop="1dp"
    android:id="@+id/buttonCancelCall"
    android:textColor="@color/abc_primary_text_disable_only_material_dark"
    android:background="@color/material_blue_grey_800"/>
</LinearLayout>//call cancel call
```

```
</LinearLayout>//ola ta linear
```

```
</LinearLayout>
```

StarterActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/accent_material_dark"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:text="WELCOME"
        android:textSize="50dp"
        android:textAlignment="center"
        android:layout_alignBottom="@+id/textView"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="48dp"
        android:id="@+id/textView2" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Emergency Contact:"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:id="@+id/textView"
        android:textSize="30dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="153dp" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Number"
        android:inputType="number"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:id="@+id/emerContact"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save"
        android:id="@+id/saveContact"
        android:textColor="@color/abc_primary_text_disable_only_material_dark"
        android:background="@color/material_blue_grey_800"
```

```

android:onClick="save"

android:layout_below="@+id/emerContact"
android:layout_toLeftOf="@+id/emerContact"
android:layout_toStartOf="@+id/emerContact"
android:layout_marginTop="48dp" />

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Skip"
android:id="@+id/skip"
android:textColor="@color/abc_primary_text_disable_only_material_dark"
android:background="@color/material_blue_grey_800"
android:layout_alignTop="@+id/saveContact"
android:layout_alignRight="@+id/textView2"
android:layout_alignEnd="@+id/textView2"
android:onClick="skip"/>

</RelativeLayout>

```

Contact.java

```
package ekoudcom.harris.helptheoldones;
```

```

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.text.TextUtils;
import android.text.format.Time;
import android.view.Menu;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;
import android.widget.TextView;
import android.widget.Toast;

```

```
public class Contact extends Activity {
```

```

    Time today = new Time(Time.getCurrentTimezone());
    DBAdapter myDb;
    EditText contactName,contactNumber;
    long selectedContact;

```

```
@Override
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.contact_layout);
    contactName =(EditText)findViewById(R.id.contactName);
    contactNumber =(EditText)findViewById(R.id.contactNumber);
    openDB();//otan dhmiourgeitai anoigei th DB
    populateListView();
    listViewItemClickListener();//gia na mporw na kalesw ton itemListener
}

```

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

private void openDB(){
    myDb = new DBAdapter(this);
    myDb.open();
}

public void onClick_AddContact(View v){
    //today.setToNow();
    // String timestamp= today.format("%Y-%M-%D : %H:%M:%S ");
    if(!TextUtils.isEmpty(contactName.getText().toString())){
        if(!TextUtils.isEmpty(contactNumber.getText().toString())) {
            myDb.insertRow(contactName.getText().toString(), contactNumber.getText().toString());
            contactNumber.clearFocus();
            contactName.clearFocus();
        }
        else{
            Toast toast = Toast.makeText(this, "Empty Number", Toast.LENGTH_LONG);
            toast.show();
        }
    }
    else{
        Toast toast = Toast.makeText(this, "Empty Name", Toast.LENGTH_LONG);
        toast.show();
    }
    populateListView();
}

private void populateListView(){
    Cursor cursor = myDb.getAllRows();
    String [] fromFieldNames = new String[] {DBAdapter.KEY_NAME,DBAdapter.KEY_NUMBER};
    int [] toViewIds = new int [] {R.id.text1,R.id.text2};
    SimpleCursorAdapter myCursorAdpater;
    myCursorAdpater = new
SimpleCursorAdapter(getBaseContext(),R.layout.item_layout,cursor,fromFieldNames,toViewIds,0);
    ListView myList=(ListView)findViewById(R.id.listView);
    myList.setAdapter(myCursorAdpater);
}

private void updateContact(long id){
    Cursor cursor= myDb.getRow(id);
    if (cursor.moveToFirst()) {
        String name= contactName.getText().toString();
        String number= contactNumber.getText().toString();
        //today.setToNow();
        //String date=today.format("%Y-%M-%D : %H:%M:%S ");
        myDb.updateRow(id,name,number);
    }

    cursor.close();
}

public void clearAll(View v){
    myDb.deleteAll();
    populateListView();
}

private void listViewItemClick(){
    ListView myList = (ListView)findViewById(R.id.listView);
    myList.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override

```

```

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    Cursor cursor = myDb.getRow(id);
    String number = cursor.getString(2);
    Intent intent = new Intent(Intent.ACTION_CALL);
    intent.setData(Uri.parse("tel:" + number));
    startActivity(intent);
    //updateContact(id);
    populateListView();
}
});
}
}

```

DBAdapter.java

```

package ekoudcom.harris.helpeoldones;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;

public class DBAdapter {
    private static final String TAG = "BDAdapter";

    public static final String KEY_ROWID = "_id";
    public static final String KEY_NAME = "name";
    public static final String KEY_NUMBER = "number";
    public static final String[] ALL_KEYS = new String[]{KEY_ROWID, KEY_NAME, KEY_NUMBER};

    public static final int COL_ROWID = 0;
    public static final int COL_NAME = 1;
    public static final int COL_NUMBER = 2;

    public static final String DATABASE_NAME = "dbToDo";
    public static final String DATABASE_TABLE = "mainToDo";
    public static final int DATABASE_VERSION = 2;

    private static final String DATABASE_CREATE_SQL =
        " CREATE TABLE " + DATABASE_TABLE
        + " ( " + KEY_ROWID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
        + KEY_NAME + " TEXT NOT NULL," + KEY_NUMBER + " TEXT" + ")";

    private final Context context;
    private DatabaseHelper myDBHelper;
    private SQLiteDatabase db;

    public DBAdapter(Context ctx) {
        this.context = ctx;
        myDBHelper = new DatabaseHelper(context);
    }

    public DBAdapter open() {
        db= myDBHelper.getWritableDatabase();
        return this;
    }

    public void close() {
        myDBHelper.close();
    }

    public long insertRow(String name, String number) {

```

```

ContentValues initialValues = new ContentValues();
initialValues.put(KEY_NAME, name);
initialValues.put(KEY_NUMBER, number);

return db.insert(DATABASE_TABLE, null, initialValues);
}

public boolean deleteRow(long rowId) {
String where = KEY_ROWID + "=" + rowId;
return db.delete(DATABASE_TABLE, where, null) != 0;
}

public void deleteAll() {

myDBHelper.onDelete(db);
// Cursor c = getAllRows();
//long rowId = c.getColumnIndexOrThrow(KEY_ROWID);
//if (c.moveToNext()) {
// do {
// deleteRow(c.getLong((int) rowId));
//} while (c.moveToNext());
//}
}

public Cursor getAllRows() {
String where = null;
Cursor c = db.query(true, DATABASE_TABLE, ALL_KEYS, where, null, null, null, null);

if (c != null) {
c.moveToNext();
}
return c;
}

public boolean updateRow(long rowId, String name, String number) {
String where = KEY_ROWID + "=" + rowId;
ContentValues newValues = new ContentValues();
newValues.put(KEY_NAME, name);
newValues.put(KEY_NUMBER, number);
return db.update(DATABASE_TABLE, newValues, where, null) != 0;
}

public Cursor getRow(long id) {
String where = KEY_ROWID + "=" + id;
Cursor c = db.query(true, DATABASE_TABLE, ALL_KEYS, where, null, null, null, null);
if(c!=null){
c.moveToFirst();
}
return c;
}

private static class DatabaseHelper extends SQLiteOpenHelper {
DatabaseHelper(Context context) {
super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

