



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

## ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σχεδιασμός Πληροφοριακού Συστήματος  
Καταγραφής μετρήσεων κοινής ωφελείας  
Υποβοηθούμενο από οπτική αναγνώριση μέσω  
Κινητού τηλεφώνου

Design of an Information System for recording utility  
measurements assisted by optical recognition (OCR) via  
mobile phone

**Μπούντας Δημήτρης**

**Επιβλέπων Καθηγητής : Δασυγένης Μηνάς**

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

Κοζάνη, Ιούνιος 2016



## Περίληψη

Η εξέλιξη της τεχνολογίας μαζί με τον συνδυασμό των τεχνολογικών επιτευγμάτων τα τελευταία χρόνια, μας βοηθούν όλο και περισσότερο στην εύκολη και όσο πιο δυνατόν, άμεση εξυπηρέτηση των αναγκών μας.

Στην παρούσα διπλωματική εργασία, αναπτύχθηκε ένα σύστημα το οποίο στοχεύει ακριβώς στην εξυπηρέτηση των αναγκών αυτών. Αυτό έγινε εφικτό με την υλοποίηση ενός συστήματος που συμπεριλαμβάνει μία εφαρμογή σε λειτουργικό σύστημα Android και αντίστοιχη διαδικτυακή εφαρμογή. Τα στοιχεία αλλά και τα δεδομένα και των δύο, υποστηρίζονται και καταχωρούνται σε μία βάση δεδομένων. Για την ανάπτυξη των παραπάνω αναζητήθηκαν τεχνικές σχετικά με :

- Την εύκολη και άμεση επικοινωνία της εφαρμογής Android με την αντίστοιχη διαδικτυακή εφαρμογή.
- Τη δημιουργία μίας βάσης δεδομένων που θα προσφέρει ασφάλεια στα δεδομένα των χρηστών της.
- Την, όσο το δυνατό, πιο εύστοχη αναγνώριση εικόνας από την εφαρμογή Android.
- Τη δυνατότητα αποστολής και αποθήκευσης των δεδομένων που αναγνωρίστηκαν άμεσα, στον προσωπικό λογαριασμό του χρήστη στη βάση δεδομένων.
- Τη δυνατότητα επεξεργασίας των δεδομένων αυτών.

Αποτέλεσμα της διπλωματικής εργασίας αποτέλεσε ένα πλήρες σύστημα αποτελούμενο από μία διαδικτυακή εφαρμογή και μία εφαρμογή Android, οι οποίες συνεργάζονται μεταξύ τους προσφέροντας πολλές λειτουργίες σχετικά με την καταχώρηση και επεξεργασία δεδομένων μέσω οπτικής αναγνώρισης.

**Λέξεις κλειδιά :** OCR, Android, Tesseract, tess-two, NDK, Java, Android Studio, Προγραμματισμός διαδικτύου PHP, HTML, MySQL, απλός χρήστης, τεχνικός, διαχειριστής.

## Abstract

The evolution of technology and the resulting technological advances that took place in recent years, contributed in our ability to fulfil our needs more directly and through easier ways.

The aim of the following thesis was to develop a system which would serve precisely those needs. An aim that became possible through the creation of a system that contained both an Android application and a corresponding web-based application. The information and the data of the aforementioned applications are both supported and electronically recorded at a data base. In order to implement the above, research was necessary on certain methods concerning:

- The easy and direct communication between the Android application and the corresponding web-based application.
- The creation of a data base with the capability to secure the data provided by its users.
- The capability of the Android application to astutely - as much as possible - recognize an image.
- The potential to send and save the recently recognized on the personal account of the user on the database.
- The possibility for the user to be able to edit their data.

The result of this thesis is a complete system that consists of a web-based application and an Android application which - through cooperation with one another - offer numerous functions that have to do with the input and the processing of the data acquired through optical recognition.

**Keywords:** OCR, Android, Tesseract, tess-two, NDK, Java, Android Studio, PHP Web Programming, HTML, MySQL, normal user, technician, administrator.



## Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν.1256/1982, η παρούσα Διπλωματική εργασία με τίτλο «Σχεδιασμός Πληροφοριακού Συστήματος Καταγραφής μετρήσεων κοινής ωφελείας Υποβοηθούμενο από οπτική αναγνώριση μέσω Κινητού τηλεφώνου» καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Μπούντας Δημήτριος & Μηνάς Δασυγένης, 2016, Κοζάνη

Υπογραφή Φοιτητή

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Μηνά Δασυγένη για τη δυνατότητα που μου έδωσε να ασχοληθώ με ένα πρωτότυπο και ενδιαφέρον θέμα όπως επίσης και την βοήθεια του κατά τη συγγραφή.

Επίσης θέλω να ευχαριστήσω την οικογένεια μου που ήταν δίπλα μου κάθε στιγμή, σε όλη τη διάρκεια πραγματοποίησης αυτής της διπλωματικής, καθώς και τους φίλους μου και την κοπέλα μου για την στήριξη και την ώθηση που μου έδωσαν.

# Περιεχόμενα

Ευχαριστίες .....	6
Κατάλογος εικόνων.....	10
Κατάλογος πινάκων .....	12
Κεφάλαιο 1 .....	14
Εισαγωγή.....	14
Σύνοψη κεφαλαίου .....	18
Κεφάλαιο 2 .....	19
Θεωρητικό υπόβαθρο.....	19
Προγραμματισμός διαδικτύου.....	19
2.1 HTML.....	19
2.2 Μορφοποίηση CSS .....	22
2.3 PHP.....	24
2.4 MySQL.....	25
2.5 JavaScript .....	26
2.6 AJAX.....	27
2.7 Android Studio .....	28
2.7.1 Εγκατάσταση NDK και Βιβλιοθήκης tess-two .....	33
2.7.2 Ανάπτυξη βιβλιοθήκης tess-two στο Android Studio .....	38
2.8 Notepad ++.....	40
Σύνοψη κεφαλαίου .....	40
Κεφάλαιο 3 .....	42
Περιγραφή και απαιτήσεις συστήματος.....	42
3.1 Περιγραφή συστήματος.....	42
3.2 Αρχιτεκτονική Συστήματος.....	44
3.3 Απαιτήσεις συστήματος .....	45
Σύνοψη κεφαλαίου .....	47
Κεφάλαιο 4 .....	48
Υλοποίηση λογισμικού μέρους συστήματος.....	48
4.1 Σχεδιασμός βάσης δεδομένων.....	48
4.1.1 Μοντέλο Οντοτήτων-Συσχετίσεων .....	50
4.1.1 Πίνακας users .....	51
4.1.2 Πίνακας User_Data.....	53
4.2 Ανάπτυξη εφαρμογής στο Android Studio .....	54
4.2.1 Λειτουργίες login και register εφαρμογής.....	56



4.2.2 Λειτουργίες και δυνατότητες της εφαρμογής.....	63
4.3 Ανάπτυξη ιστοχώρου .....	76
4.3.1 Αρχική σελίδα.....	76
4.3.2 Λειτουργίες Διαχειριστή (υπερχρήστη).....	77
4.3.3 Λειτουργίες χρηστών κατώτερων επιπέδων .....	82
4.4 Στοιχεία σύνδεσης διακομιστή-εφαρμογής-ιστοσελίδας .....	87
4.5 Στοιχεία κώδικα .....	88
4.6 Ασφάλεια συστήματος .....	88
4.7 Δοκιμές.....	92
Σύνοψη κεφαλαίου .....	94
Κεφάλαιο 5 .....	96
Συμπεράσματα .....	96
Μελλοντικές επεκτάσεις .....	99
Οδηγίες εγκατάστασης.....	101
Εγκατάσταση αρχείων σε διακομιστή.....	101
Εγκατάσταση εφαρμογής Android.....	102



## Κατάλογος εικόνων

<b>Εικόνα 1:</b> Το λογότυπο της HTML.....	20
<b>Εικόνα 2:</b> Κανόνας σύνταξης CSS.....	23
<b>Εικόνα 3:</b> Παράδειγμα κώδικα php.....	24
<b>Εικόνα 4:</b> Λογότυπο της MySQL .....	25
<b>Εικόνα 5:</b> Παράδειγμα κώδικα JavaScript μέσα σε κώδικα HTML.....	27
<b>Εικόνα 6:</b> Παράδειγμα διαδικτυακής εφαρμογής με και χωρίς AJAX.....	28
<b>Εικόνα 7:</b> Το λογότυπο του Android Studio .....	29
<b>Εικόνα 8:</b> Δομή του Android Project .....	30
<b>Εικόνα 9:</b> Επιλογή έκδοσης κατά τη δημιουργία του project.....	34
<b>Εικόνα 10:</b> Εντοπισμός μεταβλητών συστήματος .....	35
<b>Εικόνα 11:</b> Εισαγωγή του NDK στο Path του συστήματος.....	36
<b>Εικόνα 12:</b> Εισαγωγή μονοπατιού του NDK .....	36
<b>Εικόνα 13:</b> Άνοιγμα βιβλιοθήκης στη γραμμή εντολών .....	37
<b>Εικόνα 14:</b> Τροποποίηση βιβλιοθήκης μέσω NDK .....	37
<b>Εικόνα 15:</b> Το αρχείο build.gradle του tess-two .....	39
<b>Εικόνα 16:</b> Δήλωση βιβλιοθήκης του project .....	39
<b>Εικόνα 17:</b> Καταχώρηση βιβλιοθήκης στο project.....	39
<b>Εικόνα 18 :</b> Το Notepad++.....	40
<b>Εικόνα 19:</b> Αρχιτεκτονική συστήματος .....	45
<b>Εικόνα 20:</b> Διάγραμμα οντοτήτων-συσχετίσεων της βάσης δεδομένων .....	50
<b>Εικόνα 21:</b> Οι πίνακες της βάσης που αναπτύχθηκε .....	51
<b>Εικόνα 22:</b> Το AndroidManifest.xml της εφαρμογής.....	55
<b>Εικόνα 23:</b> Το gradle του project.....	56
<b>Εικόνα 24:</b> Επικοινωνία Android και βάσης δεδομένων μέσω JSON/PHP .....	57
<b>Εικόνα 25:</b> Αποστολή δεδομένων για είσοδο του χρήστη και δημιουργία session....	59
<b>Εικόνα 26:</b> Το login screen της εφαρμογής .....	60
<b>Εικόνα 27:</b> Δημιουργία λογαριασμού χρήστη (RegisterActivity) .....	61
<b>Εικόνα 28:</b> Το register της εφαρμογής.....	62
<b>Εικόνα 29:</b> Το MainActivity της εφαρμογής .....	64
<b>Εικόνα 30:</b> Η κύρια οθόνη (activity_main).....	65
<b>Εικόνα 31:</b> Το main.xml της εφαρμογής .....	66
<b>Εικόνα 32:</b> Παράδειγμα αποδοχής και απόρριψης φωτογραφίας.....	68
<b>Εικόνα 33:</b> Διόρθωση προσανατολισμού εικόνας .....	69
<b>Εικόνα 34:</b> Τελική επεξεργασία εικόνας (bitmap).....	70
<b>Εικόνα 35:</b> Αναγνώριση μέσω TessBaseAPI.....	71
<b>Εικόνα 36:</b> Καταχώρηση δεδομένων σε κατηγορίες .....	72
<b>Εικόνα 37:</b> Το MeasurementsActivity της εφαρμογής .....	74
<b>Εικόνα 38:</b> Το measurement.xml της εφαρμογής .....	75
<b>Εικόνα 39:</b> Διάγραμμα use-case χρήστη.....	75
<b>Εικόνα 40:</b> Αρχική σελίδα ιστοσελίδας .....	76
<b>Εικόνα 41:</b> Φόρμα προσθήκης χρήστη διαχειριστή.....	77
<b>Εικόνα 42:</b> Κεντρικό μενού διαχειριστή.....	78
<b>Εικόνα 43:</b> Πίνακες χρηστών συστήματος .....	79
<b>Εικόνα 44:</b> Φόρμα επεξεργασίας χρήστη.....	79
<b>Εικόνα 45:</b> Ο πίνακας μετρήσεων των τεχνικών .....	80

<b>Εικόνα 46:</b> Φόρμα επεξεργασίας Μέτρησης.....	81
<b>Εικόνα 47:</b> Διάγραμμα use-case Διαχειριστή .....	82
<b>Εικόνα 48:</b> Φόρμα εγγραφής χρήστη .....	83
<b>Εικόνα 49:</b> Φόρμα επικοινωνίας με διαχειριστή.....	84
<b>Εικόνα 50:</b> Η φόρμα εισόδου χρήστη .....	84
<b>Εικόνα 51:</b> Το προφίλ χρήστη.....	85
<b>Εικόνα 52:</b> Πίνακας πελατών τεχνικού.....	86
<b>Εικόνα 53:</b> Διάγραμμα use-case τεχνικού.....	86
<b>Εικόνα 54:</b> Χρήση φίλτρων μέσω της συνάρτησης filter_var .....	90
<b>Εικόνα 55 :</b> Δημιουργία εκτελέσιμου αρχείου εφαρμογής.....	103

## **Κατάλογος πινάκων**

<b>Πίνακας 1:</b> Ετικέτες HTML.....	22
<b>Πίνακας 2:</b> Πίνακας users .....	51
<b>Πίνακας 3:</b> Πίνακας User_Data .....	53



# Κεφάλαιο 1

## Εισαγωγή

Στη σύγχρονη καθημερινότητα, τα κινητά τηλέφωνα έχουν αντικατασταθεί σχεδόν πλήρως από μικρές πολυσκευές, γνωστές ως έξυπνα τηλέφωνα (smartphones). Οι δυνατότητες οι οποίες δίνουν αυτές οι συσκευές, ξεπερνούν κάθε μέρα όλο και περισσότερο κάθε προσδοκία καθώς ενσωματώνουν όλο και περισσότερα τεχνολογικά επιτεύγματα στο δυναμικό τους καθιστώντας έτσι τα κινητά τηλέφωνα κάτι πολύ περισσότερο από ένα μέσο αποστολής γραπτών μηνυμάτων ή πραγματοποίησης τηλεφωνικών κλήσεων. Ένα δείγμα των επιπλέον δυνατοτήτων είναι η περιήγηση στο διαδίκτυο, η λήψη φωτογραφιών και βίντεο πολύ μεγάλης ευκρίνειας, η διαχείριση προσωπικών λογαριασμών του χρήστη (λογαριασμού email, μέσου κοινωνικής δικτύωσης, τραπεζικού λογαριασμού), αλλά ακόμα και πιο απλές όπως αυτές της αναπαραγωγής μουσικής και εφαρμογών ψυχαγωγίας και παιχνιδιών.

Τα έξυπνα τηλέφωνα (smartphones), είναι κινητά τηλέφωνα βασισμένα σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερο προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα συμβατικό κινητό τηλέφωνο. Τα πρώτα smartphones συνδύαζαν τις λειτουργίες ενός προσωπικού ψηφιακού βοηθού (PDA) και ενός κινητού τηλεφώνου. Σε μεταγενέστερα μοντέλα προστέθηκαν οι λειτουργίες των φορητών media players, low-end compact ψηφιακές φωτογραφικές μηχανές, βιντεοκάμερες τσέπης, καθώς και μονάδες πλοήγησης GPS, με αποτέλεσμα την διαμόρφωση μιας πολυχρηστικής συσκευής. Πολλά σύγχρονα smartphones περιλαμβάνουν επίσης οθόνες αφής υψηλής ανάλυσης και web browsers που εμφανίζουν τυποποιημένες ιστοσελίδες, καθώς και βελτιστοποιημένες ιστοσελίδες για κινητά. Η πρόσβαση

σε δεδομένα υψηλής ταχύτητας παρέχεται μέσω Wi-Fi και μέσω κινητών ευρυζωνικών υπηρεσιών. Τα τελευταία χρόνια, η ταχεία ανάπτυξη στην αγορά των εφαρμογών για κινητά και στο εμπόριο κινητών τηλεφώνων έχει γίνει οδηγός για την ευρεία υιοθέτηση των smartphones.

Τα λειτουργικά συστήματα (OS) των κινητών τηλεφώνων που χρησιμοποιούνται από τα σύγχρονα smartphones περιλαμβάνουν, μεταξύ άλλων, το Android της Google, το iOS της Apple, το Symbian της Nokia, το BlackBerry OS της RIM, το Bada της Samsung, τα Windows Phone της Microsoft, το web OS της Hewlett-Packard, καθώς και ενσωματωμένες διανομές Linux όπως το Maemo και το MeeGo. Τέτοιου είδους λειτουργικά συστήματα μπορούν να εγκατασταθούν σε πολλά διαφορετικά μοντέλα κινητών τηλεφώνων και συνήθως κάθε συσκευή μπορεί να λάβει πολλές ενημερωμένες εκδόσεις λογισμικού λειτουργικού συστήματος κατά τη διάρκεια ζωής της. Μερικά άλλα επερχόμενα λειτουργικά συστήματα είναι το Firefox OS της Mozilla, το Ubuntu Phone της Canonical Ltd's και το Tizen.

Επιπλέον βρισκόμαστε στην εποχή που η εξοικείωση των ανθρώπων με τη χρήση φυλλομετρητών καθώς και την περιήγηση και χρήση ιστοσελίδων στο διαδίκτυο είναι πλέον δεδομένη και αποτελεί ένα σημαντικότατο εργαλείο για την εύκολη και άμεση εξυπηρέτηση μίας τεράστιας λίστας καθημερινών αναγκών και δραστηριοτήτων.

Η εργασία αυτή προσεγγίζει συνδυαστικά μία από τις πιο συνηθισμένες προκλήσεις στους τομείς που μόλις αναφέραμε. Συνδυάζει την αρμονική λειτουργία και συνεργασία της χρήσης εφαρμογής σε κινητό τηλέφωνο με λειτουργικό σύστημα Android με την αντίστοιχη της ιστοσελίδα για να φέρει σε πέρας την πραγματοποίηση μετρήσεων κοινής ωφέλειας με την ταχύτερη μέθοδο της οπτικής αναγνώρισης.

Ο τωρινός τρόπος καταγραφής μετρήσεων τη δεδομένη χρονική στιγμή ακολουθεί την παρακάτω διαδικασία:

- Οι μετρήσεις που αντιστοιχούν σε κάθε καταναλωτή υπάρχουν αποθηκευμένες σε καταμετρητές της εκάστοτε εταιρίας στο χώρο του καταναλωτή.



- Σε συγκεκριμένα χρονικά διαστήματα, οι τεχνικοί των εταιριών συλλέγουν τα δεδομένα που είναι αποθηκευμένα από τους καταμετρητές.
- Αφού γίνει η καταγραφή των ενδείξεων, τα δεδομένα καταχωρούνται στον κάθε καταναλωτή ξεχωριστά.
- Έχοντας όλες αυτές τις πληροφορίες, το μηχανογραφικό σύστημα κάθε εταιρίας υπολογίζει την κατανάλωση η οποία ορίζεται ως η διαφορά της τελευταίας ένδειξης από αυτήν που είχε καταγραφεί πριν από το χρονικό διάστημα που ορίζει η εταιρία.
- Τέλος, ο καταναλωτής ενημερώνεται με την αποστολή από την εταιρία του αντίστοιχου πιστοποιητικού.

Το κίνητρο που δημιουργείται λοιπόν εδώ, είναι η βελτιστοποίηση της παραπάνω διαδικασίας συνολικά για την ίδια την εταιρία, τους τεχνικούς της, αλλά ακόμα και για τον κάθε ένα καταναλωτή ξεχωριστά. Αυτό μπορεί να επιτευχθεί με την οπτική αναγνώριση μέσω της εφαρμογής Android καθώς και της χρήσης της ιστοσελίδας που διαχειρίζεται τα δεδομένα των καταγραφών.

Είναι σημαντικό να σημειωθεί πως αντίστοιχες καινοτόμες εφαρμογές έχουν αναπτυχθεί από μεγάλες εταιρίες.

Τα πρωτεία τη δεδομένη χρονική στιγμή στις εφαρμογές οπτικής αναγνώρισης με κύρια βάση τα κείμενα, κατέχει η εφαρμογή OmniPage Standard 18 [1] της εταιρίας Nuance. Η εφαρμογή προσφέρει τρομερά αξιόπιστα αποτελέσματα, δοκιμασμένη με μεγάλη επιτυχία σε διαφορετικές μορφές και απεικονίσεις κειμένου.