@Override
public void onCreate(SQLiteDatabase db) {
db.execSQL(DATABASE_CREATE_SQL);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
Log.w(TAG, "Upgrading application's database from version"+ oldVersion
+ "to" + newVersion + ", which will destroy all old data");
}
}

```

```

db.execSQL("DROP TABLE IF EXIST" + DATABASE_TABLE);

onCreate(db);
}
public void onDelete(SQLiteDatabase db) {
db.execSQL("DELETE FROM " + DATABASE_TABLE);
}
}
}

```

Email.java

```

package ekoudcom.harris.helptheoldones;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

/**
 * Created by Harris on 6/29/2015.
 */
public class Email extends Activity {
    Button send;
    EditText recipient,message;
    Intent intent;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.email_layout);

        recipient=(EditText)findViewById(R.id.recipientEmail);
        message=(EditText)findViewById(R.id.emailText);
        send= (Button)findViewById(R.id.messageSendButton);
        send.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                String url= recipient.getText().toString();
                String text=message.getText().toString();
                intent = new Intent(Intent.ACTION_SEND);
                intent.putExtra(Intent.EXTRA_EMAIL, url);
                intent.putExtra(Intent.EXTRA_SUBJECT, "Email");
                intent.putExtra(Intent.EXTRA_TEXT, text);
                intent.setType("message/rfc822");
                //chooser=Intent.createChooser(intent,"Harris");
                startActivity(intent);
            }
        });
    }
}

```

Emergency.java

```

package ekoudcom.harris.helptheoldones;

import android.app.Activity;

```

```

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.widget.TextView;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.SoapFault;
import org.ksoap2.serialization.PropertyInfo;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

import java.io.IOException;
import java.text.DecimalFormat;
import java.util.Timer;
import java.util.TimerTask;

public class Emergency extends Activity implements SensorEventListener {
private TextView xAxisLabel;
private TextView yAxisLabel;
private TextView zAxisLabel;
private TextView longt;
private TextView lat;
private TextView timer;
private TextView texttimerZ;
private SensorManager mSensorManager;
private SensorEventListener mEventListenerOrientation;
private LocationManager lm;
private LocationListener ls;
private SensorManager senSensorManager;
private Sensor senAccelerometer;
private Sensor senGrav;
Timer timerSoap = new Timer();
DecimalFormat df = new DecimalFormat("#.###");

public boolean bkeyPressed = false;

private float xAxis;
private float yAxis;
private float zAxis;
double latitude ;
double longitude ;
float maxX = 0.0f;
float maxY = 0.0f;
float maxZ = 0.0f;
int timerX = 0;
int timerY = 0;
int timerZ = 0;
int soapX = 0;
int soapY = 0;
int soapZ = 0;
int fallCounter= 0;

```



```

private void updateUI() {

    xAxisLabel.setText("" + xAxis);
    yAxisLabel.setText("" + yAxis);
    zAxisLabel.setText("" + zAxis);
    if (xAxis >= 11f) {
        //shake
        timerX++;
        soapX++;
    }

    if (zAxis >= 23.000) {
        fallCounter = 1;
    }
    if (zAxis >= 9.600 && fallCounter>0) {
        //fall
        timerZ++;
        soapZ++;
    }
    timer.setText("" + timerX);
    texttimerZ.setText("" + timerZ); //ptwsi
}

private void soapInfo() {

    if (xAxis >= 6f) {
        soapX++; //shake
    }
    if (zAxis >= 7.000) {
        soapZ++;
    }
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.emergency_layout);

    senSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    senAccelerometer = senSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    senSensorManager.registerListener((SensorEventListener) this, senAccelerometer,
    SensorManager.SENSOR_DELAY_NORMAL);

    senGrav = senSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
    senSensorManager.registerListener((SensorEventListener) this, senGrav,
    SensorManager.SENSOR_DELAY_NORMAL);

    xAxisLabel = (TextView) findViewById(R.id.editText1);
    yAxisLabel = (TextView) findViewById(R.id.editText2);
    zAxisLabel = (TextView) findViewById(R.id.editText3);
    texttimerZ = (TextView) findViewById(R.id.textZ);
    lat = (TextView) findViewById(R.id.textLat);
    longt = (TextView) findViewById(R.id.textLong);
    timer = (TextView) findViewById(R.id.editText4);
    SharedPreferences loadContact = getSharedPreferences("myData", MODE_PRIVATE);
    final String emerContact = loadContact.getString("Number", "199");

    timerSoap.schedule(new TimerTask() {
        @Override
        public void run() {
            try {
                Object outcome = new createSoap().execute().get();
                if(outcome.equals("SUCCESS")){
                    Intent setIntent = new Intent(Emergency.this, MainMenu.class);
                    setIntent.addCategory(Intent.CATEGORY_HOME);

```

```

        setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(setIntent);
    }else{
        outcome = new createSoap().execute().get();
        if(outcome.equals("SUCCESS")){
            Intent setIntent = new Intent(Emergency.this, MainMenu.class);
            setIntent.addCategory(Intent.CATEGORY_HOME);
            setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(setIntent);
        }
    }
}catch (Exception e){
    e.printStackTrace();
}
}, 3*60*1000);

Im = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
try {
    ls = new LocationListener() {

        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
            // TODO Auto-generated method stub

        }

        @Override
        public void onProviderEnabled(String provider) {
            // TODO Auto-generated method stub

        }

        @Override
        public void onProviderDisabled(String provider) {
            // TODO Auto-generated method stub

        }

        @Override
        public void onLocationChanged(Location location) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
            lat.setText(String.valueOf(latitude));
            longt.setText(String.valueOf(longitude));
        }
    };
    Im.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0, ls);

} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
} // onCreate

@Override
public void onResume() {
    super.onResume();
}

@Override
public void onPause() {
    super.onPause();
}
}

```

```

@Override
public void onStop() {
    super.onStop();
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }
    return super.onOptionsItemSelected(item);
}

@Override
public void onBackPressed() {
    Log.d("CDA", "onBackPressed Called");
    KeyEvent bKeyPressed = true;
    Intent setIntent = new Intent(Emergency.this, MainMenu.class);
    setIntent.addCategory(Intent.CATEGORY_HOME);
    setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(setIntent);
}

@Override
public void onSensorChanged(SensorEvent sensorEvent) {
    SharedPreferences loadContact = getSharedPreferences("myData", MODE_PRIVATE);
    final String emerContact = loadContact.getString("Number", "199");
    float[] values = sensorEvent.values;
    xAxis = values[0];
    yAxis = values[1];
    zAxis = values[2];
    updateUI();
    if (timerX >= 50) {
        sensorManager.unregisterListener(this);

        Intent sendIntent = new Intent(Intent.ACTION_VIEW);
        sendIntent.setType("vnd.android-dir/mms-sms");
        sendIntent.putExtra("address", emerContact);
        sendIntent.putExtra("sms_body", "Longitude " + longt.getText().toString() + " Latitude " +
lat.getText().toString());
        sendIntent.putExtra("exit_on_sent", true);
        startActivity(sendIntent);
    }
    if (timerZ >= 500) {
        sensorManager.unregisterListener(this);
        Intent intent = new Intent(Emergency.this, FallDown.class);
        intent.putExtra("lat", lat.getText().toString());
        intent.putExtra("longt", longt.getText().toString());
        startActivity(intent);
    }
}

@Override
public void onAccuracyChanged(Sensor sensor, int i) {

}

public void createSoap() {

}

public void getResponse(SoapSerializationEnvelope envelope) {
    SoapPrimitive resultString = null;

    try {

```

```

    resultString = (SoapPrimitive) envelope.getResponse();
} catch (SoapFault e) {
    e.printStackTrace();
    System.out.println("serverError retrieving the data");
}

```

```

if (resultString != null) {
    System.out.println("***** RESULT: " + resultString);
}
}
}

```

```

private class createSoap extends AsyncTask {
    @Override
    protected String doInBackground(Object[] params) {

        final String SOAP_ACTION = "http://webservice.koud.com/sensorMeasurements";
        final String METHOD_NAME = "sensorMeasurements";
        final String NAMESPACE = "http://webservice.koud.com/";
        final String URL = "http://192.168.43.151:9090/KoudService/Hello?wsdl";
        String data = null;
        SoapPrimitive res = null;
        SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
        System.out.println(request.getNamespace());
        PropertyInfo propInfo1 = new PropertyInfo();
        PropertyInfo propInfo2 = new PropertyInfo();
        PropertyInfo propInfo3 = new PropertyInfo();
        PropertyInfo propInfo4 = new PropertyInfo();
        propInfo1.name = "valueX";
        propInfo1.setValue(1);
        request.addProperty(propInfo1);
        propInfo2.name = "valueY";
        propInfo2.setValue(2);
        request.addProperty(propInfo2);
        propInfo3.name = "lat";
        propInfo3.setValue(df.format(latitude));
        request.addProperty(propInfo3);
        propInfo4.name = "longt";
        propInfo4.setValue(df.format(longitude));
        request.addProperty(propInfo4);

        SoapSerializationEnvelope envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
        envelope.setOutputSoapObject(request);
        HttpTransportSE ht = new HttpTransportSE(URL,20000);
        if (ht == null) {
            System.out.println("Problimaa0");
        }
        try {
            ht.call("\http://webservice.koud.com/sensorMeasurements\", envelope);
            getResponse(envelope);
            SoapPrimitive resultsString = (SoapPrimitive)envelope.getResponse();
            SoapObject result=(SoapObject)envelope.bodyIn;
            data = resultsString.toString();
        } catch (IOException e) {
        } catch (Exception q) {
            q.printStackTrace();
        }

        return data;
    }
}
}
}

```

FallDown.java

```

package ekoudcom.harris.helptheoldones;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.location.LocationListener;
import android.location.LocationManager;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.widget.Button;

public class FallDown extends Activity {
    float longt, lat;
    boolean stopped = false;
    Button btn;
    private LocationManager lm;
    private LocationListener ls;
    public double latitude, longitude;
    MediaPlayer mp = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.falldown_layout);
        mp = MediaPlayer.create(this, R.raw.alarm);
        AudioManager audioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);
        mp.start();

        btn = (Button) findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                mp.stop();
                stopped = true;

                Intent goBack= new Intent (FallDown.this,MainMenu.class);
                startActivity(goBack);
            }
        });
        final Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            @Override
            public void run() {

                if(stopped==false) {
                    SharedPreferences loadContact = getSharedPreferences("myData", MODE_PRIVATE);
                    final String emerContact = loadContact.getString("Number", "199");
                    Bundle extras = getIntent().getExtras();
                    String lat = extras.getString("lat");
                    String longt = extras.getString("longt");
                    Intent sendIntent = new Intent(Intent.ACTION_VIEW);
                    sendIntent.setType("vnd.android-dir/mms-sms");
                    //sendIntent.putExtra(Intent.EXTRA_TEXT, "Longitude " + longt.getText().toString() + " Latitude " +
                    lat.getText().toString());
                    sendIntent.putExtra("address", emerContact);
                    sendIntent.putExtra("sms_body", "I fall down at " + lat + " " + longt);
                }
            }
        }, 1000);
    }
}

```

```

        sendIntent.putExtra("exit_on_sent", true);
        startActivity(sendIntent);
    }

}
}, 22000);

}

@Override
public void onBackPressed() {
    mp.stop();
    stopped=true;
    Log.d("CDA", "onBackPressed Called");
    Intent setIntent = new Intent(FallDown.this,MainMenu.class);
    setIntent.addCategory(Intent.CATEGORY_HOME);
    setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(setIntent);
}

}

```

Flash.java

```

package ekoudcom.harris.helptheoldones;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.hardware.Camera;
import android.os.Bundle;
import android.speech.RecognizerIntent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.Button.OnClickListener;
import android.widget.ListView;

import java.util.ArrayList;

/**
 * Created by harris on 2/11/2015.
 */

public class Flash extends Activity implements OnClickListener {

    public ListView mList;
    public Button speakButton;
    Camera cam;
    boolean open;

    public static final int VOICE_RECOGNITION_REQUEST_CODE = 1234;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.flash_layout);

        speakButton = (Button) findViewById(R.id.btn_speak);
        speakButton.setOnClickListener(this);

        voiceinputbuttons();
    }

```

```

}

public void openFlash(View v) {

    if (open == false) {
        cam = Camera.open();
        Camera.Parameters p1 = cam.getParameters();
        p1.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
        cam.setParameters(p1);
        cam.startPreview();
        open = true;
    }