Η Adobe [2], μία από τις πιο γνωστές εταιρίες παγκοσμίως σε πάρα πολλούς τομείς, έχει στο δυναμικό της μίας από τις καλύτερες εφαρμογές οπτικής αναγνώρισης, την Adobe Pro DC (OCR). Προσφέροντας εκπληκτική ευκρίνεια και τη δυνατότητα πολλαπλής αναγνώρισης από εικόνα ή κείμενο.

Τέλος, η εταιρία ABBYY [3] είναι μία από τις μεγαλύτερες και πιο επιτυχημένες στον τομέα της οπτικής αναγνώρισης. Καλύπτοντας ένα ευρύτατο φάσμα στον τομέα της οπτικής αναγνώρισης κειμένων και στην ακριβή απεικόνιση έχει δημιουργήσει την εφαρμογή ABBYY Fine Reader χρησιμοποιώντας τεχνολογίες με βάση γλωσσικά στοιχεία εξυπηρετώντας έτσι

άμεσα και με μεγάλη επιτυχία πληθώρα χρηστών. Η εταιρία μετρά σημαντικά βραβεία για τη συγκεκριμένη εφαρμογή, αλλά και το γενικότερο έργο της.

Εκτός από το λογισμικό το οποίο κατασκευάστηκε από τις παραπάνω εταιρίες, υπάρχουν και άλλα εργαλεία λογισμικού ανοιχτού κώδικα ευρέως διαδεδομένα και εύχρηστα μετά από μερική επεξεργασία.

Για την ακόμη καλύτερη απεικόνιση και παρουσίαση του αντικειμένου, η εφαρμογή συνεργάζεται με μία ιστοσελίδα που αναπτύχθηκε στα πλαίσια της εργασίας. Σχεδιασμένη με προφίλ εταιρίας, παρέχει σε διαφορετικά επίπεδα χρήστη τις δυνατότητες που αντιστοιχούν και διευκολύνει την αποθήκευση και εποπτεία των δεδομένων.

Το παρόν κείμενο είναι δομημένο από 5 κεφάλαια που καλύπτουν όλο το πεδίο γνώσεων και απαιτήσεων που συνέβαλαν στην πραγματοποίηση της διπλωματικής εργασίας.

Στο παρόν κεφάλαιο παρουσιάστηκε το αντικείμενο που πραγματεύεται η εργασία καθώς και η γενικότερη δομή της.

Στο 2<sup>ο</sup> κεφάλαιο θα γίνει αναφορά στο θεωρητικό υπόβαθρο του συστήματος που υλοποιήθηκε, με αναφορές σε όλα τα τεχνολογικά μέσα και εργαλεία που χρησιμοποιήθηκαν ώστε να φέρουμε σε πέρας την πραγματοποίησή του.

Στο 3<sup>ο</sup> κεφάλαιο θα γίνει περιγραφή της αρχιτεκτονικής του συστήματος και θα οριστούν όλες οι δυνατότητες και οι απαιτήσεις του, που παρουσιάστηκαν πριν αλλά και κατά τη διάρκεια υλοποίησής του.

Στο 4<sup>ο</sup> κεφάλαιο θα πραγματοποιήσουμε αναλυτική αναφορά και περιγραφή, όλων των τμημάτων λογισμικού που αναπτύχθηκαν στα πλαίσια της παρούσας διπλωματικής εργασίας. Θα παρουσιαστούν λεπτομερώς οι λειτουργίες και οι δυνατότητες του συστήματος που υλοποιήθηκε και θα γίνουν αναφορές στην ασφάλεια που συνοδεύει το σύστημα σε όλες του τις πτυχές. Τέλος θα γίνει αναφορά στις δοκιμές που πραγματοποιήθηκαν με την ολοκλήρωσή του.

Στο 5<sup>ο</sup> κεφάλαιο θα γίνει αναφορά στα συμπεράσματα που προέκυψαν με την πραγματοποίησή του συστήματος, καθώς και στα πλεονεκτήματα και τα

μειονεκτήματα που το συνοδεύουν. Επίσης θα γίνει αξιολόγηση των δυνατών και αδύνατων σημείων του, και θα αναφερθούν οι πιθανές μελλοντικές του επεκτάσεις.

Τέλος, θα παρατεθεί ένα παράρτημα με οδηγίες εγκατάστασης των επιμέρους τμημάτων που υλοποιήθηκαν.

## **Σύνοψη κεφαλαίου**

Στο κεφάλαιο αυτό αναφερθήκαμε στα κίνητρα που μας οδήγησαν στην ανάπτυξη αυτού του συστήματος, έγινε περιγραφή του γενικού μοντέλου της εργασίας με διευκρινήσεις σε συγκεκριμένες προσθήκες για την επίτευξη του στόχου και αναφέρθηκαν κάποια παραδείγματα παρόμοιων εφαρμογών, κλείνοντας με τη δομή των κεφαλαίων που θα ακολουθήσουν.

## Κεφάλαιο 2

### Θεωρητικό υπόβαθρο

Για την πραγματοποίηση αυτής της διπλωματικής εργασίας, χρησιμοποιήθηκαν διάφορες πλατφόρμες ανάπτυξης λογισμικού καθώς και πολλά επιπλέον εργαλεία διαμόρφωσης και πραγματοποίησης συγκεκριμένων ενεργειών.

Στο κεφάλαιο αυτό αναφέρονται και αναλύονται τα παραπάνω, όπως επίσης εισάγονται και έννοιες για την καλύτερη και ευκολότερη κατανόηση της συνολικής λειτουργίας της εργασίας.

### Προγραμματισμός διαδικτύου

Ένα από τα σημαντικότερα κομμάτια του συστήματος που δημιουργήθηκε, είναι η ιστοσελίδα. Για το σχεδιασμό της χρησιμοποιήθηκαν γλώσσες προγραμματισμού που καλύπτουν τις απαιτήσεις σύστασης μιας δυναμικής και λειτουργικής ιστοσελίδας όπως επίσης και τις ανάγκες της συγκεκριμένης εργασίας. Επιπλέον σχεδιάστηκε και δημιουργήθηκε και μία βάση δεδομένων η οποία συμπεριλαμβάνει όλες τις πληροφορίες που είναι απαραίτητες για την σωστή λειτουργία της ιστοσελίδας αλλά και της εφαρμογής.

#### 2.1 HTML

Η HTML [4] (HyperText Markup Language ή Γλώσσα Σήμανσης Υπερκειμένου) είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

Η HTML γράφεται υπό μορφή στοιχείων HTML τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα «μεγαλύτερο από» και «μικρότερο από» (για παράδειγμα `<html>`), μέσα στο περιεχόμενο της ιστοσελίδας. Οι ετικέτες HTML συνήθως λειτουργούν ανά ζεύγη (για παράδειγμα `<h1>` και `</h1>`), με την πρώτη να ονομάζεται ετικέτα έναρξης και τη δεύτερη ετικέτα λήξης (ή σε άλλες περιπτώσεις ετικέτα ανοίγματος και ετικέτα κλεισίματος αντίστοιχα). Ανάμεσα στις ετικέτες, οι σχεδιαστές ιστοσελίδων μπορούν να τοποθετήσουν κείμενο, πίνακες, εικόνες κλπ. Ο σκοπός ενός φυλλομετρητή (web browser) είναι να διαβάσει τα έγγραφα HTML και να τα συνθέτει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Ο browser δεν εμφανίζει τις ετικέτες HTML, αλλά τις χρησιμοποιεί για να ερμηνεύσει το περιεχόμενο της σελίδας.

Τα στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους τους ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων (δηλαδή εγγράφων που αποτελούνται από το περιεχόμενο που μεταφέρουν και από τον κώδικα μορφοποίησης του περιεχομένου) καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML. Στην Εικόνα 1, εμφανίζεται το λογότυπο της HTML.



Εικόνα 1: Το λογότυπο της HTML

Στη συνέχεια δίνεται ο πίνακας με τις HTML ετικέτες που χρησιμοποιήθηκαν για την δημιουργία της ιστοσελίδας (Πίνακας 1).

<b>Ετικέτες</b>	<b>Επεξήγηση</b>
<code>&lt;!DOCTYPE&gt;</code>	Ορίζει τον αριθμό έκδοσης της HTML του αρχείου. Αυτή η ετικέτα πρέπει να βρίσκεται πάντα πρώτη μέσα στο HTML έγγραφο.
<code>&lt;html&gt;&lt;/html&gt;</code>	Ορίζει ότι το αρχείο είναι HTML και βρίσκεται στην αρχή. Σε αυτήν συμπεριλαμβάνονται όλες οι υπόλοιπες ετικέτες.
<code>&lt;head&gt;&lt;/head&gt;</code>	Από τις βασικότερες ετικέτες. Ορίζει στοιχεία όπως το είδος των χαρακτήρων που χρησιμοποιούνται (π.χ. utf-8), εξωτερικές πηγές αρχείων κλπ.
<code>&lt;title&gt;&lt;/title&gt;</code>	Ορίζει τον τίτλο της ιστοσελίδας στον εκάστοτε φυλλομετρητή.
<code>&lt;meta&gt;&lt;/meta&gt;</code>	Ορίζει τα μεταδεδομένα μέσα σε ένα έγγραφο HTML
<code>&lt;body&gt;&lt;/body&gt;</code>	Ορίζει το σώμα του εγγράφου και περικλείει όλα τα περιεχόμενα όπως κείμενα, υπερσυνδέσμους, πίνακες, λίστες, εικόνες κλπ.
<code>&lt;script&gt;&lt;/script&gt;</code>	Ορίζει την χρήση εξωτερικού αρχείου που εκτελείται τοπικά.
<code>&lt;style&gt;&lt;/style&gt;</code>	Ορίζει την μορφοποίηση της σελίδας.
<code>&lt;li&gt;&lt;/li&gt;</code>	Ορίζει ένα αντικείμενο λίστας.
<code>&lt;ul&gt;&lt;/ul&gt;</code>	Ορίζει μία μη διατεταγμένη λίστα.

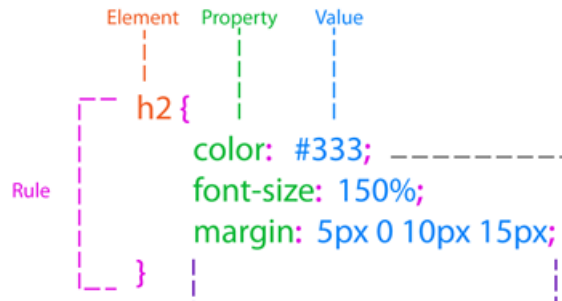
<code>&lt;p&gt;&lt;/p&gt;</code>	Ορίζει μια παράγραφο.
<code>&lt;h1&gt;&lt;/h1&gt; ... &lt;h4&gt;&lt;/h4&gt;</code>	Ορίζουν το μέγεθος των γραμμάτων στις επικεφαλίδες (ισχύει φθίνουσα πορεία).
<code>&lt;a&gt;&lt;/a&gt;</code>	Ορίζει υπερσύνδεσμο σε άλλη ιστοσελίδα.
<code>&lt;img&gt;&lt;/img&gt;</code>	Ορίζει την εισαγωγή εικόνας.
<code>&lt;table&gt;&lt;/table&gt;</code>	Ορίζει την εισαγωγή πίνακα.
<code>&lt;div&gt;&lt;/div&gt;</code>	Ορίζει ένα τμήμα του εγγράφου, ώστε να μπορεί να χρησιμοποιηθεί επανειλημμένα με τη βοήθεια ενός id.
<code>&lt;button&gt;&lt;/button&gt;</code>	Ορίζει ένα κουμπί με κάποια λειτουργία.
<code>&lt;form&gt;&lt;/form&gt;</code>	Ορίζει την ύπαρξη φόρμας.
<code>&lt;input&gt;&lt;/input&gt;</code>	Ορίζει ένα πεδίο εισαγωγής δεδομένων
<code>&lt;select&gt;&lt;/select&gt;</code>	Ορίζει ένα πεδίο επιλογής μια τιμής.

Πίνακας 1: Ετικέτες HTML

## 2.2 Μορφοποίηση CSS

Η μορφοποίηση CSS [5] (Cascading Style Sheets - Αλληλουχία Φύλλων Στυλ), είναι μια γλώσσα υπολογιστή, που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει το παρουσιαστικό μιας ιστοσελίδας δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες

δυνατότητες σε σχέση με την html. Στην παρακάτω εικόνα (Εικόνα 2) φαίνεται αναλυτικά ο κανόνας σύνταξης CSS.



**Εικόνα 2:** Κανόνας σύνταξης CSS

Η χρήση της, στην ιστοσελίδα που δημιουργήθηκε στα πλαίσια της εργασίας, ήταν απαραίτητη ώστε να υπάρχει ένα εμφανίσιμο και ελκυστικό αποτέλεσμα. Αναφορικά κάποια από τα στοιχεία που παρέχει η CSS είναι τα εξής:

- Παρέχει τρομερή ευελιξία στην εμφάνιση της ιστοσελίδας καθώς δίνονται δυνατότητες οι οποίες δεν είναι εφικτές μόνο με τη χρήση της HTML.
- Κάθε κανόνας μορφοποίησης γράφεται μία φορά και δίνεται η δυνατότητα χρήσης του, οποτεδήποτε επιθυμούμε απλά και μόνο με την κλήση του αντίστοιχου αρχείου και κανόνα, κάτι που καθιστά πολύ πιο ευανάγνωστο το κείμενο της HTML.
- Εύκολη και γρήγορη μορφοποίηση, καθώς ολόκληρη η εμφάνιση ενός ιστοχώρου μπορεί να επεξεργαστεί από ένα και μόνο εξωτερικό αρχείο CSS.
- Την πρώτη φορά που ένας φυλλομετρητής θα επισκεφθεί μία ιστοσελίδα που συνοδεύεται από εξωτερικό αρχείο CSS, θα το αποθηκεύσει στην cache, οπότε δε χρειάζεται να το φορτώνει ξεχωριστά κάθε φορά που ο χρήστης θέλει να επισκεφθεί την ιστοσελίδα.



## 2.3 PHP

Η PHP [6] είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Είναι γλώσσα βασισμένη σε σενάρια και κατάλληλη για την ανάπτυξη διαδικτυακών εφαρμογών και προγραμμάτων συστήματος. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML. Κάποια από τα σημαντικότερα στοιχεία που προσφέρει η PHP είναι:

- Περιέχει ενσωματωμένες βιβλιοθήκες για συνηθισμένες διαδικασίες διαδικτύου
- Δημιουργεί διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων
- Είναι γλώσσα ανοιχτού κώδικα και υποστηρίζει πολλές πλατφόρμες
- Βοηθά στο δυναμικό σχεδιασμό της ιστοσελίδας
- Μπορεί να χρησιμοποιηθεί για δημιουργία session ή cookies

Στην παρακάτω εικόνα (Εικόνα 3) ακολουθεί ένα παράδειγμα κώδικα σε PHP.

```

1 <?php
2 $str_isset = "";
3 $bol_isset = isset($str_isset);
4
5 If ($bol_isset){
6     echo "The variable is set";
7 }
8 else {
9     echo "The variable is not set";
10 }
11 ?>
12

```

**Εικόνα 3:** Παράδειγμα κώδικα php

Ο κώδικας της PHP γράφεται μέσα στα αρχεία της HTML και περικλείεται μέσα στις ετικέτες <?php ?>. Ο φυλλομετρητής κάνει ένα αίτημα στον διακομιστή και η απάντηση που λαμβάνει είναι σε μορφή HTML όπου μπορεί να την διαχειριστεί και να την προβάλλει. Ο κώδικας της PHP τρέχει πάντα κατά τη διάρκεια της φόρτωσης

της ιστοσελίδας. Ένα αρχείο της PHP έχει κατάληξη .php και μπορεί είτε να αποτελεί μέρος μια ιστοσελίδας είτε να είναι αρχείο που δεν χρειάζεται την παρέμβαση του φυλλομετρητή για να τρέξει.

## 2.4 MySQL

Η MySQL [7] είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που μετρά περισσότερες από 11 εκατομμύρια εγκαταστάσεις. Τα δεδομένα στη MySQL αποθηκεύονται σε αντικείμενα βάσης δεδομένων, τα οποία ονομάζονται πίνακες. Ένας πίνακας, είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές. Οι βάσεις δεδομένων είναι χρήσιμες για την αποθήκευση πληροφοριών σε κατηγορίες. Η διαχείριση των δεδομένων στις βάσεις, γίνονται μέσω των SQL εντολών, στη MySQL.

Αναφορικά κάποιες από τις πιο συνηθισμένες και εύχρηστες λειτουργίες που πραγματοποιούνται σε μία βάση δεδομένων είναι οι εξής:

- Δημιουργία / Επεξεργασία ή Διαγραφή ενός πίνακα στη βάση δεδομένων
- Δημιουργία / Ανάγνωση / Ενημέρωση ή Διαγραφή δεδομένων ενός πίνακα στη βάση δεδομένων (γνωστό ως CRUD – Create-Read-Update-Delete)
- Αναζήτηση δεδομένων σε πίνακα της βάσης δεδομένων



Εικόνα 4: Λογότυπο της MySQL

## 2.5 JavaScript

Η JavaScript [8] είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και τα πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

Αναφορικά κάποιες από τις βασικές λειτουργίες της JavaScript είναι οι εξής:

- Επικοινωνία με php, xml, json αρχεία, καθώς και με άλλες ιστοσελίδες
- Δυναμική αλλαγή ετικετών και περιεχομένου HTML
- Δυνατότητα χειρισμού CSS
- Δυνατότητα αλλαγής περιεχομένων ιστοσελίδας

- Εκτέλεση ενεργειών μετά από προκαθορισμένο συμβάν

Το γεγονός ότι η υλοποίηση της JavaScript γίνεται με μορφή υπολογιστής – πελάτης (client-side), παρέχει ασφάλεια στα δεδομένα που εισάγονται καθώς ελέγχεται η εγκυρότητα τους πριν αποσταλούν στο διακομιστή (server). Έτσι αποφεύγονται η επιθέσεις από κακόβουλους χρήστες.

Στην παρακάτω εικόνα (Εικόνα 5) φαίνεται ένα παράδειγμα κώδικα JavaScript, ενσωματωμένο μέσα σε κώδικα HTML.

```
</head>
<body>

<script type="application/x-javascript">
  var field;
  field = document.getElementById("main");
  field.setAttribute("class", "foo");
</script>

<div id="infobox"></div>
<div class="code html">
```

Εικόνα 5: Παράδειγμα κώδικα JavaScript μέσα σε κώδικα HTML

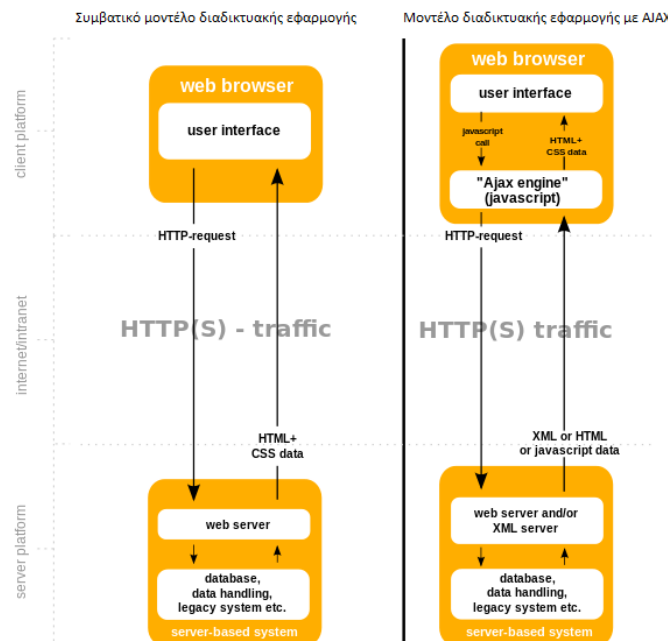
## 2.6 AJAX

Η αρχιτεκτονική AJAX [9] (Asynchronous JavaScript and XML) είναι ένα σύνολο τεχνικών ανάπτυξης, που χρησιμοποιεί πολλές τεχνολογίες web για τη δημιουργία ασύγχρονων εφαρμογών. Με την AJAX διαδικτυακές εφαρμογές μπορούν να στείλουν και να λάβουν δεδομένα σε ένα διακομιστή ασύγχρονα στο παρασκήνιο, χωρίς να παρεμβαίνει στην εμφάνιση και την συμπεριφορά της υπάρχουσας σελίδας. Με την αποσύνδεση του στρώματος (layer) ανταλλαγής δεδομένων από αυτό της παρουσίασης, η AJAX επιτρέπει στις ιστοσελίδες και κατ' επέκταση στις διαδικτυακές εφαρμογές, να αλλάξουν δυναμικά το περιεχόμενο της σελίδας χωρίς να χρειάζεται να φορτωθεί εκ νέου ολόκληρη η σελίδα.