}

public void closeFlash(View v) {
    if (cam != null) {
        cam.stopPreview();
        cam.release();
        cam = null;
        open = false;
    }
}

public void informationMenu() {
    startActivity(new Intent("android.intent.action.INFOSCREEN"));
}

public void voiceinputbuttons() {
    speakButton = (Button) findViewById(R.id.btn_speak);
    mList = (ListView) findViewById(R.id.list);
}

public void startVoiceRecognitionActivity() {
    Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
        "Speech recognition demo");
    startActivityForResult(intent, VOICE_RECOGNITION_REQUEST_CODE);
}

public void onClick(View v) {
    // TODO Auto-generated method stub
    startVoiceRecognitionActivity();
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == VOICE_RECOGNITION_REQUEST_CODE && resultCode == RESULT_OK) {
        ArrayList matches = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
        mList.setAdapter(new ArrayAdapter(this, android.R.layout.simple_list_item_1, matches));

        if (matches.contains("flashlight")) {
            cam = Camera.open();
            Camera.Parameters p = cam.getParameters();
            p.setFlashMode(Camera.Parameters.FLASH_MODE_TORCH);
            cam.setParameters(p);
            cam.startPreview();
        }
    }
}
}

```

```

public void openStarter(View v) {
    SharedPreferences loadSettings = getSharedPreferences("settings", MODE_PRIVATE);
    SharedPreferences.Editor editor = loadSettings.edit();
    editor.putBoolean("Contact", false);
    editor.commit();
    Intent starter = new Intent(Flash.this, StarterActivity.class);
    startActivity(starter);
}

```

MainMenu.java

```

package ekoudcom.harris.helptheoldones;

```

```

import android.app.Activity;
import android.content.Intent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.util.Log;
import android.view.MotionEvent;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

```

```

public class MainMenu extends Activity {

```

```

    //View elements
    Button phone;
    Button contacts;
    Button message;
    Button email;
    Button camera,gallery;
    Button emergency,flashlight;
    private SensorManager mSensorManager;
    private SensorEventListener mEventListenerOrientation;
    private float zAxis,timerZ;
    private final static int INTERVAL = 1000 * 30 * 1; //2 minutes
    private int sendToCloud = 0;

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_menu);

```

```

        if(sendToCloud >0) {
            Toast.makeText(getApplicationContext(), "UpLoaded to Cloud", Toast.LENGTH_SHORT).show();
        }

```

```

        phone = (Button)findViewById(R.id.phone);
        final MediaPlayer mp= MediaPlayer.create(this, R.raw.phone);
        phone.setOnTouchListener(new View.OnTouchListener() {

```



```

@Override
public boolean onTouch(View v, MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        mp.start();
        Intent intent = new Intent(MainMenu.this, Phone.class);
        startActivity(intent);
        return true;
    }
    return false;
}
});

```

```

contacts = (Button)findViewById(R.id.contacts);
final MediaPlayer mp1= MediaPlayer.create(this, R.raw.contacts);
contacts.setOnClickListener(new View.OnClickListener(){

```

```

@Override
public boolean onTouch(View v, MotionEvent event) {
    if(event.getAction() == MotionEvent.ACTION_DOWN){
        mp1.start();
        Intent intent = new Intent(MainMenu.this, Contact.class);
        startActivity(intent);
        return true;
    }
    return false;
}
});

```

```

message = (Button)findViewById(R.id.message);
final MediaPlayer mp2= MediaPlayer.create(this, R.raw.message);
message.setOnClickListener(new View.OnClickListener(){

```

```

@Override
public boolean onTouch(View v, MotionEvent event) {
    if(event.getAction() == MotionEvent.ACTION_DOWN){
        mp2.start();
        Intent intent = new Intent(MainMenu.this, Message.class);
        startActivity(intent);
        return true;
    }
    return false;
}
});

```

```

email = (Button)findViewById(R.id.email);
final MediaPlayer mp3= MediaPlayer.create(this, R.raw.email);
email.setOnClickListener(new View.OnClickListener(){

```

```

@Override
public boolean onTouch(View v, MotionEvent event) {
    if(event.getAction() == MotionEvent.ACTION_DOWN){
        mp3.start();
        Intent intent = new Intent(MainMenu.this, Email.class);
        startActivity(intent);
        return true;
    }
    return false;
}
});

```

```

camera = (Button)findViewById(R.id.camera);
final MediaPlayer mp4= MediaPlayer.create(this, R.raw.camera);
camera.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public boolean onTouch(View v, MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {

```

```

        mp4.start();
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivity(intent);
        return true;
    }
    return false;
}
});

gallery = (Button)findViewById(R.id.gallery);
final MediaPlayer mp5= MediaPlayer.create(this, R.raw.gallery);
gallery.setOnClickListener(new View.OnClickListener() {

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            mp5.start();
            Intent intent = new Intent(Intent.ACTION_VIEW,
            Uri.parse("content://media/internal/images/media"));
            startActivity(intent);
            return true;
        }
        return false;
    }
});

emergency = (Button) findViewById(R.id.emergency);
final MediaPlayer mp6= MediaPlayer.create(this, R.raw.emergency);
emergency.setOnClickListener(new View.OnClickListener() {

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            mp6.start();
            Intent intent = new Intent(MainMenu.this,Emergency.class);
            startActivity(intent);
            return true;
        }
        return false;
    }
});

flashlight = (Button) findViewById(R.id.flash);
flashlight.setOnClickListener(new View.OnClickListener() {

    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            Intent intent = new Intent(MainMenu.this,Flash.class);
            startActivity(intent);
            return true;
        }
        return false;
    }
});

}
@Override
public void onBackPressed() {
    Log.d("CDA", "onBackPressed Called");
    Intent setIntent = new Intent(MainMenu.this,StarterActivity.class);
    setIntent.addCategory(Intent.CATEGORY_HOME);
    setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(setIntent);
}
}

```

```
}
.
}
```

Message.java

```
package ekoudcom.harris.helptheoldones;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

/**
 * Created by Harris on 6/29/2015.
 */
public class Message extends Activity {
    Button send;
    EditText recipient,message;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.message_layout);

        recipient=(EditText)findViewById(R.id.recipientNumber);
        message=(EditText)findViewById(R.id.messageText);
        send= (Button)findViewById(R.id.messageSendButton);
        send.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            String url= recipient.getText().toString();
            String text=message.getText().toString();
            Intent smsIntent = new Intent(Intent.ACTION_VIEW);
            smsIntent.setType("vnd.android-dir/mms-sms");
            smsIntent.putExtra("address", url);
            smsIntent.putExtra("sms_body",text);
            //sendIntent.putExtra(Intent.EXTRA_TEXT,text);
            //sendIntent.putExtra("address", phone);
            startActivity(smsIntent);

        }
        });
    }
}
```

Phone.java

```
package ekoudcom.harris.helptheoldones;

import android.annotation.TargetApi;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.NavUtils;
import android.view.Menu;
import android.view.MenuItem;
```

```

import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class Phone extends Activity {

    TextView textViewCallNumber;
    Button one,two,three,four,five,six,seven,eight,nine,star,zero,hash,backspace,call;
    String number = " ";

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.phone_layout);
        textViewCallNumber= (TextView)findViewById(R.id.textViewCallNumber);

        zero= (Button)findViewById(R.id.button0);
        zero.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                number=number+"0";
                textViewCallNumber.setText(number);
            }
        });
        one= (Button)findViewById(R.id.button1);
        one.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                number=number+"1";
                textViewCallNumber.setText(number);
            }
        });
        two= (Button)findViewById(R.id.button2);
        two.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                number=number+"2";
                textViewCallNumber.setText(number);
            }
        });
        three= (Button)findViewById(R.id.button3);
        three.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                number=number+"3";
                textViewCallNumber.setText(number);
            }
        });
        four= (Button)findViewById(R.id.button4);
        four.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                number=number+"4";
                textViewCallNumber.setText(number);
            }
        });
        five= (Button)findViewById(R.id.button5);
        five.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                // TODO Auto-generated method stub
                number = number + "5";
                textViewCallNumber.setText(number);
            }
        });
        six= (Button)findViewById(R.id.button6);

```

```

six.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=number+"6";
    textViewCallNumber.setText(number);
}
});
seven= (Button)findViewById(R.id.button7);
seven.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=number+"7";
    textViewCallNumber.setText(number);
}
});
eight= (Button)findViewById(R.id.button8);
eight.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=number+"8";
    textViewCallNumber.setText(number);
}
});
nine= (Button)findViewById(R.id.button9);
nine.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=number+"9";
    textViewCallNumber.setText(number);
}
});
hash= (Button)findViewById(R.id.buttonSem);
hash.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=number+"#";
    textViewCallNumber.setText(number);
}
});
star= (Button)findViewById(R.id.buttonStar);
star.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=number+"*";
    textViewCallNumber.setText(number);
}
});
backspace= (Button)findViewById(R.id.buttonDelete);
backspace.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub
    number=(String) number.subSequence(0,number.length()-1);
    textViewCallNumber.setText(number);
}
});

call= (Button)findViewById(R.id.buttonCall);
call.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    String url= textViewCallNumber.getText().toString();
    Intent intent = new Intent(Intent.ACTION_CALL);
    intent.setData(Uri.parse("tel:" +number));
    startActivity(intent);

if(number.length()>10){
    textViewCallNumber.setTextSize(50f);

```

```

    }

    }
});

}

}

```

StarterActivity.java

```

package ekoudcom.harris.helptheoldones;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class StarterActivity extends Activity {
    EditText number;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        SharedPreferences loadSettings = getSharedPreferences("settings", MODE_PRIVATE);
        boolean isContact = loadSettings.getBoolean("Contact", false);
        if (isContact) {
            Intent intent = new Intent(StarterActivity.this, MainMenu.class);
            startActivity(intent);
        } else {
            setContentView(R.layout.starteractivity);
            number = (EditText) findViewById(R.id.emerContact);
        }
    }

    public void save(View v) {

        /*{
        int temp=0;
        for (int i=0;i<number.getText().toString().length(); i++){
            if(number.getText().toString().charAt(i)="0" || number.getText().toString().charAt(i)="9" ){
                temp++;
            }
        }
        */

        if (number.getText().toString().contains("[a-zA-Z]+") == false && number.getText().toString().length() >
2) {
            {
                SharedPreferences shared = getSharedPreferences("myData", MODE_PRIVATE);
                SharedPreferences.Editor editor = shared.edit();
                editor.putString("Number", number.getText().toString());
                editor.commit();
                SharedPreferences settings = getSharedPreferences("settings", MODE_PRIVATE);
                SharedPreferences.Editor editor1 = settings.edit();
                editor1.putBoolean("Contact", true);
            }
        }
    }
}

```

```
editor1.commit();
Toast.makeText(this, "Saved Successfully", Toast.LENGTH_SHORT).show();
Intent intent = new Intent(StarterActivity.this, MainMenu.class);
startActivity(intent);
}

} else {
    Toast.makeText(this, "You Haven't entered a Number", Toast.LENGTH_SHORT).show();
}

}

public void skip(View v) {
    Intent intent = new Intent(StarterActivity.this, MainMenu.class);
    startActivity(intent);
}

public void onBackPressed() {
    Log.d("CDA", "onBackPressed Called");
    Intent setIntent = new Intent(StarterActivity.this, StarterActivity.class);
    setIntent.addCategory(Intent.CATEGORY_HOME);
    setIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
    startActivity(setIntent);
}

}
```

Παράρτημα Β

Κώδικας εφαρμογής φιλοξενούμενης στο σύννεφο

Index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>
<title>ARaynorDesign Template</title>
<meta name="description" content="free website template" />
<meta name="keywords" content="enter your keywords here" />
```

```

<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/jquery.easing.min.js"></script>
<script type="text/javascript" src="js/jquery.nivo.slider.pack.js"></script>
<script type="text/javascript">
$(window).load(function() {
  $('#slider').nivoSlider();
});
</script>
</head>

<body>
<div id="main">

  <div id="menubar">
    <div id="welcome">
      <h1><a href="#">Help The Old Ones</a></h1>
    </div><!--close welcome-->
  <div id="menu_items">
    <ul id="menu">
      <li class="current"><a href="index.html">Αρχική</a></li>
      <li><a href="%="ourwork1.jsp"%>Εντοπισμός</a></li>
      <li><a href="%="contact1.jsp"%>Επικοινωνία</a></li>
    </ul>
  </div><!--close menu-->
</div><!--close menubar-->

  <div id="site_content">

    <div id="banner_image">
      <div id="slider-wrapper">
        <div id="slider" class="nivoSlider">
          
          
        </div><!--close slider-->
      </div><!--close slider_wrapper-->
    </div><!--close banner_image-->

    <div class="sidebar_container">
      <div class="sidebar">
        <div class="sidebar_item">
          <h2>Νέα Ιστοσελίδα</h2>
          <p>Είναι η νέα μας ιστοσελίδα. Οποιαδήποτε απόψη ή σχόλιο θα εκτιμηθεί προκειμένου να την κάνουμε ακόμη πιο προσιτή και εύκολη στη χρήση.</p>
        </div><!--close sidebar_item-->
      </div><!--close sidebar-->

      <div class="sidebar">
        <div class="sidebar_item">
          <h2>Τελευταία Ενημέρωση</h2>
          <h3>Ιούνιος 2017</h3>
          <p>Παρουσίαση διπλωματικής διατριβής.</p>