Ο τρόπος λειτουργίας της αρχιτεκτονικής μπορεί να περιγραφεί συνοπτικά με τα ακόλουθα βήματα:

1. Δημιουργείται επικοινωνία φυλλομετρητή και διακομιστή μέσω του αντικειμένου XMLHttpRequest
2. Η διαχείριση του αντικειμένου αυτού καλύπτεται από την JavaScript η οποία στέλνει ένα αίτημα HTML όταν το αντικείμενο αυτό καλεστεί
3. Ο διακομιστής λαμβάνει αυτό το αίτημα, το επεξεργάζεται κατάλληλα και απαντά με δεδομένα μορφής XML ή JSON
4. Ο φυλλομετρητής εμφανίζει τα HTML και CSS με τα δεδομένα που δέχθηκε.

Στην εικόνα που ακολουθεί (Εικόνα 6), εμφανίζεται ένα συμβατικό μοντέλο διαδικτυακής εφαρμογής, και ένα μοντέλο που χρησιμοποιεί αρχιτεκτονική AJAX.



Εικόνα 6: Παράδειγμα διαδικτυακής εφαρμογής με και χωρίς AJAX

## 2.7 Android Studio

Το Android Studio [10] είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Το Android Studio είναι διαθέσιμο δωρεάν με την άδεια Apache License 2.0.

Το Android Studio ήταν διαθέσιμο σε πρώιμο στάδιο για προεπισκόπηση ξεκινώντας από την έκδοση 0.1 τον Μάιο του 2013, έπειτα ξεκίνησε το δοκιμαστικό στάδιο από την έκδοση 0.8 που βγήκε τον Ιούνιο του 2014. Η πρώτη σταθερή έκδοση βγήκε το Δεκέμβριο του 2014, με την έκδοση 1.0. Βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, το Android Studio σχεδιάστηκε αποκλειστικά για προγραμματισμό Android. Είναι διαθέσιμο για Windows, Mac OS X και Linux, και αντικατέστησε το Eclipse Android Development Tools (ADT) ως το κύριο IDE της Google για ανάπτυξη εφαρμογών Android.

Το Android Studio χρησιμοποιεί το Android SDK (Software Development Kit) το οποίο περιλαμβάνει τα απαραίτητα εργαλεία και βιβλιοθήκες για την ανάπτυξη εφαρμογών στο Android.

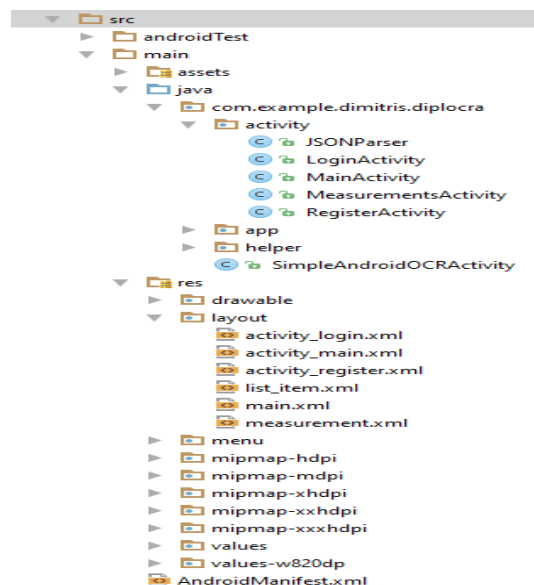
Η σχεδίαση εφαρμογής Android, που είναι ένας από τους βασικούς πυλώνες της παρούσας διπλωματικής εργασίας, επηρεάζονται από το λειτουργικό σύστημα Android αλλά και από τις βιβλιοθήκες / ενέργειες που εισάγονται στην εφαρμογή. Χαρακτηριστικό παράδειγμα αποτελεί η χρήση του OCR στην εφαρμογή η οποία έχει ελάχιστη απαίτηση την έκδοση λειτουργικού συστήματος 2.3 και άνω, όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο. Η επόμενη εικόνα (Εικόνα 7), παρουσιάζει το λογότυπο του Android Studio.



**Εικόνα 7:** Το λογότυπο του Android Studio

Η εφαρμογή Android για έξυπνα για συσκευές που υποστηρίζουν το λειτουργικό, δημιουργήθηκε ώστε να εισάγουμε την δυνατότητα της οπτικής αναγνώρισης με την οποία οι χρήστες της μπορούν να καταγράφουν και να αποστέλλουν τα δεδομένα τους άμεσα στη βάση δεδομένων. Η εφαρμογή απαιτεί την σύνδεση της συσκευής στο διαδίκτυο, είτε μέσω ασύρματου δικτύου (Wi-Fi), είτε μέσω δορυφορικού δικτύου εταιρίας (3G / 4G). Επιπλέον υπάρχει η απαίτηση εγκατεστημένης κάμερας στη συσκευή από την οποία χρησιμοποιείται. Τέλος, για την ορθή λειτουργία της αναγνώρισης, θα πρέπει η συσκευή να έχει εγκατεστημένο λειτουργικό σύστημα έκδοσης 2.3 ή νεότερη.

Στην εικόνα που ακολουθεί (Εικόνα 8), εμφανίζεται η δομή του Android Project που υλοποιήθηκε.



**Εικόνα 8:** Δομή του Android Project

Στην δομή της εικόνας αρχικά εντοπίζουμε τον κατάλογο src. Στον κατάλογο αυτό συμπεριλαμβάνονται όλα τα Java.class αρχεία που αποτελούν τις δραστηριότητες (Activities) της εφαρμογής. Στον κατάλογο res συμπεριλαμβάνονται όλα τα xml αρχεία τα οποία αποτελούν ουσιαστικά το περιεχόμενο που βλέπει ο κάθε χρήστης στην οθόνη του. Τέλος, μέσα στο γενικότερο κατάλογο διακρίνουμε το αρχείο AndroidManifest.xml. Στο αρχείο αυτό δηλώνονται όλες οι άδειες που απαιτεί η εφαρμογή, καθώς και όλα τα Activities που χρησιμοποιεί η εφαρμογή.

Το βασικό εργαλείο, που χρησιμοποιήθηκε στην πραγματοποίηση εφαρμογής με οπτική αναγνώριση σε αυτήν την εργασία, είναι το Tesseract [11]. Το Tesseract είναι μία μηχανή οπτικής αναγνώρισης χαρακτήρων για διάφορα λειτουργικά συστήματα, ανεπτυγμένη και υποστηριζόμενη από την Google. Αναπτύχθηκε αρχικά ως ιδιόκτητο λογισμικό στα εργαστήρια της Hewlett Packard στο Μπρίστολ και το Κολοράντο σχεδόν για μία δεκαετία (1985-1994). Το 1996 υπέστη τροποποιήσεις ώστε να μπορεί να είναι διαθέσιμο σε λειτουργικά συστήματα Windows και 2 χρόνια αργότερα μεταφέρθηκε από περιβάλλον γλώσσας προγραμματισμού C σε C++. Ο κυριότερος λόγος της μετατροπής ήταν ο μεγάλος όγκος κώδικα που υπήρχε και η δυνατότητα να μπορεί να μεταγλωττιστεί τουλάχιστον με ένα μεταγλωττιστή της C++. Στα επόμενα χρόνια έγινε ελάχιστη δουλειά σε αυτόν τον τομέα, με αποτέλεσμα να εκδοθεί ως open source code το 2005 από την Hewlett Packard και το πανεπιστήμιο της Νεβάδα. Η ανάπτυξη του Tesseract χρηματοδοτείται από την Google από το 2007.

Το Tesseract μέχρι και την 2<sup>η</sup> έκδοση του μπορούσε να δεχθεί ως δεδομένα μόνο εικόνες τύπου TIFF απλού κειμένου μονής στήλης. Στις πρώτες εκδόσεις δεν συμπεριλαμβανόταν ανάλυση της διάταξης της εικόνας, οπότε και η αποτύπωση πολλών αλφαριθμητικών δεδομένων ταυτόχρονα είχε ως έξοδο πολύ αλλοιωμένα αποτελέσματα. Από την έκδοση 3.00 κι έπειτα το Tesseract υποστηρίζει μορφοποίηση κειμένου εξόδου, πληροφορίες θέσης και ανάλυση διάταξης. Επιπλέον για την υποστήριξη πολλών format εικόνων προστέθηκε η βιβλιοθήκη Leptonica [12].

Για την προσθήκη του Tesseract σε λογισμικό Android χρησιμοποιήθηκε μία επέκτασή του ειδικά σχεδιασμένη για εργαλεία και λειτουργίες σε πλατφόρμες ανάπτυξης Android (Tesseract Tools for Android). Για την ορθή προσθήκη της στην πλατφόρμα του Android Studio όπου πραγματοποιήθηκε η εφαρμογή, ήταν απαραίτητη η προσθήκη της εργαλειοθήκης NDK [13] (Native Development Kit).

Η εργαλειοθήκη NDK είναι η γέφυρα με την οποία γίνεται δυνατό να χρησιμοποιηθούν βιβλιοθήκες γραμμένες σε γλώσσα C και C++ (αλλά και άλλες



γλώσσες) στο προγραμματιστικό περιβάλλον του Android. Μόλις οι βιβλιοθήκες αυτές μεταγλωττιστούν σε ARM, MIPS ή x86 native code, γίνεται εγκατάσταση τους μέσω NDK. Από αυτό το σημείο κι έπειτα μπορεί να γίνει κλήση από κώδικα Java μέσω του Dalvik Virtual Machine που αποτελεί κομμάτι των βασικών κλάσεων Android Java.

Σε αντίθεση με την ανάπτυξη εφαρμογών σε γλώσσα προγραμματισμού Java σε ένα προγραμματιστικό περιβάλλον, όπως το Android Studio, το NDK βασίζεται σε εργαλεία της γραμμής εντολών και απαιτεί την κλήση τους χειροκίνητα ώστε να κατασκευάσει, να εγκαταστήσει και να πραγματοποιήσει εντοπισμό σφαλμάτων στις εφαρμογές στις οποίες χρησιμοποιείται.

Η έκδοση του NDK που χρησιμοποιήθηκε για την σωστή εγκατάσταση και λειτουργία εξωτερικών βιβλιοθηκών, είναι συγκεκριμένη για κάθε λειτουργικό σύστημα στο οποίο εκτελείται από τον εκάστοτε προγραμματιστή. Μετά την επιτυχή εγκατάσταση του NDK και της βιβλιοθήκης που απαιτεί η περίπτωση, ο προγραμματιστής μπορεί να ανατρέξει και να χρησιμοποιήσει οποιαδήποτε στιγμή επιθυμεί τη βιβλιοθήκη που χρειάζεται χωρίς να πραγματοποιήσει περαιτέρω ενέργειες. Βέβαια, παρόλο που είναι εφικτό να μεταγλωττιστούν και να εγκατασταθούν κανονικά ολοκληρωμένες εφαρμογές χρησιμοποιώντας τα βασικά εργαλεία προγραμματισμού Android, σύμφωνα με την τεκμηρίωση Android, το NDK δεν θα πρέπει να χρησιμοποιείται από τον προγραμματιστή ως μοναδικό μέσο για ανάπτυξη κώδικα σε C/C++, καθώς αυξάνει κατά πολύ την πολυπλοκότητα της εφαρμογής, κάτι το οποίο δεν είναι επιθυμητό για την λειτουργικότητα των εφαρμογών και θα πρέπει να αποφεύγεται.

Με βασική προϋπόθεση λοιπόν την εισαγωγή του NDK, για να επιτευχθεί η λειτουργία της οπτικής αναγνώρισης, έγινε εισαγωγή της βιβλιοθήκης tess-two [14].

Η βιβλιοθήκη tess-two είναι μία επέκταση των ειδικά σχεδιασμένων εργαλείων του Tesseract γύρω από το Android. Συγκεκριμένα, λειτουργεί με Tesseract 3.05 00dev, Leptonica 1.73, libjpeg 9b και libpng 1.6.20. Η ενότητα tess-two περιέχει εργαλεία που μεταγλωττίζουν τις βιβλιοθήκες Tesseract και

Leptonica ώστε να χρησιμοποιηθούν σε πλατφόρμες Android. Αυτό γίνεται εφικτό καθώς παρέχεται ένα Java API (Application Programming Interface) το οποίο δίνει πρόσβαση στα μεταγλωττισμένα APIs των Tesseract και Leptonica.

Να σημειωθεί πως μέσα στο tess-two περιέχεται ακόμη μία ενότητα με όνομα eyes-two. Αυτή περιέχει κώδικα από το eyes-free project και συμβάλει στην καλύτερη επεξεργασία εικόνων παρέχοντας εργαλεία όπως ακριβής εντοπισμός κειμένου, εντοπισμός θαμπάδας κ.ά.

Για τη χρήση του tess-two βέβαια, υπάρχουν κάποια προαπαιτούμενα από το σύστημα. Αυτά είναι :

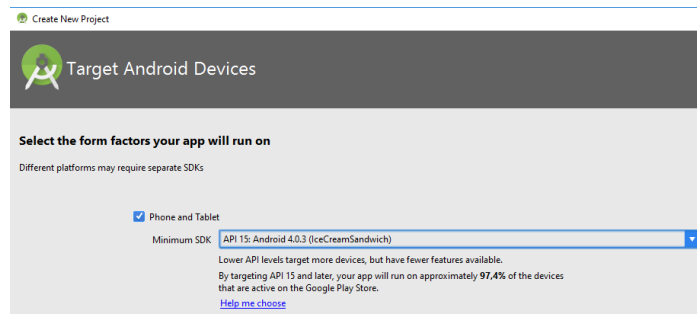
- χρήση πλατφόρμας Android έκδοσης 2.3 και άνω
- χρήση ενός εκπαιδευμένου αρχείου δεδομένων (trained data file) έκδοσης 3.02+ και άνω, το οποίο προσδιορίζει την γλώσσα στη οποία θα γίνει αναγνώριση.

Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν εκπαιδευμένα αρχεία δεδομένων για αναγνώριση αλφαριθμητικών δεδομένων με βάση του λατινικούς χαρακτήρες. Η προσθήκη άλλων γλωσσών απαιτεί πολύ μικρές τροποποιήσεις στον πηγαίο κώδικα.

Η εφαρμογή που αναπτύχθηκε μπορεί να βρεθεί στο Google App Store και είναι δωρεάν. Επιπλέον, μπορεί να εγκατασταθεί και να εκτελεστεί χωρίς κανένα πρόβλημα σε οποιαδήποτε συσκευή λειτουργεί με λογισμικό Android έκδοσης 2.3 και άνω.

### **2.7.1 Εγκατάσταση NDK και Βιβλιοθήκης tess-two**

Σε πρώτο στάδιο δημιουργήθηκε ένα κενό project μέσα στο Android Studio, χρησιμοποιώντας σχεδόν όλες τις προεπιλεγμένες ρυθμίσεις, εκτός από αυτήν της έκδοσης, όπως απεικονίζεται παρακάτω (Εικόνα 9), όπου επιλέχθηκε αρκετά ανώτερή της 2.1 (προτεινόμενη) για λόγους συμβατότητας.



**Εικόνα 9:** Επιλογή έκδοσης κατά τη δημιουργία του project

Για να ξεκινήσει η δημιουργία της εφαρμογής απαραίτητη προϋπόθεση αποτέλεσε η εκτέλεση κάποιων ενεργειών εκτός του προγραμματιστικού περιβάλλοντος του Android Studio, που θα είχαν ως αποτέλεσμα την εισαγωγή όλων των προαπαιτούμενων λειτουργιών και βιβλιοθηκών μέσα σε αυτό. Οι ενέργειες αυτές σχετίζονται με την σωστή λειτουργία του NDK μέσα στην πλατφόρμα, καθώς και με την ορθή εισαγωγή της βιβλιοθήκης tess-two (επέκταση του Tesseract όπως προαναφέρθηκε).

Αρχικά έγινε λήψη της τελευταίας έκδοσης του NDK (android-ndk-r10c-windows-x86\_64.exe) μέσω της ιστοσελίδας του (<https://developer.android.com/ndk/index.html>) και εγκαταστάθηκε στο σύστημα. Στις εκδόσεις που ακολούθησαν αργότερα, το NDK μπορούσε να εισαχθεί μέσα στο ίδιο το Android Studio μέσω της επιλογής του ADT (Android Development Tools). Η τελευταία έκδοση του Android Studio που κυκλοφόρησε κατά τη συγγραφή της διπλωματικής εργασίας, συμπεριλαμβάνει εξ ολοκλήρου το NDK στο λογισμικό της.

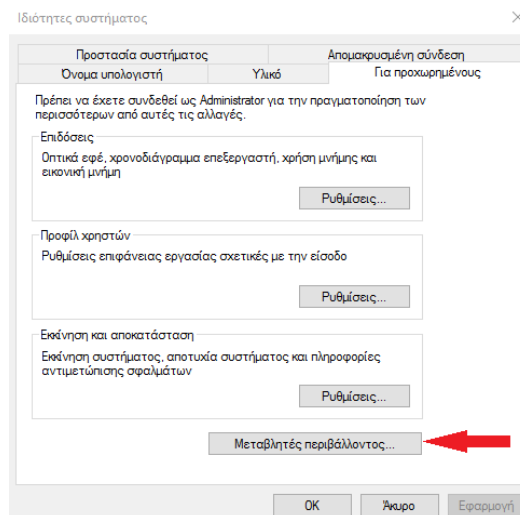
Στη συνέχεια έγινε λήψη της επεξεργασμένης μορφής της βιβλιοθήκης tess-two που προορίζεται για χρήση σε συστήματα Android μέσω του GitHub. Για να εφαρμόσουμε την τεχνική αναγνώρισης χαρακτήρων έπρεπε να γίνει επιλογή γλώσσας. Η επιλογή αυτή γίνεται από μία τεράστια λίστα διάφορων γλωσσικών στοιχείων με μορφή αλγορίθμου με όνομα Language trained data. Με την ταχύτερη ανάπτυξη του Tesseract τα τελευταία χρόνια, η λίστα αυτή αποτελείται από χιλιάδες γνωστές γλώσσες. Είναι άξιο αναφοράς ότι υπάρχει η δυνατότητα στους χρήστες να εγκαταστήσουν το Language trained data ακόμα και για αναγνώριση σε αρχαία ελληνικά. Για το παρόν project επιλέχθηκε η

eng.traineddata v3.02 για αναγνώριση λατινικών αλφαριθμητικών χαρακτήρων και συμβόλων.

Έχοντας εγκατεστημένο λοιπόν το Android NDK και αφού επιλέξαμε τη γλώσσα της βιβλιοθήκης, έπρεπε να γίνουν μερικές αλλαγές στις μεταβλητές περιβάλλοντος των Windows (environment variables) ώστε να κάνουμε compile τη βιβλιοθήκη μέσω του NDK. Έπρεπε λοιπόν να προστεθεί στο σύστημα το directory με μορφή "C:\android\_ndk\_r10c". Με την ολοκλήρωση αυτής της αλλαγής ήταν αναγκαίο να γίνει επανεκκίνηση του υπολογιστή ώστε να αποθηκευτούν οι αλλαγές και να προχωρήσουμε στις υπόλοιπες ενέργειες.

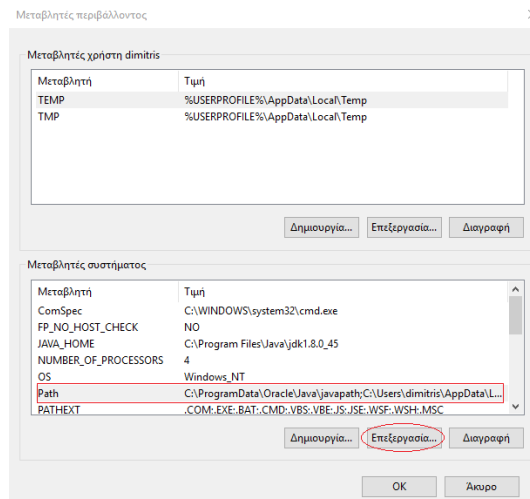
Παρακάτω παρουσιάζεται λεπτομερώς η διαδικασία σε 3 βήματα με τη βοήθεια εικόνων.