```



```

        </div><!--close sidebar_item-->
</div><!--close sidebar-->
        <div class="sidebar">
</div><!--close sidebar-->
</div><!--close sidebar_container-->

        <div id="content">
<div class="content_item">
        <h1>Καλώ ήρθατε στο Το Help The Old Ones</h1>
        <p>Με αυτή την εφαρμογή θελήσαμε να δώσουμε τη δυνατότητα σε άτομα που δεν μπορούν να βρίσκονται μαζί με τα αγαπημένα τους άτομα, αλλά επιθυμούν να γνωρίζουν σε ποια κατάσταση βρίσκονται.
        <p>.Πως επιτυγχάνεται αυτό; Μα φυσικά με την εφαρμογή για Android συσκευές, Help The Old Ones. Πλέον μπορείται να γνωρίζεται την ακριβή τοποθεσία του αγαπημένου σας προσώπου, με ένα μόνο κλικ</p>

        </div><!--close content_item-->
</div><!--close content-->

</div><!--close main-->

<div id="footer">
        <a href="http://validator.w3.org/check?uri=referer">Valid XHTML</a> | <a href="http://fotogrph.com/">Images</a> | <a href="http://www.heartinternet.co.uk/web-hosting/">Web Hosting</a> | website template by <a href="http://www.araynordesign.co.uk">ARaynorDesign</a>
</div><!--close footer-->

</body>
</html>

```

Ourwork.jsp

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="com.koud.webservice.SENSORSUtil"%>
<%@page import="com.koud.webservice.Obj.s.SensorData"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
<style>
#map {
height: 400px;
width: 100%;
}
</style>
</head>
<title>Help The Old Ones</title>
<meta name="description" content="free website template" />
<meta name="keywords" content="enter your keywords here" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script type="text/javascript" src="js/jquery.min.js"></script>

```

```

<script type="text/javascript" src="js/jquery.easing.min.js"></script>
<script type="text/javascript" src="js/jquery.nivo.slider.pack.js"></script>
<script type="text/javascript">
$(window).load(function () {
    $('#slider').nivoSlider();
});

function reload(){
    location.reload();
}
</script>

<meta http-equiv="refresh" content="30" />
</head>

<body>
<div id="main">

<div id="menubar">
<div id="welcome">
<h1><a href="#">Help The Old Ones</a></h1>
</div><!--close welcome-->
<div id="menu_items">
<ul id="menu">
<li><a href="index1.jsp">Αρχική</a></li>
<li class="current"><a href="ourwork1.jsp">Εντοπισμός</a></li>
<li><a href="contact1.jsp">Επικοινωνία</a></li>
</ul>
</div><!--close menu-->
</div><!--close menubar-->
<%
SensorData sData = SENSORSUtil.loadLatestData();
%>
<input type="hidden" id="lat" value="<%=sData.getLatParam()%>"/>
<input type="hidden" id="longt" value="<%=sData.getLongtParam()%>"/>

<div id="site_content">
<div>
<h3>Google Maps</h3>
<div id="map"></div>
<script>
function initMap() {
    var lat = $("#lat").val();
    var longt = $("#longt").val();
    //alert(lat);
    lat = parseFloat(lat);
    //alert(longt);
    longt = parseFloat(longt);
    //var uluru = {lat:38.061, lng:23.845 };
    var uluru = {lat: longt, lng: lat};
    var map = new google.maps.Map(document.getElementById('map'), {

```

```

        zoom: 4,
        center: uluru
    });
    var marker = new google.maps.Marker({
        position: uluru,
        map: map
    });
}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=AlzaSyDgr1fplggpBEG505PInTj4mKUU97tdwgg&callb
ack=initMap">
</script>

</div>
<div class="sidebar_container">
<div class="sidebar">
<div class="sidebar_item">
<h2>Νέα Ιστοσελίδα</h2>
<p>Αποτελεί μέρος της διπλωματικής διατριβής. Σκοπό έχει η ιστοσελίδα να φιλοξενηθεί σε
εξυπηρετητή στο σύννεφο.
</p>
</div><!--close sidebar_item-->
</div><!--close sidebar-->
<div class="sidebar">
<div class="sidebar_item">
<h2>Τελευταία Ενημέρωση</h2>
<h3>June 2017</h3>
<p>Παρουσίαση διπλωματικής διατριβής.</p>
</div><!--close sidebar_item-->
</div><!--close sidebar-->

</div><!--close sidebar_container-->

<button style="margin-top: 5px; width: 30%; height: 40px; background: #B29E74; border-radius: 15px
15px 15px 15px;" onclick="reload();">Εκ Νέου Εντοπισμός</button>

<div id="content">
<div class="content_item">
<h2>Our Work</h2>
<p>Μπορείτε μέσα από αυτή τη σελίδα να ενημερωθείτε για την ακριβή τοποθεσία του αγαπημένου
σας ατόμου. Η σελίδα έχει οριστεί να
ενημερώνεται με την τελευταία θέση, που έχει αποσταλλεί από την συσκευή Android, στην οποία είναι
εγκατεστημένη η Help The Old Ones.</p>

</div><!--close content_item-->
</div><!--close content-->
</div><!--close site_content-->

</div><!--close main-->

<div id="footer">

```

```

    <a href="http://validator.w3.org/check?uri=referer">Valid XHTML</a> | <a
href="http://fotogrph.com/">Images</a> | <a href="http://www.heartinternet.co.uk/web-hosting/">Web
Hosting</a> | website template by <a href="http://www.araynordesign.co.uk">ARaynorDesign</a>
  </div><!--close footer-->

</body>
</html>

```

Contact1.jsp

```

<%--
Document : contact1
Created on : Mar 12, 2017, 11:21:45 PM
Author : harris
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>
<title>Help The Old Ones</title>
<meta name="description" content="free website template" />
<meta name="keywords" content="enter your keywords here" />
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
<script type="text/javascript" src="js/jquery.min.js"></script>
<script type="text/javascript" src="js/jquery.easing.min.js"></script>
<script type="text/javascript" src="js/jquery.nivo.slider.pack.js"></script>
<script type="text/javascript">
$(window).load(function() {
    $('#slider').nivoSlider();
});
</script>
</head>