1. Εντοπισμός μεταβλητών περιβάλλοντος από τις ιδιότητες του συστήματος. (Εικόνα 10)



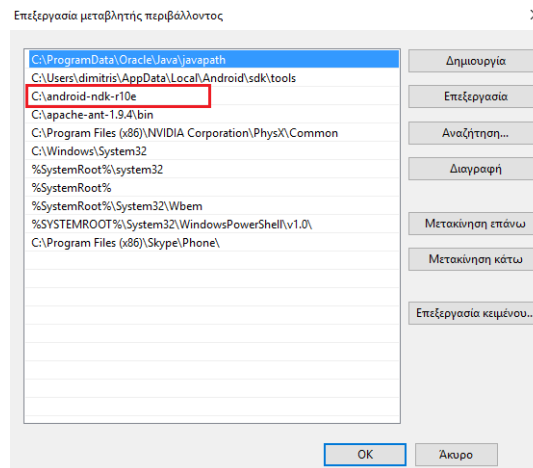
**Εικόνα 10:** Εντοπισμός μεταβλητών συστήματος

2. Επεξεργασία μεταβλητής Path για την εισαγωγή του NDK ώστε να χρησιμοποιηθεί ως compiler. (Εικόνα 11)



**Εικόνα 11:** Εισαγωγή του NDK στο Path του συστήματος

3. Τελική τροποποίηση και αποθήκευση αλλαγών στο σύστημα. (Εικόνα 12)

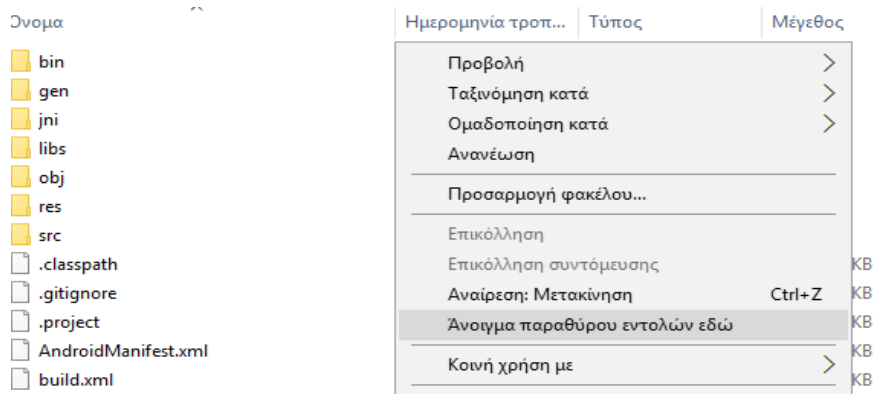


**Εικόνα 12:** Εισαγωγή μονοπατιού του NDK

Αξίζει να σημειωθεί ότι η παρουσίαση των βημάτων με λεπτομέρειες μέσω εικόνων κρίθηκε αναγκαία, διότι οι αλλαγές πρέπει να γίνουν με ακρίβεια για την σωστή μεταγλώττιση και επεξεργασία της βιβλιοθήκης tess-two, αλλά και για να μην προκληθεί οποιαδήποτε δυσλειτουργία στο λειτουργικό σύστημα του υπολογιστή.

Αφού ολοκληρώθηκαν τα βήματα αυτά χωρίς λάθη ή σφάλματα, υπήρχε πλέον τη δυνατότητα να χρησιμοποιήσουμε το NDK μέσω της γραμμής εντολών. Μέσα στο φάκελο του tess-two πραγματοποιήθηκε άνοιγμα παραθύρου γραμμής εντολών. Για να εμφανιστεί η επιλογή, ο χρήστης πρέπει

να κάνει δεξί κλικ μέσα στο φάκελο, έχοντας κρατημένο το πλήκτρο Shift όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 13).



**Εικόνα 13:** Άνοιγμα βιβλιοθήκης στη γραμμή εντολών

Στο παράθυρο γραμμής εντολών, εισάγουμε την εντολή “ndk-build” ώστε να γίνουν οι απαραίτητες τροποποιήσεις της βιβλιοθήκης. Η διαδικασία αυτή δεν θα πρέπει σε καμία περίπτωση να διακοπεί διότι μπορεί να προκαλέσει ζημιά στον πηγαίο κώδικα, οπότε μετά θα πρέπει να πραγματοποιηθεί εκ νέου λήψη της βιβλιοθήκης, και να πραγματοποιηθούν τα βήματα από την αρχή. Παρακάτω εμφανίζεται μέρος του compile (Εικόνα 14).

```

C:\WINDOWS\system32\cmd.exe
C:\Users\dimitris\Desktop\tess\tess-two>ndk-build
[armeabi-v7a] SharedLibrary : libhydrogen.so
[armeabi-v7a] Install      : libhydrogen.so => libs/armeabi-v7a/libhydrogen.so
[armeabi-v7a] SharedLibrary : libimageutils.so
[armeabi-v7a] Install      : libimageutils.so => libs/armeabi-v7a/libimageutils.so
[armeabi-v7a] SharedLibrary : libopticalflow.so
[armeabi-v7a] Install      : libopticalflow.so => libs/armeabi-v7a/libopticalflow.so
[x86] SharedLibrary : libhydrogen.so
[x86] Install      : libhydrogen.so => libs/x86/libhydrogen.so
[x86] SharedLibrary : libimageutils.so
[x86] Install      : libimageutils.so => libs/x86/libimageutils.so
[x86] SharedLibrary : libopticalflow.so
[x86] Install      : libopticalflow.so => libs/x86/libopticalflow.so
[mips] SharedLibrary : libhydrogen.so
[mips] Install      : libhydrogen.so => libs/mips/libhydrogen.so
[mips] SharedLibrary : libimageutils.so
[mips] Install      : libimageutils.so => libs/mips/libimageutils.so
[mips] SharedLibrary : libopticalflow.so
[mips] Install      : libopticalflow.so => libs/mips/libopticalflow.so

```

**Εικόνα 14:** Τροποποίηση βιβλιοθήκης μέσω NDK

Αναφορικά να σημειωθεί ότι, η μεταγλώττιση (compile) της βιβλιοθήκης μπορεί να διαρκέσει έως και 30 λεπτά, ανάλογα με την υπολογιστική δύναμη του συστήματος.

Με την ολοκλήρωση της μεταγλώττισης, εισάγουμε την εντολή “android update project –path (το μονοπάτι στο οποίο αποθηκεύσαμε το φάκελο tess-two)

-target -XX” , όπου XX προσθέτουμε το target SDK που θα χρησιμοποιήσουμε στην ανάπτυξη της εφαρμογής μας. Στην παρούσα εφαρμογή το minSdkVersion είναι 15 και το targetSdkVersion είναι 23.

Μόλις η εντολή εκτελεστεί επιτυχώς, το τελευταίο βήμα στη γραμμή εντολών είναι να εκτελέσουμε την εντολή “ant release”. Με αυτήν η βιβλιοθήκη μας θα μπορεί πλέον να επεξεργαστεί μέσα σε μία εφαρμογή γλώσσας JAVA, όπως είναι και οι εφαρμογές σε Android.

Να σημειωθεί πως στην περίπτωση που η εντολή “ant release” παρουσιάζει σφάλμα εύρεσης, θα πρέπει να εισάγουμε μία καινούρια μεταβλητή περιβάλλοντος με όνομα “JAVA\_HOME” με μορφή “C:\Program Files\Java\jdk1.8.0\_45” (όπως φαίνεται στην εικόνα 13) η οποία να υποδεικνύει το μονοπάτι για το JDK (Java-Development-Kit) στον υπολογιστή

### 2.7.2 Ανάπτυξη βιβλιοθήκης tess-two στο Android Studio

Με την ολοκλήρωση όλων των προαπαιτούμενων ενεργειών, η υλοποίηση της εφαρμογής μεταφέρθηκε πλήρως στο προγραμματιστικό περιβάλλον του Android Studio. Η υλοποίηση ξεκίνησε με τον ορισμό της έτοιμης πλέον βιβλιοθήκης tess-two, ως βασική βιβλιοθήκη του project. Αρχικά στο κενό project που υπήρχε, δημιουργήθηκε ένας καινούριος κατάλογος (directory) με όνομα “libraries”. Σε αυτόν προσθέσαμε έναν υποκατάλογο με όνομα “tess-two”. Στη συνέχεια μεταφέραμε το περιεχόμενο του φακέλου tess-two που επεξεργάστηκε, μέσα στον αντίστοιχο υποκατάλογο που φτιάξαμε. Εκεί δημιουργήθηκε ένα νέο αρχείο build.gradle με τις απαραίτητες λεπτομέρειες. Στην παρακάτω εικόνα (Εικόνα 15) φαίνονται οι βασικές ρυθμίσεις του αρχείου gradle.

```

}buildscript {
}  repositories {
    |   mavenCentral()
}  }
}  dependencies {
    |   classpath 'com.android.tools.build:gradle:1.3.1'
}  }
}

apply plugin: 'com.android.library'

}android {
    compileSdkVersion 23
    buildToolsVersion "23.0.1"

}  defaultConfig {
    minSdkVersion 15
    |   targetSdkVersion 23|
}  }

}  sourceSets.main {
    manifest.srcFile 'AndroidManifest.xml'
    java.srcDirs = ['src']
    resources.srcDirs = ['src']
    res.srcDirs = ['res']
    |   jniLibs.srcDirs = ['libs']
}  }
}
}

```

**Εικόνα 15:** Το αρχείο build.gradle του tess-two

Με την ολοκλήρωση αυτού του βήματος, επόμενη κίνηση είναι η δήλωση του tess-two μέσα στο settings.gradle ολόκληρου του project ως καταχωρημένης βιβλιοθήκης, για να μπορεί να κληθεί όποτε αυτό είναι αναγκαίο. Ο απαραίτητος κώδικας παρουσιάζεται στην επόμενη εικόνα (Εικόνα 16).

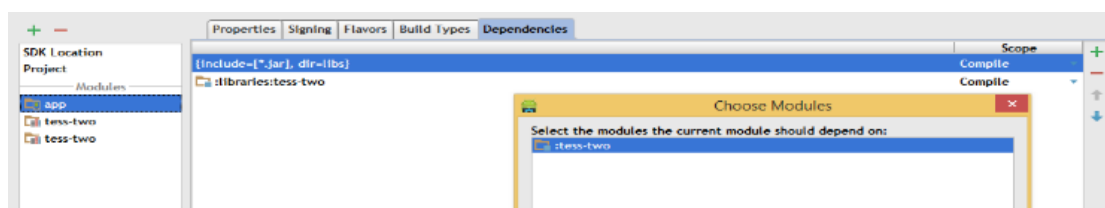
```

include ':app'
include ':libraries:tess-two'|

```

**Εικόνα 16:** Δήλωση βιβλιοθήκης του project

Η τελευταία ενέργεια ώστε να είναι λειτουργική η βιβλιοθήκη μας, ήταν να δηλωθεί στα modules του project και να καταχωρηθεί ως “is Library”, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 17).



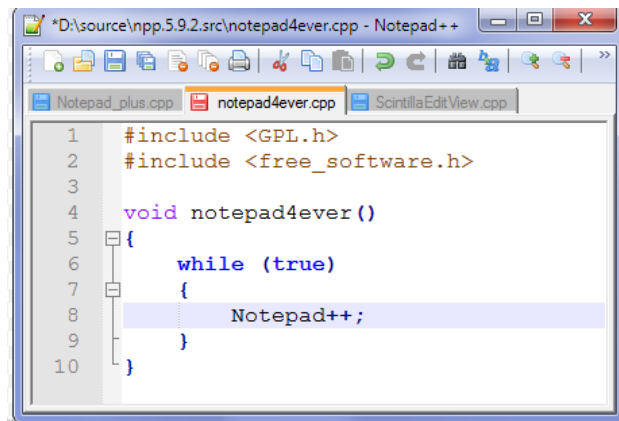
**Εικόνα 17:** Καταχώρηση βιβλιοθήκης στο project



Έχοντας πλέον τη βιβλιοθήκη tess-two έτοιμη και λειτουργική μέσα στο project μας, προχωρήσαμε στην υλοποίηση μιας συνολικά λειτουργικής και εύκολης στη χρήση εφαρμογής, η οποία θα χρησιμοποιεί τη βάση δεδομένων που αναπτύχθηκε, για την εύρεση χρηστών, την καταγραφή και αποστολή μετρήσεων με οπτική αναγνώριση και τη χρήση των δεδομένων αυτών ανάλογα με το επίπεδο του χρήστη που τη χρησιμοποιεί.

## 2.8 Notepad ++

Το Notepad ++ [15] είναι ένα πρόγραμμα επεξεργασίας κειμένου και πηγαίου κώδικα για λειτουργικό σύστημα Windows. Σε αντίθεση με το συνηθισμένο σημειωματάριο, το ενσωματωμένο πρόγραμμα επεξεργασίας κειμένου υποστηρίζει πολλαπλές καρτέλες επεξεργασίας. Το Notepad ++ χρησιμοποιήθηκε κατά κόρων στην παρούσα διπλωματική εργασία σε όλα τα αρχεία php, css, javascript, html, MySQL, Ajax και αποτελεί πρόγραμμα ανοιχτού λογισμικού και διανέμεται δωρεάν. Παρακάτω εμφανίζεται αναφορικά το περιβάλλον του (Εικόνα 18).



Εικόνα 18 : Το Notepad++

## Σύνοψη κεφαλαίου

Σε αυτό το κεφάλαιο αναφερθήκαμε στις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος, που έχει ως κύριους πυλώνες του την ιστοσελίδα και την εφαρμογή Android. Έγινε γενική αναφορά

στα χαρακτηριστικά και τις δυνατότητες των τεχνολογιών αυτών καθώς και στα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για να φέρουν εις πέρας την ανάπτυξη του συστήματος.

## Κεφάλαιο 3

### Περιγραφή και απαιτήσεις συστήματος

Στο κεφάλαιο αυτό θα περιγραφούν τα μέρη του συστήματος που υλοποιήθηκε στην παρούσα διπλωματική εργασία και θα οριστούν οι απαιτήσεις του συστήματος. Επίσης, θα γίνει αναφορά στους ρόλους του συστήματος αλλά και τις απαιτήσεις που προκύπτουν από αυτούς. Τέλος θα γίνει περιγραφή της αρχιτεκτονικής του συστήματος συνολικά, δίνοντας μια ξεκάθαρη εικόνα για το σύστημα που αναπτύχθηκε.

#### 3.1 Περιγραφή συστήματος

Το σύστημα που αναπτύχθηκε αποτελείται από 2 βασικά μέρη. Την εφαρμογή Android και την αντίστοιχη ιστοσελίδα με την οποία συνεργάζεται. Για την υλοποίηση της εφαρμογής Android, χρησιμοποιήθηκαν διάφορα εργαλεία όπως αναφέρθηκαν στο προηγούμενο κεφάλαιο. Για την επίτευξη των βασικών απαιτήσεων της εφαρμογής δημιουργήθηκε ένας μεγάλος κατάλογος αρχείων τα οποία καλύπτουν αυτές τις απαιτήσεις, καθώς ακόμα και εξωτερικές προσθήκες και βιβλιοθήκες.

Αντίστοιχα για την δημιουργία της ιστοσελίδας, χρησιμοποιήθηκε πρότυπο ιστοσελίδας μιας εταιρίας, προσφέροντας όλες τις απαραίτητες δυνατότητες σε οποιονδήποτε χρήστη που την επισκέπτεται, εγγεγραμμένου και μη, με κυριότερες :

- Την σωστή και προσεγμένη παρουσίαση του συστήματος που υλοποιήθηκε
- Την επεξήγηση των βασικών του σημείων
- Την περιγραφή των δυνατοτήτων του
- Τη δημιουργία λογαριασμού

Ο σύνδεσμος των 2 αυτών στοιχείων του συστήματος, είναι μία βάση δεδομένων η οποία σχεδιάστηκε με σκοπό να λειτουργεί αποθηκεύοντας τα στοιχεία των χρηστών της εφαρμογής Android και της ιστοσελίδας, καθώς και τα δεδομένα σε κάθε επίπεδο χρήστη που υπάρχει στο σύστημα.

Το σύστημα που υλοποιήθηκε συμπεριλαμβάνει 3 βασικότετους ρόλους οι οποίοι είναι :

- Διαχειριστής / admin
- Τεχνικός / tech
- Εγγεγραμμένος Χρήστης / user

Δυνατότητες των ρόλων του συστήματος μπορούν να περιγραφούν συνοπτικά ως εξής :

#### **Δυνατότητες εγγεγραμμένου χρήστη**

- Ο χρήστης επισκέπτεται την ιστοσελίδα και δημιουργεί τον προσωπικό του λογαριασμό.
- Εναλλακτικά εγκαθιστά την εφαρμογή και δημιουργεί τον προσωπικό του λογαριασμό μέσω αυτής.
- Ο χρήστης καταχωρείται αυτομάτως ως «εγγεγραμμένος χρήστης», εκτός εάν υπάρχει κάποια άλλη ένδειξη οπότε και αντιμετωπίζεται από τον διαχειριστή του συστήματος.
- Ο χρήστης μπορεί να επικοινωνήσει με τον διαχειριστή του συστήματος για οποιοδήποτε σχετικό θέμα.
- Ο χρήστης, αφού εισέλθει στον προσωπικό του λογαριασμό μπορεί μέσω της εφαρμογής στην συσκευή του να φωτογραφίσει και να αποστείλει τα δεδομένα που αναγνωρίστηκαν στη βάση.
- Εναλλακτικά μπορεί να εισάγει τα δεδομένα του μέσω της ιστοσελίδας.
- Ο χρήστης μπορεί οποιαδήποτε στιγμή να ανατρέξει και να εποπτεύσει τα προσωπικά του δεδομένα.

### Δυνατότητες τεχνικού

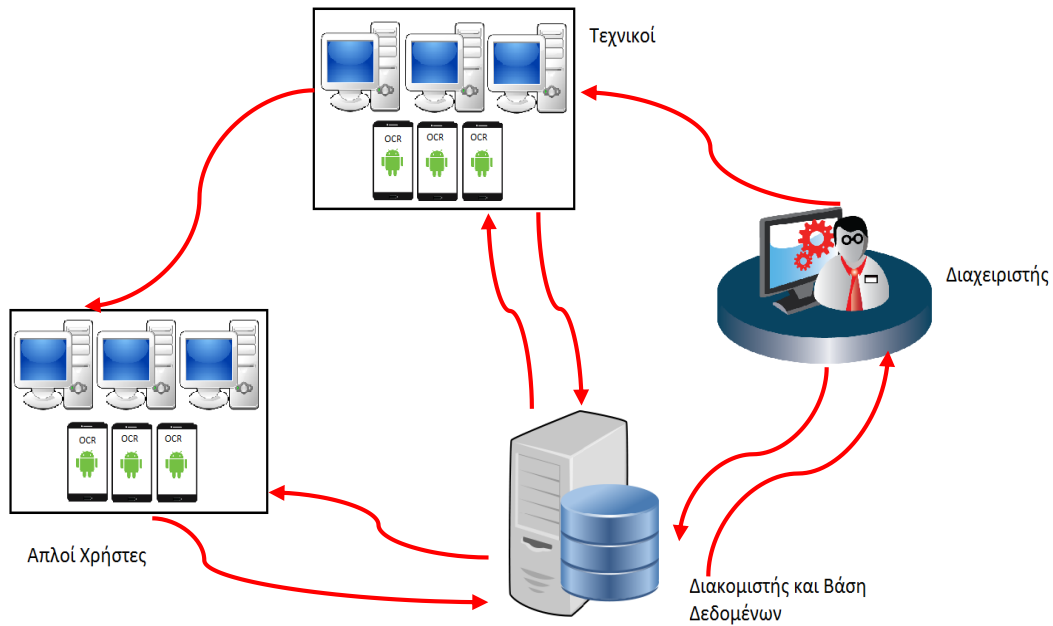
- Ο τεχνικός επισκέπτεται την ιστοσελίδα και δημιουργεί τον προσωπικό του λογαριασμό.
- Εγκαθιστά την εφαρμογή στην συσκευή του.
- Ο τεχνικός αποστέλλει σχετικό αίτημα στον διαχειριστή και αφού ελεγχθεί, ορίζεται ως «τεχνικός» μέσα στο σύστημα, πράγμα που του παρέχει επιπλέον δυνατότητες και αρμοδιότητες.
- Ο τεχνικός μπορεί να επικοινωνήσει με τον διαχειριστή του συστήματος για οποιοδήποτε σχετικό θέμα.
- Ο τεχνικός αναλαμβάνει να κάνει τις απαραίτητες μετρήσεις, όπως αυτές θα του ανατεθούν από το διαχειριστή.
- Ο τεχνικός μπορεί οποιαδήποτε στιγμή να ανατρέξει και να εποπτεύσει τα προσωπικά του δεδομένα αλλά και τα δεδομένα των χρηστών που του έχουν ανατεθεί.

### Δυνατότητες διαχειριστή

Ο διαχειριστής αποτελεί την οντότητα με τις περισσότερες δυνατότητες μέσα στο σύστημα. Μπορεί ανά πάσα στιγμή να προσθέσει, να αφαιρέσει ή να τροποποίηση χρήστες οποιουδήποτε επιπέδου. Ταυτόχρονα μπορεί να ανατρέξει και να εποπτεύσει τα δεδομένα όλων των εγγεγραμμένων χρηστών οποιουδήποτε επιπέδου. Είναι η οντότητα που διαχειρίζεται τα προβλήματα μέσω της φόρμας επικοινωνίας και αναθέτει έργο όπου αυτό είναι απαραίτητο. Η σωστή λειτουργία του συστήματος ελέγχεται απόλυτα από το διαχειριστή.

## **3.2 Αρχιτεκτονική Συστήματος**

Το μοντέλο της αρχιτεκτονικής του συστήματος απεικονίζεται παρακάτω (Εικόνα 19).



**Εικόνα 19:** Αρχιτεκτονική συστήματος

Στο μοντέλο αυτό εμφανίζεται ο διαχειριστής του συστήματος ως υπερχρήστης. Έχει δυνατότητες επεξεργασίας και διαγραφής δεδομένων και χρηστών οποιουδήποτε επιπέδου, όταν αυτό κρίνεται αναγκαίο. Το επίπεδο χρήστη των τεχνικών που είναι αρμόδιο για την σωστή καταγραφή και αποθήκευση μετρήσεων και τέλος το επίπεδο των απλών χρηστών, των οποίων η δυνατότητα περιορίζεται στο να πραγματοποιούν μετρήσεις και να τις αποστέλλουν και να παρακολουθούν τυχόν αλλαγές μέσω του προσωπικού τους λογαριασμού.

Όπως γίνεται προφανές, όλα τα επίπεδα χρηστών μπορούν να διαβάζουν και να αποστέλλουν δεδομένα με το διακομιστή και τη βάση δεδομένων χωρίς περιορισμούς. Για τις επιπλέον λειτουργίες επεξεργασίας και διαγραφών οι δυνατότητες αλλάζουν όσο ανεβαίνουμε επίπεδο χρήστη.

### 3.3 Απαιτήσεις συστήματος

Με την ανάλυση των δυνατοτήτων των ρόλων που περιγράψαμε, είναι ξεκάθαρο ότι υπάρχουν κάποιες απαιτήσεις από το σύστημα που

δημιουργήσαμε, ώστε να εξυπηρετεί πλήρως και με ορθό τρόπο τις ζητούμενες ενέργειες. Οι απαιτήσεις αυτές είναι ένα πολύ σημαντικό κομμάτι καθώς το σύστημα που δημιουργήσαμε φέρει την τεράστια ευθύνη της ασφαλούς καταχώρησης δεδομένων και ταυτόχρονα της άμεσης προσπέλασης και επεξεργασίας τους.

Μία από τις βασικότερες απαιτήσεις είναι η πολλαπλή αποθήκευση καταχωρήσεων. Καθώς το σύστημα αντιμετωπίζει χρήστες στον τομέα των κοινωφελών μετρήσεων, θα πρέπει να δίνει την δυνατότητα στο χρήστη να μπορεί να εισάγει κάθε φορά καινούρια δεδομένα χωρίς να χρειάζεται να αντικαταστήσει τα ήδη υπάρχοντα. Ταυτόχρονα όμως θα πρέπει να δίνει τη δυνατότητα στο χρήστη να ενημερώνει κάποια υπάρχουσα καταγραφή, ή να μπορεί να διορθώσει μία εσφαλμένη.

Συνεχίζοντας, υπάρχει η απαίτηση από το σύστημα να καταγράφει αναλυτικά πότε έγινε κάποια καταμέτρηση (ημερομηνία και ώρα) ώστε να διευκολύνει τον χρήστη οποιουδήποτε επιπέδου, στην περίπτωση που ψάχνει κάτι συγκεκριμένο.

Το σύστημα θα πρέπει να ενημερώνει το χρήστη τη στιγμή που γίνεται κάποια καταμέτρηση εάν αυτή καταχωρήθηκε επιτυχώς ή όχι. Στην ίδια λογική, μόλις μία καταγραφή αποθηκευτεί επιτυχώς, θα πρέπει να δίνεται η δυνατότητα στο χρήστη να ανατρέξει τα προσωπικά του δεδομένα και να την δει. Εδώ δημιουργείται άμεσα μία ακόμα απαίτηση από το σύστημα το οποίο θα πρέπει να έχει ένα φιλικό και ευανάγνωστο γραφικό περιβάλλον σχεδιασμένο για το κάθε επίπεδο χρήστη ξεχωριστά. Το περιβάλλον αυτό θα πρέπει να προσαρμόζεται αυτομάτως στις ανάγκες του χρήστη (π.χ. αυτόματη μορφοποίηση της εικόνας ανάλογα με τη συσκευή από την οποία βρίσκεται συνδεδεμένος ο χρήστης).

Ακόμα, μιας και η εξοικείωση κάθε χρήστη με τη χρήση ιστοσελίδων και εφαρμογών σε αντίστοιχες συσκευές διαφέρει κατά πολύ, το σύστημά μας θα πρέπει να διευκολύνει τους χρήστες του παρέχοντας τις πιο συνηθισμένες και απαραίτητες ενέργειες με πολύ απλές διαδικασίες (π.χ. αναζήτηση ή άθροισμα μετρήσεων με το πάτημα ενός κουμπιού μέσα στο γραφικό του περιβάλλον).

Όπως είναι φυσικό, οι απαιτήσεις θα πρέπει να είναι δεδομένες για όλα τα επίπεδα χρηστών. Καθώς όμως προχωρούμε προς τα πάνω στα επίπεδα των χρηστών ανακαλύπτουμε μερικές ακόμα.

Είναι απαραίτητη προδιαγραφή από το σύστημα να δίνει ασφαλή και άμεση πρόσβαση στους χρήστες επιπέδου «τεχνικός», σε δεδομένα χρηστών για τα οποία είναι υπεύθυνοι. Ταυτόχρονα όταν πραγματοποιείται κάποια καινούρια καταμέτρηση ή ενημέρωση καταμέτρησης από κάποιο χρήστη, ο τεχνικός που του αντιστοιχεί θα πρέπει άμεσα να ενημερώνεται και το αντίστροφο.

Ακόμα, ο διαχειριστής του συστήματος θα πρέπει να έχει τη δυνατότητα να πραγματοποιεί μία τεράστια γκάμα ενεργειών. Αποτελεί τον υπερχρήστη του συστήματος καθώς μπορεί να κάνει προσθήκες, τροποποιήσεις και διαγραφές των χρηστών που υπάρχουν. Επίσης θα μπορεί να παρακολουθεί όλες τις καταχωρήσεις μετρήσεων σε ανάλογους πίνακες (ή γραφήματα όπου είναι εφικτό), σε γραφικό περιβάλλον σχεδιασμένο για αυτόν, και να αναθέτει ενέργειες στους τεχνικούς όταν αυτό κρίνεται αναγκαίο.

## **Σύνοψη κεφαλαίου**

Στο κεφάλαιο αυτό έγινε παρουσίαση και περιγραφή των απαιτήσεων του συστήματος στη γενική του μορφή αλλά και στα επιμέρους στοιχεία του. Επίσης, διευκρινίστηκαν οι ρόλοι του συστήματος και έγινε περιγραφή της αρχιτεκτονικής του συστήματος. Αυτές οι πληροφορίες είναι πολύ σημαντικές για την κατανόηση και επεξήγηση του επόμενου κεφαλαίου.



## Κεφάλαιο 4

### Υλοποίηση λογισμικού μέρους συστήματος

Σκοπός αυτού του κεφαλαίου είναι η παρουσίαση και ανάλυση του λογισμικού μέρους του συστήματος που αναπτύχθηκε. Το λογισμικό μέρος αποτελεί το κυριότερο κομμάτι αυτής της διπλωματικής εργασίας και αποτελείται από 3 βασικούς πυλώνες: Βάση δεδομένων, Εφαρμογή Android και ιστοσελίδα. Έχοντας ήδη περιγράψει το σύστημα και τους ρόλους του, και έχοντας θέσει τις απαιτήσεις του, θα παρουσιαστούν και θα αναλυθούν στην πράξη τα μέρη του συστήματος και πως αυτά συνδέονται μεταξύ τους. Θα περιγραφούν όλες οι λειτουργίες και η ασφάλεια που παρέχουν, μέσω κώδικα. Τέλος, θα γίνει αναφορά σε μετρικά στοιχεία κώδικα αλλά και σε δοκιμές που έγιναν.

#### 4.1 Σχεδιασμός βάσης δεδομένων

Αρχικά θα γίνει παρουσίαση της βάσης δεδομένων. Η βάση δεδομένων που σχεδιάστηκε στα πλαίσια της διπλωματικής αυτής, αποτελεί το συνδετικό κρίκο ανάμεσα στην εφαρμογή Android και την ιστοσελίδα που υλοποιήθηκε. Η βάση αυτή ονομάστηκε APPDB και αποτελείται από 2 πίνακες. Δεδομένου ότι αποτελεί το μέρος στο οποίο είναι αποθηκευμένα όλα τα στοιχεία μας (λογαριασμοί χρηστών και καταγραφές μετρήσεων) σχεδιάστηκε με τρόπο τέτοιο ώστε να παρέχει ασφάλεια διατηρώντας ταυτόχρονα τη δυναμικότητα των στοιχείων που αποθηκεύει.

Για το σχεδιασμό της βάσης δεδομένων διακρίναμε τις απαιτήσεις που υπάρχουν ώστε η τελική υλοποίηση, να εξυπηρετεί πλήρως όλες τις λειτουργίες και τις ροές εκτέλεσης που είναι απαραίτητες για το σύστημα που αναπτύξαμε.

Αρχικά, κατηγοριοποιήθηκαν τα επίπεδα χρηστών που ήταν αναγκαία. Αυτά ήταν ο διαχειριστής, οι τεχνικοί και οι απλοί χρήστες. Τα επίπεδα αυτά

αποτελέσαν ουσιαστικά τις οντότητες της βάσης μας. Οι απαιτήσεις λοιπόν που δημιουργήθηκαν αποτελούσαν ουσιαστικά τις δυνατότητες και τις λειτουργίες των οντοτήτων μας. Συγκεκριμένα διακρίναμε την ανάγκη δημιουργίας πολλαπλών πεδίων για την καταχώρηση των προσωπικών στοιχείων κάθε οντότητας όπως για παράδειγμα, το ονοματεπώνυμο, η διεύθυνση email, το διακριτικό επιπέδου του χρήστη και η διεύθυνση του. Όπως είναι προφανές η διατήρηση αυτών των στοιχείων στη βάση θα έπρεπε να γίνεται με απόλυτη ασφάλεια απέναντι σε κάθε χρήστη. Έτσι λοιπόν, αυτομάτως δημιουργήθηκε η ανάγκη πραγματοποίησης πεδίων που δίνουν στους χρήστες τη δυνατότητα να αποθηκεύουν τα στοιχεία τους σε συνδυασμό με ένα κωδικό. Πηγαίνοντας ένα βήμα παραπέρα σε αυτή τη δυνατότητα, πραγματοποιήθηκαν πεδία για την αποθήκευση των κωδικών σε κρυπτογραφημένη μορφή αλλά και ένα πεδίο στο γινόταν αυτόματη καταχώρηση της κρυπτογράφησης του κωδικού.

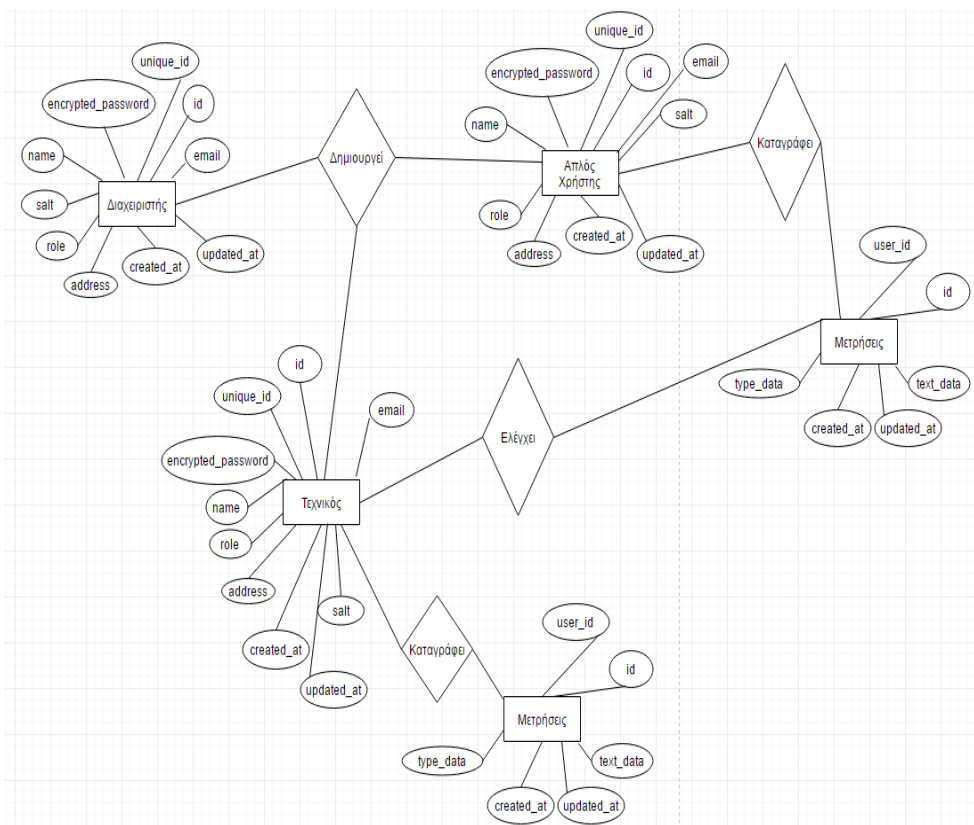
Ακόμα, το σύστημα, ήταν σχεδιασμένο για να προσφέρει τη δυνατότητα ποικίλων μορφών επεξεργασίας στους χρήστες του. Αυτό άμεσα σήμαινε τη δημιουργία πεδίων τα οποία αποθηκεύουν χρονικές «σφραγίδες» από τη δημιουργία και την επεξεργασία των υπόλοιπων πεδίων.

Συνεχίζοντας στην ίδια φιλοσοφία, η βάση δεδομένων που υλοποιήθηκε ήταν υπεύθυνη για την καταχώρηση μετρήσεων στους λογαριασμούς των χρηστών της. Άρα, κρίθηκε αναγκαίο να πραγματοποιηθούν αντίστοιχα πεδία καταχώρησης τιμών τα οποία όμως ταυτόχρονα θα είναι συνδεδεμένα με τον κάθε χρήστη ξεχωριστά. Επιπλέον, δημιουργήθηκαν πεδία στα οποία γινόταν κατηγοριοποίηση ανάλογα με τον τύπο της μέτρησης που καταχωρεί ο εκάστοτε χρήστης. Ακολουθώντας την ίδια λογική με πριν, τα πεδία των μετρήσεων αναπτύχθηκαν θεωρώντας δεδομένη τη δυνατότητα να μπορούν να υποστούν επεξεργασία. Οπότε όπως ήταν φυσικό πραγματοποιήθηκαν αντίστοιχα πεδία για το χρόνο αποθήκευσης και επεξεργασίας οποιασδήποτε μέτρησης. Τέλος, στην περίπτωση διαγραφής στοιχείων του ενός πίνακα, υπήρχαν δεδομένα σε διαφορετικό πίνακα τα οποία θα έπρεπε επίσης να διαγραφούν διότι δεν αντιστοιχούσαν πλέον σε κάποιο χρήστη. Για τη λειτουργία αυτή προστέθηκε η

δυνατότητα διαγραφής αλληλουχιών, μεταξύ στοιχείων των οντοτήτων τα οποία είναι συνδεδεμένα σε διαφορετικούς πίνακες.

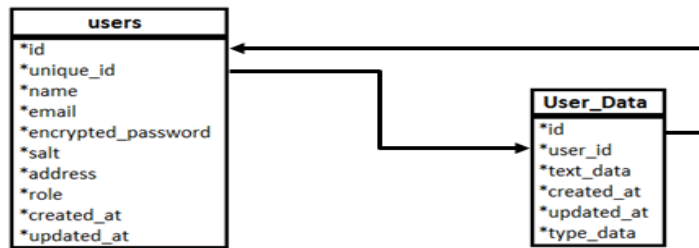
#### 4.1.1 Μοντέλο Οντοτήτων-Συσχετίσεων

Το μοντέλο οντοτήτων-συσχετίσεων [16] (μοντέλο Ο/Σ - ER model) είναι ένα αφαιρετικό ιδεατό μοντέλο δεδομένων, τα οποία έχουν καθορισμένη δομή. Στη μηχανική λογισμικού χρησιμοποιείται για να παρέχει ένα εννοιολογικό σχήμα κατά τη σχεδίαση βάσεων δεδομένων, ως μοντέλο δεδομένων ενός συστήματος και των απαιτήσεών του με top-down προσέγγιση. Ένα διάγραμμα που δημιουργείται με αυτή τη διαδικασία σχεδίασης καλείται διάγραμμα οντοτήτων-συσχετίσεων, ή διάγραμμα Ο/Σ ή ΟΣΔ εν συντομία. Το διάγραμμα οντοτήτων συσχετίσεων που περιγράφει τη βάση δεδομένων που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας, εμφανίζεται στην παρακάτω εικόνα (Εικόνα 20).



Εικόνα 20: Διάγραμμα οντοτήτων-συσχετίσεων της βάσης δεδομένων

Στην επόμενη εικόνα (Εικόνα 21), δίνεται μια γενική παρουσίαση των πινάκων που σχεδιάστηκαν στη βάση δεδομένων.



**Εικόνα 21:** Οι πίνακες της βάσης που αναπτύχθηκε

Στη συνέχεια θα παρουσιαστούν οι πίνακες της βάσης που δημιουργήθηκε και θα εξηγηθούν όλα τα στοιχεία που περιλαμβάνονται σε αυτούς.

#### 4.1.1 Πίνακας users

Στον πίνακα αυτόν αποθηκεύονται τα προσωπικά στοιχεία όλων των χρηστών, καθώς και ο ρόλος τους (Πίνακας 2).

Πεδίο	Τύπος	Κλειδί
id	int(11)	Primary
unique_id	varchar(23)	Unique
name	varchar(50)	
email	varchar(100)	Unique
encrypted_password	varchar(80)	
salt	varchar(10)	
address	varchar(50)	
role	enum('user', 'tech', 'admin')	
created_at	datetime	
updated_at	datetime	

**Πίνακας 2:** Πίνακας users

### Επεξήγηση πίνακα

- **id** : Το id είναι ένας μοναδικός και αύξων αριθμός για κάθε χρήστη. Δεν επηρεάζεται από τον χρήστη και αυξάνεται αυτόματα κάθε φορά που υπάρχει μια νέα εγγραφή στον πίνακα.
- **unique\_id** : Το unique\_id είναι μία τιμή διαφόρων χαρακτήρων που δημιουργείται αυτόματα κάθε φορά που ένας νέος χρήστης δημιουργεί ένα λογαριασμό. Συνδέεται με το λογαριασμό του χρήστη και είναι μοναδικός. Ένα παράδειγμα της μορφής που έχει αυτή η τιμή είναι η 4f074eca601fb8.88015924 .
- **name** : Το name αποτελεί τη μεταβλητή στην οποία καταχωρείται το ονοματεπώνυμο ενός χρήστη. Το συγκεκριμένο πεδίο έχει υποστεί τροποποίηση ώστε να δέχεται και να αντιστοιχίζει χωρίς κανένα πρόβλημα ελληνικούς χαρακτήρες.
- **email** : Το email είναι η μεταβλητή στην οποία καταχωρείται η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη. Έχει όριο τους 100 χαρακτήρες και αποτελεί μοναδικό κλειδί για κάθε χρήστη.
- **encrypted\_password** : Η μεταβλητή αυτή αποθηκεύει τον προσωπικό κωδικό κάθε χρήστη (συνθηματικό). Το συνθηματικό καταχωρείται με τη χρήση μεθόδου base64\_encode [17] . Για την ασφαλέστερη καταχώρηση του χρειαζόμαστε 2 στήλες στη βάση, η μία για το κρυπτογραφημένο συνθηματικό και η άλλη για την κρυπτογράφηση του.
- **salt** : Η μεταβλητή αυτή ελέγχει την κρυπτογράφηση του συνθηματικού και σκοπός της είναι να «αμύνεται» απέναντι στις επιθέσεις κακόβουλων χρηστών.
- **address** : Η μεταβλητή αυτή καταχωρεί την διεύθυνση του εκάστοτε χρήστη.
- **role** : Η μεταβλητή αυτή καταχωρεί με τον τύπο ENUM [18] το επίπεδο χρήστη που δημιουργεί το λογαριασμό. Το προεπιλεγμένο επίπεδο είναι

«απλός χρήστης» (user) και μόνο ο διαχειριστής μπορεί να πραγματοποιήσει αλλαγές όταν αυτό είναι επιθυμητό ή αναγκαίο.