<body>
<div id="main">

    <div id="menubar">
        <div id="welcome">
            <h1><a href="#">Help The Old Ones</a></h1>
        </div><!--close welcome-->
    <div id="menu_items">
        <ul id="menu">
            <li><a href="index1.jsp">Αρχική</a></li>
            <li><a href="ourwork1.jsp">Εντοπισμός</a></li>
            <li class="current"><a href="contact1.jsp">Επικοινωνία</a></li>
        </ul>
    </div>
</div><!--close menu-->
</div><!--close menubar-->

```

```

<div id="site_content">

  <div id="banner_image">
    <div id="slider-wrapper">
      <div id="slider" class="nivoSlider">
        
        
      </div><!--close slider-->
    </div><!--close slider_wrapper-->
  </div><!--close banner_image-->

  <div class="sidebar_container">
    <div class="sidebar">
      <div class="sidebar_item">
        <h2>Help The Old Ones</h2>
        <p>Είναι η νέα μας ιστοσελίδα. Οποιαδήποτε απόψη ή σχόλιο θα εκτιμηθεί προκειμένου να την κάνουμε
        ακόμη πιο προσιτή και εύκολη στη χρήση.</p>
      </div><!--close sidebar_item-->
    </div><!--close sidebar-->
    <div class="sidebar">
      <div class="sidebar_item">
        <h2>Τελευταία Ενημέρωση</h2>
        <h3>Ιούνιος 2017</h3>
        <p>Παρουσίαση διπλωματικής διατριβής.</p>
      </div><!--close sidebar_item-->
    </div><!--close sidebar-->
    <div class="sidebar">

</div><!--close sidebar-->
</div><!--close sidebar_container-->

<div id="content">
<div class="content_item">
  <div class="form_settings">
    <h2>Contact Us</h2>
    <p>Όνομάζομαι Κουδουμάς Χαρίδημος και είμαι απόφοιτος του Πανεπιστημίου ΟΠΑ. Αυτή τη στιγμή
    είμαι στη φάση της διπλωματικής διατριβής, μέρος της οποία αποτελεί η παρούσα
    ιστοσελίδα. Επιπρόσθετα εργάζομαι ως Software Engineer στην πολυεθνική εταιρεία Accenture.</p>
    <p>&nbsp;</p>
    <p><span>Name</span><input class="contact" type="text" name="your_name"
    value="" /></p>
    <p><span>Email Address</span><input class="contact" type="text"
    name="your_email" value="" /></p>
    <p><span>Message</span><textarea class="contact textarea" rows="8"
    cols="50" name="your_message"></textarea></p>
    <p style="padding: 10px 0 10px 0;">Please enter the answer to this simple maths question (to prevent
    spam)</p>
    <p><span>Maths Question: 9 + 3 = ?</span><input type="text"
    name="user_answer" class="contact" /><input type="hidden" name="answer" value="4d76fe9775" /></p>
    <p style="padding-top: 15px"><span>&nbsp;</span><input class="submit" type="submit"
    name="contact_submitted" value="Send" /></p>
  </div><!--close form_settings-->
</div><!--close content_item-->

```

```
</div><!--close content-->
```

```
</div><!--close site_content-->
```

```
</div><!--close main-->
```

```
<div id="footer">
```

```
    <a href="http://validator.w3.org/check?uri=referer">Valid XHTML</a> | <a
href="http://fotogrph.com/">Images</a> | <a href="http://www.heartinternet.co.uk/web-hosting/">Web
Hosting</a> | website template by <a href="http://www.araynordesign.co.uk">ARaynorDesign</a>
```

```
</div><!--close footer-->
```

```
</body>
```

```
</html>
```

Hello.java

```
package com.koud.webservice;
```

```
import com.koud.webservice.Objcs.SensorData;
```

```
import javax.jws.WebService;
```

```
import javax.jws.WebMethod;
```

```
import javax.jws.WebParam;
```

```
/**
```

```
*
```

```
* @author harris
```

```
*/
```

```
@WebService(serviceName = "Hello")
```

```
public class Hello {
```

```
/**
```

```
* This is a sample web service operation
```

```
*/
```

```
@WebMethod(operationName = "hello")
```

```
public String hello(@WebParam(name = "name") String txt) {
```

```
    return "Hello " + txt + " !";
```

```
}
```

```
@WebMethod(operationName = "sensorMeasurements")
```

```
public String sensorMeasurements(
```

```
    @WebParam(name = "valueX") String valueX,
```

```
    @WebParam(name = "valueY")String valueY,
```

```
    @WebParam(name = "lat") String lat,
```

```
    @WebParam(name = "longt") String longt ) {
```

```
try{
```

```
    SENSORSUtil.storMeasurements(valueX,valueY,lat,longt);
```

```
    SensorData sdata = SENSORSUtil.loadData();
```

```

    return "SUCCESS" ;
}
catch(Exception e){

    return "FAIL";
}

}
}

```

StoredProc.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.koud.webservice;

import java.sql.*;
import java.util.Scanner;
import oracle.jdbc.*;

public class StoredProc {

    public static void test(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc=new Scanner(System.in);
        Connection cn=null;
        CallableStatement cst=null;
        int id=0,yr;
        String fname,lname,email,dept;
        try{
            DriverManager.registerDriver(new OracleDriver());
            cn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"SENSORS", "7899");

            cst=cn.prepareCall("{call insertStudent(?,?,?,?,?)}");
            System.out.println("Enter Student's First Name: ");
            fname=sc.nextLine();
            System.out.println("Enter Student's Last Name: ");
            lname=sc.nextLine();
            System.out.println("Enter Student's Email: ");
            email=sc.nextLine();
            System.out.println("Enter Student's Year: ");
            yr=Integer.parseInt(sc.nextLine());
            System.out.println("Enter Student's Dept: ");
            dept=sc.nextLine();

```

```

        cst.setString(1, fname);
        cst.setString(2, lname);
        cst.setString(3, dept);
        cst.setInt(4, yr);
        cst.setString(5, email);

        cst.registerOutParameter(6, java.sql.Types.DECIMAL);

        int r=cst.executeUpdate();
        if(r>0)
            id=cst.getInt(6);

        if(id>0)
            System.out.println("Details Inserted. SID: "+id);

    }catch(Exception e)
    {
        e.printStackTrace();
    }
    finally{
        try{
            if(cst!=null)
                cst.close();

            if(cn!=null)
                cn.close();
        }catch(Exception e){}
    }
}
}
}