- **created\_at** : Σε αυτήν τη μεταβλητή καταχωρείται η ακριβής ημερομηνία και ώρα στην οποία δημιουργήθηκε ο κάθε λογαριασμός.
- **updated\_at** : Σε αυτήν τη μεταβλητή καταχωρείται η ακριβής ημερομηνία και ώρα σε περίπτωση που ένας υπάρχων λογαριασμός υπέστη τροποποιήσεις.

#### 4.1.2 Πίνακας User\_Data

Στον πίνακα αυτό αποθηκεύονται τα δεδομένα που καταχωρεί ο χρήστης είτε μέσω οπτικής αναγνώρισης από την εφαρμογή, είτε χειροκίνητα μέσω της ιστοσελίδας (Πίνακας 3).

Πεδίο	Τύπος	Κλειδί
id	int(11)	Primary
user_id	varchar(23)	Foreign Key
text_data	varchar(600)	
created_at	datetime	
updated_at	datetime	
type_data	enum('Electricity', 'Water', 'Natural Gas', 'Extra')	

**Πίνακας 3:** Πίνακας User\_Data

#### Επεξήγηση πίνακα

- **id** : Το id είναι ένας μοναδικός και αύξων αριθμός για κάθε χρήστη. Δεν επηρεάζεται από τον χρήστη και αυξάνεται αυτόματα κάθε φορά που υπάρχει μια νέα εγγραφή στον πίνακα.
- **user\_id** : Το user\_id αποτελεί τη μεταβλητή του πίνακα User\_Data που είναι εξωτερικό κλειδί του πίνακα users και συνδέεται με τη μεταβλητή

unique\_id, ώστε να καταχωρούνται τα δεδομένα ξεχωριστά στο id του κάθε χρήστη.

- **text\_data** : Στη μεταβλητή text\_data αποθηκεύονται οι μετρήσεις του χρήστη από την εφαρμογή Android ή την ιστοσελίδα.
- **created\_at** : Σε αυτήν τη μεταβλητή αποθηκεύεται η ακριβής ημερομηνία και ώρα στην οποία καταχωρήθηκε μία μέτρηση.
- **updated\_at** : Σε αυτήν τη μεταβλητή αποθηκεύεται η ακριβής ημερομηνία και ώρα σε περίπτωση που μία υπάρχουσα καταγραφή, υπέστη τροποποιήσεις.
- **type\_data** : Σε αυτήν τη μεταβλητή αποθηκεύεται το είδος της μέτρησης που πραγματοποιήθηκε. Υπάρχουν 4 επιλογές καταχώρησης, μία για ηλεκτρικό ρεύμα, μία για μέτρηση νερού, μία για μέτρηση φυσικού αερίου και μία επιπλέον για να προσφέρει ευελιξία στο χρήστη.

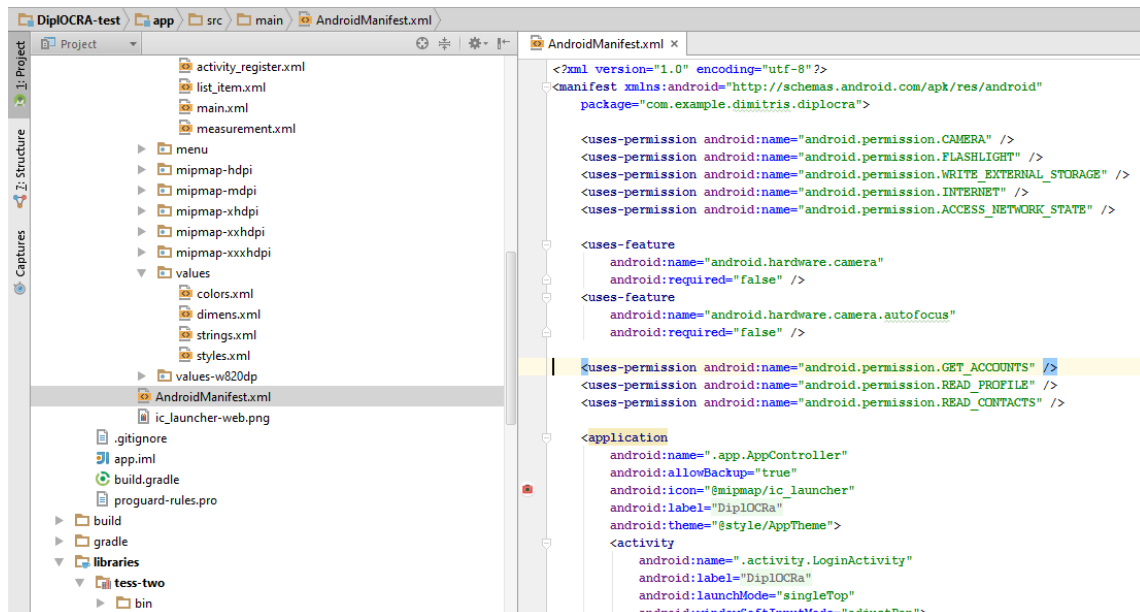
## 4.2 Ανάπτυξη εφαρμογής στο Android Studio

Έχοντας θέσει τα δεδομένα που θα πραγματευτούμε με την επεξήγηση της βάσης δεδομένων, προχωρούμε στην ανάπτυξη του λογισμικού της εφαρμογής Android. Για την δημιουργία της εφαρμογής είναι απαραίτητο να γίνει αναφορά στις βασικές έννοιες στις οποίες στηρίζεται η δομή της εφαρμογής (activity, manifest, layouts, values, strings κλπ) καθώς και στα αρχεία από τα οποία αποτελείται και τον τρόπο που αυτά συνδέονται.

Ας αρχίσουμε την προσέγγιση από το AndroidManifest. Το AndroidManifest.xml, θα πρέπει να αναφέρουμε ότι, είναι ένα αρχείο που βρίσκεται στον κατάλογο main. Αυτό το XML αρχείο ενημερώνει το σύστημα για όλα τα απαιτούμενα που έχει μια εφαρμογή Android. Η ύπαρξή του αποτελεί απαραίτητη προϋπόθεση για την κατασκευή και την σωστή λειτουργία της εφαρμογής.

Στην εικόνα που ακολουθεί (Εικόνα 22) παρουσιάζονται τα στοιχεία του AndroidManifest.xml της παρούσας διπλωματικής εργασίας. Στο αρχείο αυτό δηλώνονται υποχρεωτικά όλες οι δραστηριότητες (activities) της εφαρμογής καθώς επίσης σημειώνονται με την εντολή “uses-permission” και όλες οι

λειτουργίες που απαιτεί η εφαρμογή από το την συσκευή στην οποία είναι εγκατεστημένη. Συγκεκριμένα, στην εφαρμογή που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής ζητείται άδεια χρήσης της κάμερας (android.permission.CAMERA), του φλας (android.permission.FLASHLIGHT), της χρήσης εξωτερικής μνήμης (android.permission.WRITE\_EXTERNAL\_STORAGE αν υπάρχει) και η χρήση σύνδεσης στο διαδίκτυο (android.permission.INTERNET / ACCESS\_NETWORK\_STATE).



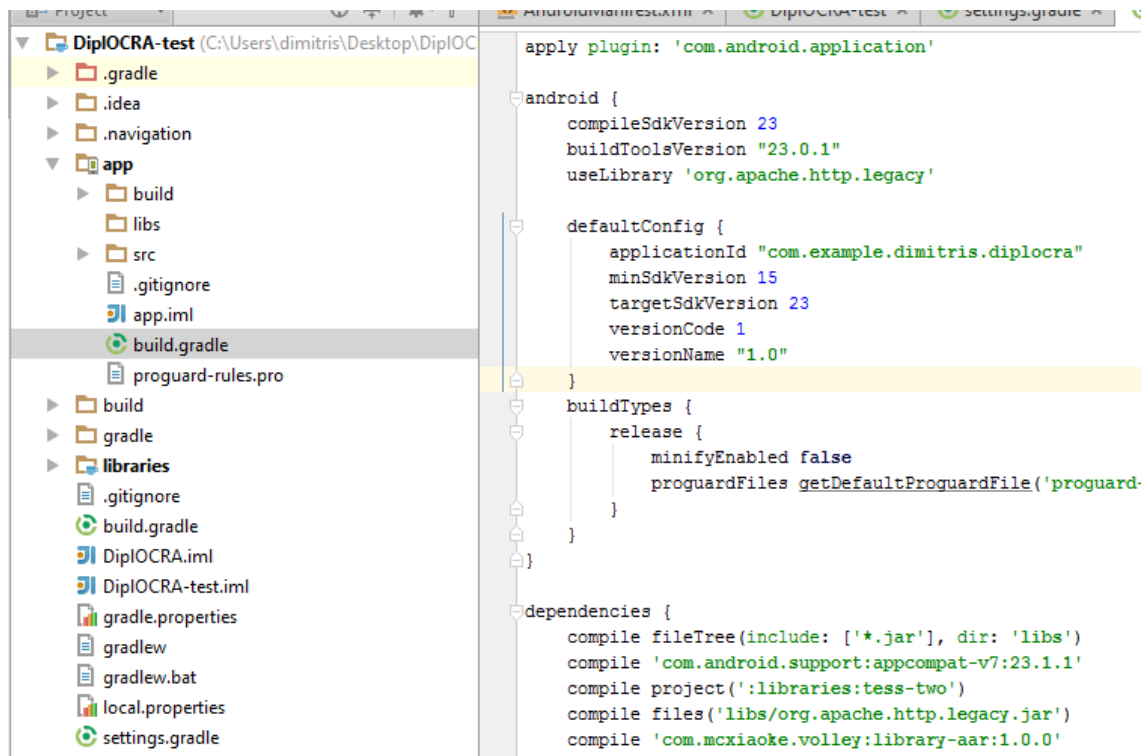
Εικόνα 22: Το AndroidManifest.xml της εφαρμογής

Όπως έχει αναφερθεί και προηγουμένως, ήταν απαραίτητη προϋπόθεση να δηλωθούν όλα αυτά τα στοιχεία, για τη σωστή λειτουργία της εφαρμογής.

Ένα ακόμα πολύ σημαντικό κομμάτι της εφαρμογής, είναι ο κατάλογος Gradle. Το Gradle είναι ένα σύστημα το οποίο εισήχθη σαν έννοια στον προγραμματισμό εφαρμογών Android μέσω του Android Studio. Αποτελεί ένα build system που χρησιμοποιείται για την κατασκευή και μεταγλώττιση ενός project, σε ένα APK (Android application package), το οποίο με τη σειρά του μπορεί να εγκατασταθεί σε μία συσκευή. Βρίσκεται στον αρχικό κατάλογο του project και μέσα σε αυτό ορίζονται επιλογές διαμόρφωσης, όπως π.χ. το κατώτερο επίπεδο στο οποίο θα είναι εκτελέσιμη η εφαρμογή. Επιπλέον ορίζονται όλες οι εξαρτήσεις (dependencies) του project ώστε να παρέχεται υποστήριξη και συμβατότητα.



Τα χαρακτηριστικά του gradle της εφαρμογής Android που αναπτύχθηκε, παρουσιάζονται στη εικόνα που ακολουθεί (Εικόνα 23). Αναφορικά φαίνονται κατά σειρά, η έκδοση της εφαρμογής και η χρήση βιβλιοθηκών, και ακολουθούν η έκδοση στην οποία στοχεύει το σύστημα, μαζί με την κατώτερη έκδοση που μπορεί να το υποστηρίξει. Τέλος, εμφανίζονται όλα τα dependencies με την περιγραφή της μεταγλώττισής τους.



Εικόνα 23: Το gradle του project

Όπως φαίνεται και στην εικόνα, υπήρχαν περιπτώσεις βιβλιοθηκών οι οποίες για να λειτουργήσουν και να μεταγλωττιστούν ήταν αναγκαία η εισαγωγή τους, μέσω εξωτερικών αρχείων [π.χ. compile files. ('libs/org.apache/http/legacy.jar')].

#### 4.2.1 Λειτουργίες login και register εφαρμογής

Στη συνέχεια θα αναφέρουμε τα οπτικά και λειτουργικά σημεία της εφαρμογής με τα οποία ουσιαστικά αλληλεπιδρά ο χρήστης, αναλύοντας τα, μέσα από την εφαρμογή την οποία αναπτύξαμε.

Αρχικά, την πρώτη φορά που κάποιος χρήστης εγκαθιστά και τρέχει την εφαρμογή, του εμφανίζεται το `activity_login.xml` και όπως υποδηλώνει το όνομα του είναι το σημείο στο οποίο δίνεται η προτροπή στον χρήστη να εισέλθει στο λογαριασμό του, εάν φυσικά έχει. Συμπληρώνοντας σωστά τα στοιχεία του, ο χρήστης στη συνέχεια πρέπει να πατήσει το κουμπί «Είσοδος» για να εισέλθει στον προσωπικό του λογαριασμό. Με το πάτημα του κουμπιού αυτού ξεκινά η διαδικασία αποστολής των δεδομένων που συμπλήρωσε ο χρήστης. Τη διαδικασία αυτή, την αναλαμβάνει το `LoginActivity` που είναι το `activity` το οποίο είναι υπεύθυνο για τη λειτουργία του `activity_login`.

Σε αυτό το σημείο πρέπει να αναφερθεί ότι για την άμεση επικοινωνία του Android με τη βάση δεδομένων χρησιμοποιείται επιπλέον κώδικας που συμπεριλαμβάνει στοιχεία PHP και JSON (objects και arrays). Ο τρόπος με τον οποίο πραγματοποιείται η επικοινωνία αυτή, παρουσιάζεται στην επόμενη εικόνα (Εικόνα 24) με χαρακτηριστικό τρόπο.



**Εικόνα 24:** Επικοινωνία Android και βάσης δεδομένων μέσω JSON/PHP

Βάση αυτού του προτύπου λοιπόν, τα δεδομένα στέλνονται από την εφαρμογή μέσω ενός αρχείου PHP το οποίο πραγματοποιεί σύνδεση στη βάση και επεξεργάζεται τα δεδομένα για τον έλεγχο του login. Όταν ολοκληρωθεί ο

έλεγχος τα στοιχεία αποθηκεύονται σε αντικείμενα μεταβλητών JSON και στέλνονται στην εφαρμογή. Αναλυτικότερα το παραπάνω σενάριο πραγματοποιείται με τα παρακάτω βήματα:

- Το email και ο κωδικός χρήστη στέλνονται μέσω αιτήματος POST σε HTTP server.
- Ο server δέχεται τα δεδομένα και πραγματοποιεί έλεγχο επαλήθευσης μέσω του αντίστοιχου αρχείου, αν τα στοιχεία τα οποία δόθηκαν υπάρχουν στη βάση και αν είναι σωστά.
- Σε περίπτωση που είναι σωστά ο έλεγχος επιστρέφει το αποτέλεσμα σε μία μεταβλητή και η είσοδος του χρήστη πραγματοποιείται κανονικά. (error: FALSE, message: SUCCESS).
- Σε περίπτωση που ένα εκ των 2 στοιχείων είναι λάθος, ο έλεγχος επιστρέφει την αντίστοιχη μεταβλητή και προτρέπει τον χρήστη να ξαναπροσπαθήσει (error: TRUE, message: Λάθος στοιχεία, παρακαλώ προσπαθήστε ξανά)
- Τέλος, σε περίπτωση που και τα 2 στοιχεία είναι λάθος, δε θα βρεθούν στη βάση, οπότε ο έλεγχος επιστρέφει την αντίστοιχη μεταβλητή. (error: TRUE, message: Δεν βρέθηκαν τα στοιχεία).

Είναι σημαντικό να σημειωθεί σε αυτό το σημείο πώς αφού πραγματοποιηθεί η είσοδος του χρήστη με επιτυχία, μέσα στο activity πραγματοποιείται και μία διαδικασία καταχώρησης session στη συσκευή ώστε να μην είναι αναγκαίο από τον χρήστη να καταχωρεί τα προσωπικά του στοιχεία κάθε φορά που θέλει να εισέλθει. Το session αυτό καταχωρεί μέσω SQLite τα στοιχεία που μόλις δόθηκαν στη συσκευή, έπειτα από επιτυχή είσοδο. Τα στοιχεία παραμένουν αποθηκευμένα μέχρι ο χρήστης να πραγματοποιήσει διαδικασία εξόδου. Έως ότου πραγματοποιηθεί έξοδος ο χρήστης μπορεί να κλείσει και να ξανά εκκινήσει την εφαρμογή χωρίς να χρειάζεται να συμπληρώσει τα προσωπικά του στοιχεία του. Στην εικόνα που ακολουθεί (Εικόνα 25) εμφανίζονται κατά σειρά η αποστολή των στοιχείων που συμπλήρωσε ο χρήστης, αποθηκευμένα στις μεταβλητές String email και

password, στο αρχείο php στον διακομιστή (ο σύνδεσμος είναι αποθηκευμένος στη μεταβλητή URL\_LOGIN). Στη συνέχεια ακολουθεί ο έλεγχος αν ο χρήστης εισήλθε επιτυχώς, και ξεκινά η διαδικασία κατά την οποία δημιουργείται το session από τα στοιχεία αυτά. Τα στοιχεία που αποθηκεύονται είναι το uid, name email, created\_at και αντιστοιχούν στα δεδομένα του χρήστη από τη βάση. Τέλος, φαίνεται η ανακατεύθυνση του χρήστη στο MainActivity όταν η διαδικασία ολοκληρωθεί επιτυχώς.

```

btnLogin.setOnClickListener((View) -> {
    String email = inputEmail.getText().toString().trim();
    String password = inputPassword.getText().toString().trim();

    // Check for empty data in the form
    if (!email.isEmpty() && !password.isEmpty()) {
        // login user
        checkLogin(email, password);
    } else {
        // Prompt user to enter credentials
        Toast.makeText(getApplicationContext(), "Παρακαλώ συμπληρώστε όλα τα πεδία", Toast.LENGTH_LONG)
            .show();
    }
});

private void checkLogin(final String email, final String password) {
    // Tag used to cancel the request
    String tag_string_req = "req_login";

    progressDialog.setMessage("Είσοδος ...");
    showDialog();

    StringRequest strReq = new StringRequest(Method.POST,
        AppConfig.URL_LOGIN, new Response.Listener<String>() {

            @Override
            public void onResponse(String response) {
                Log.d(TAG, "Μήνυμα Login: " + response.toString());
                hideDialog();

                try {
                    JSONObject jsonObj = new JSONObject(response);
                    boolean error = jsonObj.getBoolean("error");

                    // Check for error node in json
                    if (!error) {
                        // user successfully logged in
                        // Create login session
                        session.setLogin(true);

                        // Now store the user in SQLite
                        String uid = jsonObj.getString("uid");

                        JSONObject user = jsonObj.getJSONObject("user");
                        String name = user.getString("name");
                        String email = user.getString("email");
                        String created_at = user.getString("created_at");

                        // Inserting row in users table
                        db.addUser(name, email, uid, created_at);

                        // Launch main activity
                        Intent intent = new Intent(LoginActivity.this,
                            MainActivity.class);

```

**Εικόνα 25:** Αποστολή δεδομένων για είσοδο του χρήστη και δημιουργία session

Στη συνέχεια παρουσιάζεται η οθόνη της εφαρμογής Android (Εικόνα 26) που δημιουργήθηκε, η οποία εμφανίζεται στο χρήστη προτρέποντας τον να εισάγει τα στοιχεία του και να εισέλθει στο λογαριασμό του.



**Εικόνα 26:** Το login screen της εφαρμογής

Για να είναι η εφαρμογή πολύ πιο φιλική προς τον χρήστη και για να προσφέρει μεγαλύτερη ευκολία, σε περίπτωση που ο χρήστης δεν έχει λογαριασμό, μπορεί άμεσα να δημιουργήσει έναν μέσω της εφαρμογής. Η προτροπή βρίσκεται στην αρχική οθόνη κάτω ακριβώς από το login, με τίτλο «Δεν είστε μέλος; Εγγραφείτε εδώ.», όπως φαίνεται στην προηγούμενη εικόνα.

Με το πάτημα της επιλογής αυτής ο χρήστης μεταφέρεται στο activity\_register το οποίο περιέχει μία απλή φόρμα συμπλήρωσης των στοιχείων του. Τα στοιχεία του activity\_register τα επεξεργάζεται το RegisterActivity. Εκεί ακολουθείται μία διαδικασία παρόμοια με αυτήν του LoginActivity, με μόνη διαφορά ότι αυτή τη φορά τα στοιχεία καταχωρούνται στη βάση μετά από έλεγχο. Η διαδικασία που ακολουθείται περιλαμβάνει τα παρακάτω βήματα:

- Ο χρήστης συμπληρώνει τα απαιτούμενα στοιχεία για τη δημιουργία λογαριασμού (Όνοματεπώνυμο, Email, Κωδικός)
- Τα στοιχεία αποστέλλονται όπως και πριν με POST στο αντίστοιχο αρχείο PHP στο διακομιστή για να πραγματοποιηθεί έλεγχος.
- Σε περίπτωση που ο έλεγχος εντοπίσει στη βάση πως το email που έδωσε ο χρήστης υπάρχει ήδη, επιστρέφει την αντίστοιχη τιμή μέσω μιας μεταβλητής και ενημερώνει το χρήστη. (error: TRUE , message: Το email που πληκτρολογήσατε χρησιμοποιείται ήδη).
- Σε περίπτωση που δεν εντοπιστεί κάποιο σφάλμα από τον έλεγχο τότε επιστρέφεται σε μία μεταβλητή η αντίστοιχη τιμή, και στέλνει το χρήστη

στη διαδικασία εισόδου (error: FALSE, message: Η εγγραφή ολοκληρώθηκε επιτυχώς).

Στη φόρμα δημιουργίας λογαριασμού ενός χρήστη, το μόνο δεδομένο στο οποίο βασίζεται ο έλεγχος ως μοναδικό στοιχείο ανά χρήστη είναι το email του, το οποίο θα πρέπει να έχει και μία σωστή μορφή. Επίσης, κάτω από την φόρμα βρίσκεται μία προτροπή σε περίπτωση που ο χρήστης εισήλθε κατά λάθος, και τον αποστέλλει ξανά πίσω στο login.

Στην εικόνα που ακολουθεί (Εικόνα 27) φαίνεται το περιεχόμενο του RegisterActivity στο οποίο γίνεται έλεγχος των στοιχείων και αν είναι συμπληρωμένα όλα τα πεδία εγγραφής. Αν ο έλεγχος είναι επιτυχής τα στοιχεία στέλνονται στο αντίστοιχο αρχείο php στο διακομιστή (ο σύνδεσμος βρίσκεται αποθηκευμένος στη μεταβλητή URL\_REGISTER). Τα πεδία μεταβλητών είναι Fullname, Email, Password και αποστέλλονται στα αντίστοιχα πεδία της βάσης (name, email, encrypted password).

```
// Register Button Click event
btnRegister.setOnClickListener((view) -> {
    String name = inputFullName.getText().toString().trim();
    String email = inputEmail.getText().toString().trim();
    String password = inputPassword.getText().toString().trim();

    if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty()) {
        registerUser(name, email, password);
    } else {
        Toast.makeText(getApplicationContext(),
            "Παρακαλώ συμπληρώστε όλα τα πεδία", Toast.LENGTH_LONG)
            .show();
    }
});

// Link to Login Screen
btnLinkToLogin.setOnClickListener((view) -> {
    Intent i = new Intent(getApplicationContext(),
        LoginActivity.class);
    startActivity(i);
    finish();
});
}

private void registerUser(final String name, final String email,
    final String password) {
    String tag_string_req = "req_register";

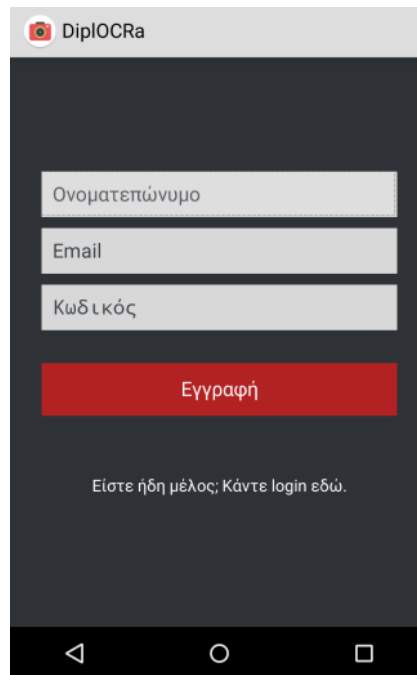
    progressDialog.setMessage("Registering ...");
    progressDialog.show();

    StringRequest strReq = new StringRequest(Method.POST,
        AppConfig.URL_REGISTER, new Response.Listener<String>() {
```

Εικόνα 27: Δημιουργία λογαριασμού χρήστη (RegisterActivity)

Το στοιχείο κειμένου που εισάγει ο χρήστης μετατρέπονται σε strings και αν είναι αποδεκτά και περάσουν τον έλεγχο, μεταφέρονται ως POST request στο σύνδεσμο με το διακομιστή.

Στην επόμενη εικόνα (Εικόνα 28) παρουσιάζεται το γραφικό περιβάλλον activity\_register.xml, που σχεδιάστηκε για τη δημιουργία λογαριασμού χρήστη αλλά και η προτροπή για επαναφορά της εφαρμογής στο login screen σε περίπτωση που ο χρήστης έχει ήδη λογαριασμό.



**Εικόνα 28:** Το register της εφαρμογής

Με την υλοποίηση και της εγγραφής (register), καλύφθηκε πλήρως η αναγκαία ύπαρξη ενός προσωπικού λογαριασμού από τους χρήστες της εφαρμογής, ακολουθώντας πολύ απλά και εύκολα βήματα. Θα πρέπει να σημειωθεί βέβαια, πώς για να έχει πρόσβαση ένας χρήστης στην εφαρμογή, θα πρέπει να σταλεί σχετικό αίτημα στο διαχειριστή. Από εκεί κι έπειτα ο χρήστης έχει τη δυνατότητα δημιουργίας λογαριασμού

## 4.2.2 Λειτουργίες και δυνατότητες της εφαρμογής

Έχοντας ολοκληρώσει λοιπόν αυτά τα βασικά σημεία, προχωρήσαμε στην κατασκευή του μέρους που συμπεριλαμβάνει τις λειτουργίες που παρέχει η εφαρμογή μας στο χρήστη.

Αμέσως μετά την επιτυχή είσοδο και τη δημιουργία session, ο χρήστης μεταφέρετε στο γραφικό περιβάλλον του main\_activity.xml. Αυτό αποτελεί το γραφικό περιβάλλον που πραγματοποιήθηκε και περιλαμβάνει τις δυνατότητες που έχει ένας χρήστης.

Αρχικά η οθόνη που βλέπει ο χρήστης ξεκινά με ένα μήνυμα που τον καλωσορίζει στην εφαρμογή και εμφανίζει τα στοιχεία του. Ακριβώς από κάτω του εμφανίζονται 3 κουμπιά, σε μορφή μενού, με τις δυνατότητες που έχει ο χρήστης.

Οι δυνατότητες αυτές είναι :

- Μεταφορά του χρήστη στο μενού λειτουργίας της κάμερας για τη λήψη και αναγνώριση δεδομένων.
- Εποπτεία τα των μετρήσεων που έχει ήδη πραγματοποιήσει, και
- Πραγματοποίηση εξόδου από τον προσωπικό του λογαριασμό.

Ο κώδικας της δραστηριότητας η οποία είναι υπεύθυνη για τη λειτουργία του activity\_main, βρίσκεται στο αρχείο με όνομα MainActivity. Στην επόμενη εικόνα (Εικόνα 29)παρουσιάζονται οι δυνατότητες που μόλις αναφέραμε.



```

HashMap<String, String> user = db.getUserDetails();

String name = user.get("name");
String email = user.get("email");

// Displaying the user details on the screen
txtName.setText(name);
txtEmail.setText(email);

// Logout button click event
btnLogout.setOnClickListener((v) -> { logoutUser(); });

btnOcr.setOnClickListener((view) -> {
    Intent i = new Intent(getApplicationContext(),
        SimpleAndroidOCRActivity.class);

    startActivity(i);
    finish();
});

btnmeasure.setOnClickListener((view) -> {
    Intent i = new Intent(getApplicationContext(), MeasurementsActivity.class);
    startActivity(i);
    finish();
});
}

private void logoutUser() {
    session.setLogin(false);

    db.deleteUsers();

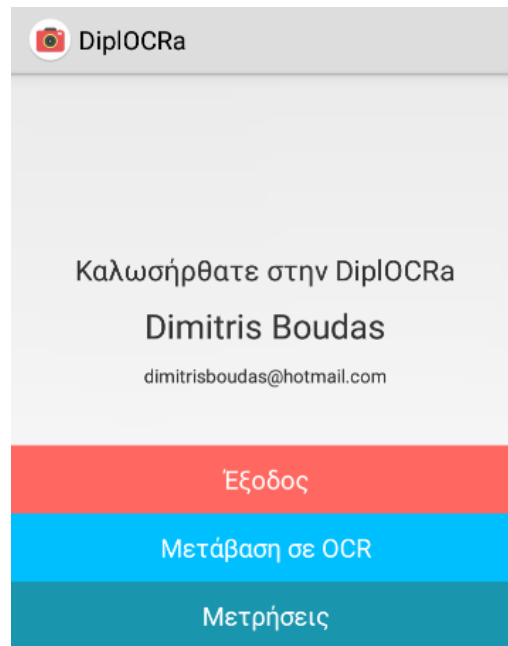
    // Launching the login activity
    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
    startActivity(intent);
    finish();
}

```

**Εικόνα 29:** Το MainActivity της εφαρμογής

Εδώ παρατηρούμε τον τρόπο με τον οποίο εμφανίζονται τα στοιχεία του χρήστη στην κύρια οθόνη, καθώς και τις λειτουργίες των κουμπιών με που υπάρχουν στην οθόνη του χρήστη. Τέλος, εμφανίζεται η διαδικασία του logout και η ανακατεύθυνση του χρήστη στην οθόνη του login, όταν πραγματοποιήσει έξοδο.

Στη συνέχεια θα παρουσιάσουμε την οθόνη που βλέπει ο χρήστης (Εικόνα 30) και το πώς πραγματοποιήθηκε με τη δημιουργία του activity\_main.xml. Με τη βοήθεια της εικόνας που ακολουθεί θα γίνει μία αναλυτική περιγραφή των δυνατοτήτων των κουμπιών που αναφέραμε παραπάνω. Στην παρακάτω εικόνα εμφανίζεται η οθόνη του main\_activity που βλέπει ο χρήστης μετά από επιτυχή είσοδο.



**Εικόνα 30:** Η κύρια οθόνη (activity\_main)

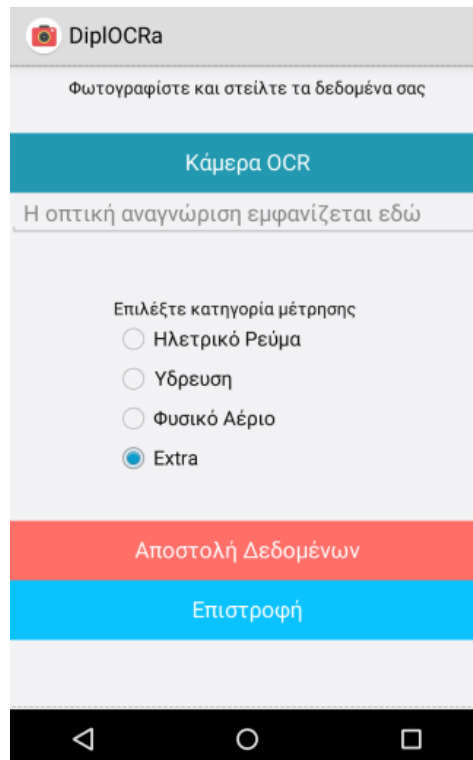
Τα κουμπιά τα οποία φαίνονται εδώ είναι οι λειτουργίες οι οποίες αναφέρθηκαν επιγραμματικά πιο πάνω. Αναλυτικότερα, με την σειρά που εμφανίζονται, το κουμπί «Έξοδος» είναι υπεύθυνο για την αποσύνδεση του χρήστη από την εφαρμογή. Το πάτημα του κουμπιού αυτού επαναφέρει το χρήστη στην λειτουργία του login. Βέβαια, επειδή όπως αναφέραμε, κατά την είσοδο του χρήστη δημιουργείται session που διατηρεί τα στοιχεία του, για να παρέχουμε στο χρήστη ασφάλεια αποσύνδεσης, το session αυτό καταστρέφεται και τα στοιχεία διαγράφονται από τον προσωρινό πίνακα της συσκευής.

Αμέσως μετά ακολουθεί το κουμπί «Μετάβαση σε OCR». Όπως υποδηλώνει και ο τίτλος του, το κουμπί αυτό μεταφέρει το χρήστη στη λειτουργία όπου γίνεται λήψη της φωτογραφίας μέσω της κάμερας της συσκευής.

Τέλος το κουμπί «Μετρήσεις» μεταφέρει το χρήστη σε μία οθόνη στην οποία μπορεί να δει τις μετρήσεις που είναι καταχωρημένες στο λογαριασμό του, καθώς και την ημερομηνία πραγματοποίησής τους. Αξίζει να σημειωθεί σε αυτό το σημείο πως, στην περίπτωση που κάποιος χρήστης δεν έχει κάποια ενεργή μέτρηση αποθηκευμένη στο λογαριασμό του, η εφαρμογή

ανακατευθύνει το χρήστη στη λειτουργία «Μετάβαση σε OCR» ώστε να προστεθεί κάποια μέτρηση και να μην εμφανίζεται απλά μία κενή οθόνη.

Αναλυτικότερα, όταν ο χρήστης πατήσει το κουμπί «Μετάβαση σε OCR» ή καταλήξει εκεί λόγω έλλειψης μετρήσεων, του εμφανίζεται η οθόνη η οποία αντιστοιχεί στο αρχείο main.xml. Στην εικόνα που ακολουθεί (Εικόνα 31) εμφανίζεται η οθόνη που βλέπει ο χρήστης όταν μεταβαίνει στο main.xml.



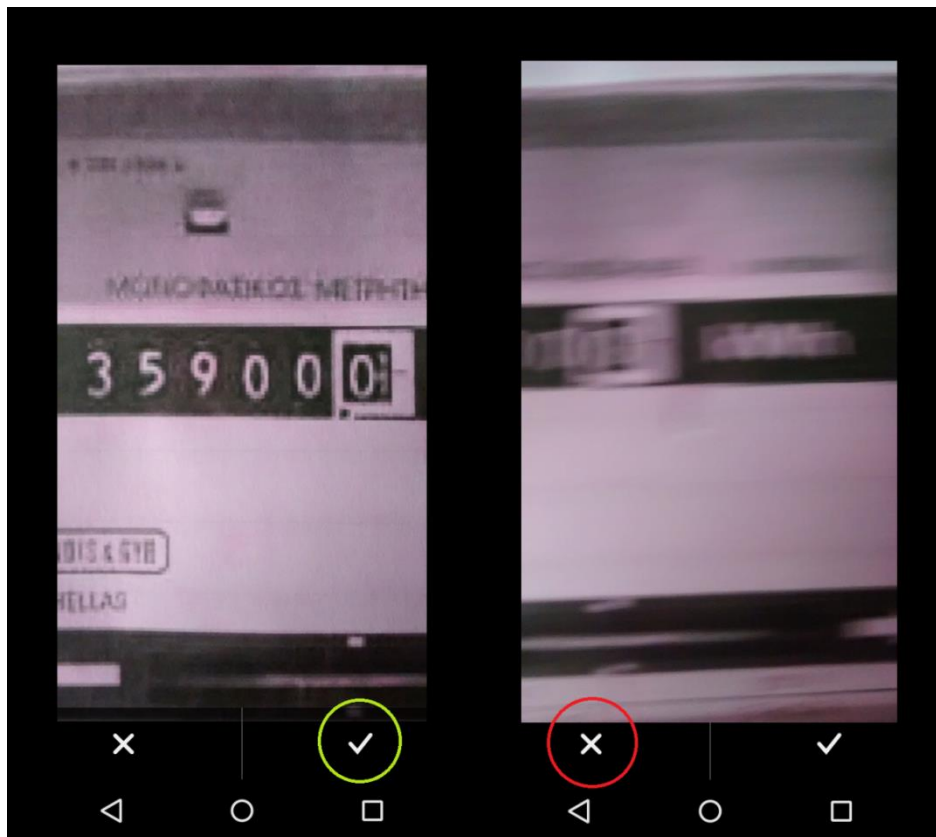
**Εικόνα 31:** Το main.xml της εφαρμογής

Στην οθόνη αυτή υπάρχουν 3 κουμπιά τα οποία μπορεί να χρησιμοποιήσει ο χρήστης, καθώς και μία λίστα επιλογών που αποτελείται από 4 επιλογές και είναι βασισμένη σε λειτουργία τύπου RadioButton list. Επιπλέον υπάρχει ένα κενό (αρχικά) πεδίο κειμένου τύπου editText στο οποίο εμφανίζεται το κείμενο που αναγνωρίστηκε.

Το πρώτο κουμπί με όνομα «Κάμερα OCR» ενεργοποιεί και μεταφέρει το χρήστη στην κάμερα της συσκευής. Να σημειωθεί πως η εφαρμογή δεν επηρεάζει καθόλου τις λειτουργίες που παρέχει η κάμερα της εκάστοτε συσκευής στην οποία είναι εγκατεστημένη η εφαρμογή (όπως για παράδειγμα auto-focus, zoom κλπ.) αλλά χρησιμοποιεί για τη διευκόλυνση της αυτές τις λειτουργίες όπου κρίνεται αναγκαίο. Αφού λοιπόν ο χρήστης φωτογραφίσει την

εικόνα, του εμφανίζεται το αντικείμενο που μόλις τραβήχτηκε με 2 επιλογές. Η μία επιλογή είναι το «OK» (με μορφή tick) και επιλέγεται από τον χρήστη για να επιβεβαιώσει ότι στην εικόνα που τραβήχτηκε φαίνεται καθαρά το περιεχόμενο που θέλει από την εφαρμογή να αναγνωρίσει. Με τα πάτημα της επιλογής αυτής η συσκευή επεξεργάζεται τα αλφαριθμητικά δεδομένα της εικόνας και εμφανίζει το αποτέλεσμα. Ταυτόχρονα η εφαρμογή ανιχνεύει εάν υπάρχει εξωτερικός χώρος αποθήκευσης στη συσκευή (π.χ. κάρτα SD) και αν δεν εντοπίσει κάτι, αποθηκεύει στη μνήμη της συσκευής την τελευταία φωτογραφία που αναγνωρίστηκε από την εφαρμογή δημιουργώντας έναν κατάλογο με όνομα «app» και ονομάζοντας το αρχείο «ocr».

Σε αντίθετη περίπτωση ο χρήστης μπορεί να πατήσει την επιλογή «Άκυρο» (με μορφή X), εάν κρίνει ότι η εικόνα δεν είναι κατάλληλη για επεξεργασία και αναγνώριση (σε περίπτωση που για παράδειγμα η κάμερα κουνήθηκε ή ο φωτισμός δεν ήταν σωστός για να βγει μία ευκρινής φωτογραφία). Σε αυτήν την περίπτωση η εφαρμογή δεν αποθηκεύει τίποτα και ανακατευθύνει το χρήστη και πάλι στην κάμερα της συσκευής για να πραγματοποιήσει εκ νέου τη λήψη μιας φωτογραφίας. Παρακάτω εμφανίζονται αυτές περιπτώσεις με παράδειγμα εικόνων (Εικόνα 32).