```

SensorsUtil.java

```

package com.koud.webservice;

import com.koud.webservice.Objcs.SensorData;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Calendar;
import java.util.List;
import java.util.Scanner;
import oracle.jdbc.OracleDriver;
import oracle.jdbc.*;

```



```

public class SENSORSUtil{

    public static void storMeasurements(String xParam, String yParam, String latParam, String
longtParam )throws SQLException {
        Connection cn=null;
        CallableStatement cst=null;
        int id=0,yr;
        try{
            DriverManager.registerDriver(new OracleDriver());
            cn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"SENSORS", "7899");
            cst=cn.prepareCall("{call LOGDATA(?,?,?,?)");
            cst.setString(1,xParam );
            cst.setString(2,yParam );
            cst.setString(3,latParam );
            cst.setString(4,longtParam );
            cst.registerOutParameter(5, java.sql.Types.VARCHAR);
            //cst.registerOutParameter(6, java.sql.Types.DECIMAL);
            int r=cst.executeUpdate();
            if(r>0)
                id=cst.getInt(3);

            if(id>0)
                System.out.println("The Values have been saved" + id );

        }catch(Exception e)
        {
            e.printStackTrace();
        }
        finally{
            try{
                if(cst!=null)
                    cst.close();

                if(cn!=null)
                    cn.close();
            }catch(Exception e){}
        }
    }

// public static List<SensorData> loadData(){
public static SensorData loadData(){

    Statement stmt = null;
    Connection cn=null;
    String query = "select XPARAM,YPARAM,LATPARAM,LONGTPARAM " +
"from measurements where id = 39999";
    SensorData sData = new SensorData ();
    try {
        DriverManager.registerDriver(new OracleDriver());
            cn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"SENSORS", "koudoumas90");
            stmt = cn.createStatement();

```

```

ResultSet rs = stmt.executeQuery(query);

while (rs.next()) {
    String xParam = rs.getString("XPARAM");
    String yParam = rs.getString("YPARAM");
    String latParam = rs.getString("LATPARAM");
    String longtParam = rs.getString("LONGTPARAM");

    sData.setXParam(xParam);
    sData.setYParam(yParam);
    sData.setLatParam(latParam);
    sData.setLongtParam(longtParam);
    System.out.println(xParam + "\t" + yParam +
        "\t" + latParam + "\t" + longtParam +
        "\t");
}
}

catch(Exception e){
}
finally{
    try{
        if(stmt!=null)
            stmt.close();

        if(cn!=null)
            cn.close();
    }catch(Exception e){}
}

return sData;
}

public static SensorData loadLatestData(){

    Statement stmt = null;
    Connection cn=null;
    String query = "select XPARAM,YPARAM,LATPARAM,LONGTPARAM " +
        "from sensors.measurements where log_date = ( select max(log_date) from sensors.measurements) ";
    SensorData sData = new SensorData ();
    try {
        DriverManager.registerDriver(new OracleDriver());
        cn=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe",
"SYSTEM", "7899");
        stmt = cn.createStatement();
        ResultSet rs = stmt.executeQuery(query);

        while (rs.next()) {
            String xParam = rs.getString("XPARAM");
            String yParam = rs.getString("YPARAM");
            String latParam = rs.getString("LATPARAM");
            String longtParam = rs.getString("LONGTPARAM");

```

```

        sData.setXParam(xParam);
        sData.setYParam(yParam);
        sData.setLatParam(latParam);
        sData.setLongtParam(longtParam);
        System.out.println(xParam + "\t" + yParam + "\t" + latParam + "\t" + longtParam + "\t");
    }
}
catch(Exception e){
    e.printStackTrace();
    System.out.println("Exception");
}
finally{
    try{
        if(stmt!=null)    stmt.close();
        if(cn!=null)    cn.close();
    }catch(Exception e){}
}
return sData;
}
}

```

Main.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.koud.webservice;

import javax.xml.ws.Endpoint;
import com.koud.webservice.Hello;
/**
 *
 * @author harris
 */
public class main {

    public static void main (String Args[]){
        Endpoint.publish("http://"+"192.168.43.151"+":"+ "9090"+"/KoudService/",new Hello());
        System.out.print("ENDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD");
    }

}

```

SensorData.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.koud.webservice.Objjs;

```

```
public class SensorData {

    private String xParam;
    private String yParam;
    private String latParam;
    private String longtParam;

    public SensorData(String xParam,String yParam,String latParam,String longtParam){
        this.xParam= xParam;
        this.yParam = yParam;
        this.latParam = latParam;
        this.longtParam = longtParam;
    }
    public SensorData(){
    }

    public String getXParam() {
        return xParam;
    }

    public void setxParam(String xParam) {
        this.xParam = xParam;
    }

    public String getYParam() {
        return yParam;
    }

    public void setyParam(String yParam) {
        this.yParam = yParam;
    }

    public String getLatParam() {
        return latParam;
    }

    public void setLatParam(String latParam) {
        this.latParam = latParam;
    }

    public String getLongtParam() {
        return longtParam;
    }

    public void setLongtParam(String longtParam) {
        this.longtParam = longtParam;
    }
}
```

Παράρτημα Γ

Scripts για τη βάση σε ORACLE CLIENT

LOGDATA (PROCEDURE)

```
CREATE OR REPLACE PROCEDURE LOGDATA (  
XPARAM IN VARCHAR2,  
YPARAM IN VARCHAR2,  
LATXPARAM IN VARCHAR2,  
LONGTPARAM IN VARCHAR2,  
OUTCOME OUT VARCHAR2)  
AS  
  
SEQ_VAL NUMBER ;  
  
BEGIN  
  
SELECT MEAS_SEQ.NEXTVAL INTO SEQ_VAL FROM DUAL ;  
  
INSERT INTO MEASUREMENTS VALUES (  
SEQ_VAL,  
SYSDATE,  
XPARAM,  
YPARAM,  
LATXPARAM,  
LONGTPARAM  
);  
OUTCOME := 'SUCCESS';  
EXCEPTION WHEN OTHERS THEN  
  
OUTCOME := 'FAIL';  
END;
```

MEAS_SEQ

```
CREATE SEQUENCE MEAS_SEQ  
START WITH 1  
INCREMENT BY 1  
NOCACHE  
NOCYCLE;
```

MEASUREMENTS

```
CREATE TABLE MEASUREMENTS  
(  
  ID NUMBER PRIMARY KEY,  
  LOG_DTM DATE,  
  XPARAM DOUBLE,  
  YPARAM DOUBLE,  
  LATPARAM DOUBLE,  
  LONGTPARAM DOUBLE  
);
```