**Εικόνα 32:** Παράδειγμα αποδοχής και απόρριψης φωτογραφίας

Όπως γίνεται προφανές, μέσα από αυτό το παράδειγμα, στην αριστερή εικόνα βλέπουμε τη λήψη μίας ευκρινέστατης φωτογραφίας με πολύ καλές πιθανότητες πλήρους αναγνώρισης από την εφαρμογή, ενώ στα δεξιά εμφανίζεται ένα παράδειγμα πολύ κακής λήψης φωτογραφίας, για την οποία θα πρέπει ο χρήστης να ξαναδοκιμάσει, διαφορετικά δεν θα λάβει σε καμία περίπτωση σωστά αποτελέσματα. Βέβαια, σε αυτό το σημείο πρέπει να επισημανθεί ότι με βάση τον αλγόριθμο που χρησιμοποιείται στην εφαρμογή, κάποιες φορές μπορεί ακόμα και η αριστερή φωτογραφία να μην αποτελεί το καλύτερο δυνατό μοντέλο προς αναγνώριση, διότι ο αλγόριθμος αυτός είναι σχεδιασμένος να επεξεργάζεται όλες τις πιθανές γραμμές και σχήματα ως δεδομένα προς αναγνώριση. Παρόλα αυτά όμως, η αριστερή εικόνα έχει πολύ υψηλές πιθανότητες να προσφέρει πιο σαφή αποτελέσματα.

Αφού λοιπόν ο χρήστης παρέχει στην εφαρμογή μια καθαρή εικόνα στην οποία να είναι ευδιάκριτα τα αριθμητικά δεδομένα της μέτρησης, η εφαρμογή αναλαμβάνει την αναγνώριση των στοιχείων. Η αναγνώριση αυτή εμφανίζεται στο πεδίο editText που βρίσκεται κάτω ακριβώς από το κουμπί «Κάμερα OCR».

Σε αυτό το πεδίο δίνεται η δυνατότητα στο χρήστη να ελέγξει κατά πόσο η αναγνώριση ήταν ακριβής και να διορθώσει κάποιες λεπτομέρειες σε περίπτωση λάθους. Διαφορετικά, ο χρήστης μπορεί να επιλέξει να δοκιμάσει ξανά, με μία καινούρια εικόνα.

Ο απαραίτητος κώδικας για τη μορφοποίηση και την αναγνώριση του κειμένου βρίσκεται στη δραστηριότητα με όνομα αρχείου `AndroidOCRActivity`. Πέρα από τον αλγόριθμο που χρησιμοποιείται για την αναγνώριση, στο activity αυτό πραγματοποιούνται και λειτουργίες μορφοποίησης. Συγκεκριμένα, για τη διευκόλυνση του αλγορίθμου αναγνώρισης, προεπιλεγμένος προσανατολισμός εικόνας ( default image orientation), αποτελείται από την λήψη εικόνας από όρθια και κατακόρυφη θέση της συσκευής. Παρόλα αυτά αν ο χρήστης δοκιμάσει να βγάλει μία φωτογραφία από διαφορετική γωνία, η εφαρμογή προσπαθεί να βελτιώσει την κλίση αυτή χωρίς να επηρεάσει το αποτέλεσμα της εικόνας. Στην παρακάτω εικόνα (Εικόνα 33) εμφανίζεται το κομμάτι κώδικα του `AndroidOCRActivity` που πραγματοποιεί αυτή τη διαδικασία.

```
protected void onPhotoTaken() {
    _taken = true;

    BitmapFactory.Options options = new BitmapFactory.Options();
    options.inSampleSize = 4;

    Bitmap bitmap = BitmapFactory.decodeFile(_path, options);

    try {
        ExifInterface exif = new ExifInterface(_path);
        int exifOrientation = exif.getAttributeInt(
            ExifInterface.TAG_ORIENTATION,
            ExifInterface.ORIENTATION_NORMAL);

        Log.v(TAG, "Orient: " + exifOrientation);

        int rotate = 0;

        switch (exifOrientation) {
            case ExifInterface.ORIENTATION_ROTATE_90:
                rotate = 90;
                break;
            case ExifInterface.ORIENTATION_ROTATE_180:
                rotate = 180;
                break;
            case ExifInterface.ORIENTATION_ROTATE_270:
                rotate = 270;
                break;
        }
    }
}
```

**Εικόνα 33:** Διόρθωση προσανατολισμού εικόνας

Εδώ παρατηρούμε τον έλεγχο που πραγματοποιείται σε περίπτωση που έχει γίνει λήψη φωτογραφίας. Μόλις αποτυπωθεί ο προσανατολισμός της εικόνας, πραγματοποιείται περιστροφή 90, 180 ή 270 μοιρών για τη διευκόλυνση του αλγορίθμου αναγνώρισης.

Στη συνέχεια, ορίζεται το ύψος και το πλάτος της εικόνας για επεξεργασία `bitmap`, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 34).

```

Log.v(TAG, "Rotation: " + rotate);

if (rotate != 0) {

    int w = bitmap.getWidth();
    int h = bitmap.getHeight();

    Matrix mtx = new Matrix();
    mtx.preRotate(rotate);

    bitmap = Bitmap.createBitmap(bitmap, 0, 0, w, h, mtx, false);

}

bitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);
} catch (IOException e) {
    Log.e(TAG, "Sfalma orientation: " + e.toString());
}

```

**Εικόνα 34:** Τελική επεξεργασία εικόνας (bitmap)

Με την επεξεργασία bitmap, η εικόνα μεταβαίνει στο τελικό στάδιο μορφοποίησης που αποτελείται από Bitmap\_Config = ARGB\_8888. Η συγκεκριμένη διαμόρφωση σε σύστημα Android λειτουργεί αποθηκεύοντας κάθε 1 pixel εικόνας σε 4 bytes. Κάθε ξεχωριστό κανάλι (RGB και Alpha για λόγους διαύγειας της εικόνας) αποθηκεύεται με 8 bits ακριβείας, πράγμα που σημαίνει 256 πιθανές τιμές. Η μορφοποίηση ARGB\_8888 είναι πάρα πολύ ευέλικτη και προσφέρει την καλύτερη δυνατή ποιότητα. Παρόλο που κάποιες φορές αυτή η ποιότητα μπορεί να θεωρηθεί μειονέκτημα (ψηλότερη ποιότητα σημαίνει ταυτόχρονα και ανάγκη για περισσότερο αποθηκευτικό χώρο), παραμένει η καλύτερη επιλογή στον τομέα αυτόν, διότι προσφέρει τις καλύτερες πιθανότητες για ακριβέστερες αναγνώρισεις εικόνων.

Στη συνέχεια εισάγεται ο απαραίτητος κώδικας που χρησιμοποιεί τη βιβλιοθήκη tess-two μέσω του προγραμματιστικού περιβάλλοντος TessBaseAPI (Tesseract Base Application Programming Interface) όπως παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 35).

```

Log.v(TAG, "Before baseApi");

TessBaseAPI baseApi = new TessBaseAPI();
baseApi.setDebug(true);
baseApi.init(DATA_PATH, "eng");
baseApi.setImage(bitmap);

String recognizedText = baseApi.getUif8Text();

baseApi.end();

Log.v(TAG, "OCRED TEXT: " + recognizedText);

if ( Lang.equalsIgnoreCase("eng") ) {
    recognizedText = recognizedText.replaceAll("[^a-zA-Z0-9]+", " ");
}

recognizedText = recognizedText.trim();

if ( recognizedText.length() != 0 ) {
    _field.setText(_field.getText().toString().length() == 0 ? recognizedText : _field.getText() + " " + recognizedText);
    _field.setSelection(_field.getText().toString().length());
}
}

```

**Εικόνα 35:** Αναγνώριση μέσω TessBaseAPI

Εδώ φαίνεται η πραγματοποίηση της αναγνώρισης και η προσθήκη του στην οθόνη σε κωδικοποίηση utf8. Στο σημείο αυτό το κείμενο έχει αναγνωριστεί και αποθηκεύεται στη μεταβλητή recognizedText. Για την ακόμη καλύτερη απόδοσή του όμως, ακολουθεί ένας έλεγχος ο οποίος ορίζει ότι γίνονται αποδεκτά μόνο τα αλφαριθμητικά στοιχεία βασισμένα στη γλώσσα αναγνώρισης (στην περίπτωση μας λατινικοί χαρακτήρες μέσω του eng.traineddata). Με αυτόν τον τρόπο απομακρύνονται κάποιοι αλλοιωμένοι χαρακτήρες από το κείμενο της αναγνώρισης, και το «καθαρό», πλέον κείμενο, εμφανίζεται στην οθόνη του χρήστη.

Όταν ο χρήστης έχει πλέον το σωστό αποτέλεσμα, του δίνεται η δυνατότητα να επιλέξει τον τύπο της μέτρησης στην οποία θέλει να καταχωρηθεί στη βάση δεδομένων. Η λίστα με τις πιθανές επιλογές βρίσκεται κάτω ακριβώς από το αναγνωρισμένο κείμενο και ακολουθεί τη μορφή RadioButton list. Δηλαδή ο χρήστης μπορεί να επιλέξει μόνο ένα είδος μέτρησης κάθε φορά (π.χ. Φυσικό Αέριο). Αφού οριστεί αυτή η λίστα με τις αντίστοιχες μεταβλητές, πραγματοποιείται ένα POST request στο αντίστοιχο αρχείο php στο διακομιστή για την αποστολή, των δεδομένων στην αντίστοιχη κατηγορία. Στο αίτημα αυτό αποθηκεύονται οι απαραίτητες μεταβλητές με τα δεδομένα της αναγνώρισης και η επιλογή του χρήστη από τη λίστα RadioButton. Ο σχετικός κώδικας παρουσιάζεται στην παρακάτω εικόνα (Εικόνα 36).



```

_btnsenddata.setOnClickListener((view) -> {
    StringRequest request = new StringRequest(Request.Method.POST, insertUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            System.out.println(response.toString());
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> parameters = new HashMap<>();
            parameters.put("data", _field.getText().toString());

            db = new SQLiteHandler(getApplicationContext());
            HashMap<String, String> user = db.getUserDetails();

            String uid = user.get("uid");
            if (_rdrevma.isChecked())
                parameters.put("type", "Electricity");
            else if (_rdaerio.isChecked())
                parameters.put("type", "Natural Gas");
            else if (_rdnero.isChecked())
                parameters.put("type", "Water");
            else if (_rdextra.isChecked())
                parameters.put("type", "Extra");
            parameters.put("uid", uid);

            return parameters;
        }
    };
    requestQueue.add(request);

    Context context = getApplicationContext();
    CharSequence text = "Επιτυχής αποστολή δεδομένων";
    int duration = Toast.LENGTH_SHORT;

```

**Εικόνα 36:** Καταχώρηση δεδομένων σε κατηγορίες

Με βάση την παραπάνω εικόνα, δημιουργείται ένα Map παραμέτρων, όπου αποθηκεύονται ως data το αναγνωρισμένο κείμενο (μέσω του πεδίου `_field` όπως εμφανίστηκε στην προηγούμενη εικόνα). Στη συνέχεια, γίνεται ο έλεγχος για ποια από τις 4 κατηγορίες είναι επιλεγμένη ορίζοντας σε 2 καινούριες μεταβλητές `type` και `Electricity` (αν επιλεγθεί η πρώτη για παράδειγμα), την επιλογή του χρήστη και το είδος που αντιπροσωπεύει η επιλογή αυτή. Σε συνδυασμό με το `uid` του χρήστη που είναι αποθηκευμένο στο `session` πραγματοποιείται το `request` (ο σύνδεσμος του `php` αρχείου βρίσκεται στην μεταβλητή `insertUrl`) και αποστέλλονται όλα τα δεδομένα. Στον διακομιστή πραγματοποιείται ο έλεγχος εγκυρότητας του `uid` (μέσω του της αντιστοιχίας του με το `unique_id` ενός χρήστη), και αν είναι επιτυχής τα δεδομένα των μεταβλητών που αναφέραμε περνούν σε σχετικό αίτημα προς τη βάση δεδομένων και καταχωρούνται στα πεδία που πρέπει. Τέλος, οι πληροφορίες αποθηκεύονται στην αντίστοιχη κατηγορία στα προσωπικά δεδομένα του ενεργού χρήστη, και εμφανίζεται κατάλληλο μήνυμα.

Το «κλειδί» της σωστής λειτουργίας σε αυτή τη διεργασία που εξασφαλίζει την ομαλή καταχώρηση των μετρήσεων κάθε χρήστη στην σωστή κατηγορία, είναι η ύπαρξη του `session` που υφίσταται σε κάθε διεργασία. Με την

ύπαρξή του κάθε χρήστης επηρεάζει μόνο τα δικά του πεδία στη βάση δεδομένων, και αποφεύγεται έτσι οποιαδήποτε περίπτωση να γίνει κάποιο μπέρδεμα με τα δεδομένα, όσοι χρήστες και αν αποστείλουν μία μέτρηση. Ένας ακόμα παράγοντας που βοηθάει σε αυτό το σημείο, είναι το γεγονός ότι κάθε επίπεδο χρήστη στέλνει τα δεδομένα σε δικό του πίνακα. Δηλαδή, ακόμη κι αν στείλουν ταυτόχρονα μία καταγραφή, ένας τεχνικός αλλά και ο απλός χρήστης για τον οποίο είναι υπεύθυνος ο τεχνικός, οι μετρήσεις θα αποθηκευτούν ξεχωριστά.

Στη λίστα αυτή που εμφανίζεται στον καταγραφέα, υπάρχει ως προεπιλογή το RadioButton «Extra». Αυτό συμβαίνει ώστε ακόμα κι αν ο χρήστης ξεχάσει να επιλέξει κάτι από τη λίστα, να μην χαθεί η μέτρηση του, αλλά να αποθηκευτεί στην επιπλέον κατηγορία του λογαριασμού του, και σε περίπτωση λάθους μπορεί αργότερα μέσω του προσωπικού του λογαριασμού να επεξεργαστεί τη μέτρηση αυτή και να την καταχωρήσει στη σωστή κατηγορία.

Το τελευταίο κουμπί που υπάρχει στην οθόνη του χρήστη με όνομα «Επιστροφή», δίνει όπως είναι προφανές τη δυνατότητα επιστροφής στην προηγούμενη, κύρια οθόνη.

Έχοντας πλέον δημιουργήσει κάποια ή κάποιες μετρήσεις, ο χρήστης μπορεί να επιστρέψει στο αρχικό μενού και να ελέγξει τις μετρήσεις που έκανε πατώντας στο κουμπί «Μετρήσεις». Εκεί εμφανίζονται πλέον τα δεδομένα του σε μορφή λίστας, και δίπλα ακριβώς εμφανίζεται η ημερομηνία αλλά και η ώρα που πραγματοποιήθηκαν. Για την πραγματοποίηση της λίστας αυτής δημιουργήθηκε ακόμα μία δραστηριότητα με όνομα MeasurementsActivity. Στην εικόνα που ακολουθεί (Εικόνα 37).εμφανίζεται ο κώδικας με τον οποίο η εφαρμογή διαβάζει τα στοιχεία του πίνακα User\_Data για τον χρήστη που είναι συνδεδεμένος και τα εμφανίζει.

```

StringRequest strReq = new StringRequest(Request.Method.POST,
    url, new Response.Listener<String>() {

    @Override
    public void onResponse(String response) {

        try {
            JSONObject json = new JSONObject(response);
            Log.d("Μετρήσεις : ", json.toString());
            int success = json.getInt(TAG_SUCCESS);

            if (success == 1) {
                measures = json.getJSONArray(TAG_MEASURES);

                for (int i = 0; i < measures.length(); i++) {
                    JSONObject c = measures.getJSONObject(i);

                    String id = c.getString(TAG_PID);
                    String name = c.getString(TAG_NAME);
                    String date = c.getString(TAG_DATE);

                    name = name + " (" + date + ") ";
                    HashMap<String, String> map = new HashMap<String, String>();

                    map.put(TAG_PID, id);
                    map.put(TAG_NAME, name);

                    measuresList.add(map);
                }
            } else {
                // no measure found
                // Open OCR Activity
                Intent i = new Intent(getApplicationContext(),
                    SimpleAndroidOCRActivity.class);
                i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(i);
            }
        }
    }
}

```

**Εικόνα 37:** Το MeasurementsActivity της εφαρμογής

Όπως απεικονίζεται παραπάνω, στη διεργασία αυτή και πάλι με τη βοήθεια μετατροπής των μεταβλητών σε αντικείμενα JSON, στέλνουμε το request με POST στο αντίστοιχο αρχείο php του διακομιστή (ο σύνδεσμος βρίσκεται στη μεταβλητή url). Για το αίτημα αυτό αποθηκεύονται στις μεταβλητές pid, name, date, οι αντίστοιχες τιμές id, text\_data, date, από τον πίνακα User\_Data. Ακολούθως, από το αρχείο του διακομιστή, γίνεται ο έλεγχος αν υπάρχουν καταγεγραμμένες μετρήσεις και αν βρίσκονται αποθηκεύονται σε αντίστοιχες μεταβλητές ώστε να πραγματοποιηθεί το αίτημα τη βάση δεδομένων. Εάν ο έλεγχος δεν βρει αποθηκευμένες τιμές, όπως αναφέραμε καινωρίτερα, βλέπουμε ότι κλείνει τη διεργασία αυτή, ανακατευθύνοντας το χρήστη στη διεργασία πραγματοποίησης καταχωρήσεων.

Στη συνέχεια με τη χρήση HashMap εμφανίζουμε τις τιμές των μεταβλητών (προσθέτοντας την ημερομηνία στην εκτύπωση, διότι μπορούν να αποθηκευτούν έως 2 μεταβλητές κάθε φορά σε ένα HashMap). Τέλος,

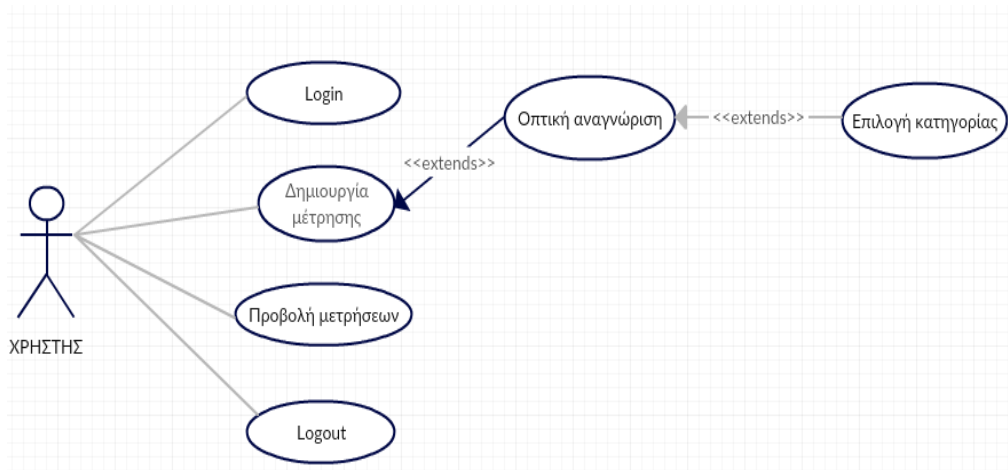
εμφανίζουμε τα αποτελέσματα από το αντίστοιχο αρχείο, measurements.xml σε μορφή list\_view, όπως αναφέρθηκε παραπάνω. Το αποτέλεσμα αυτών των δύο φαίνεται στην παρακάτω εικόνα (Εικόνα 38).

DiplOCRa	
12345687	(2016-06-20 13:07:50)
12344321	(2016-06-20 13:09:01)
12345678	(2016-06-20 13:05:40)
12345678	(2016-06-20 13:05:42)

**Εικόνα 38:** Το measurement.xml της εφαρμογής

Εδώ εμφανίζονται στα δεξιά τα δεδομένα της κάθε μέτρησης, ενώ από δίπλα τους, μέσα σε παρενθέσεις βλέπουμε την ημερομηνία και την ώρα όπου πραγματοποιήθηκαν οι μετρήσεις αυτές.

Παρακάτω εμφανίζεται η συνολική λειτουργία της εφαρμογής από τους χρήστες μέσω διαγράμματος use-case (Εικόνα 39).



**Εικόνα 39:** Διάγραμμα use-case χρήστη

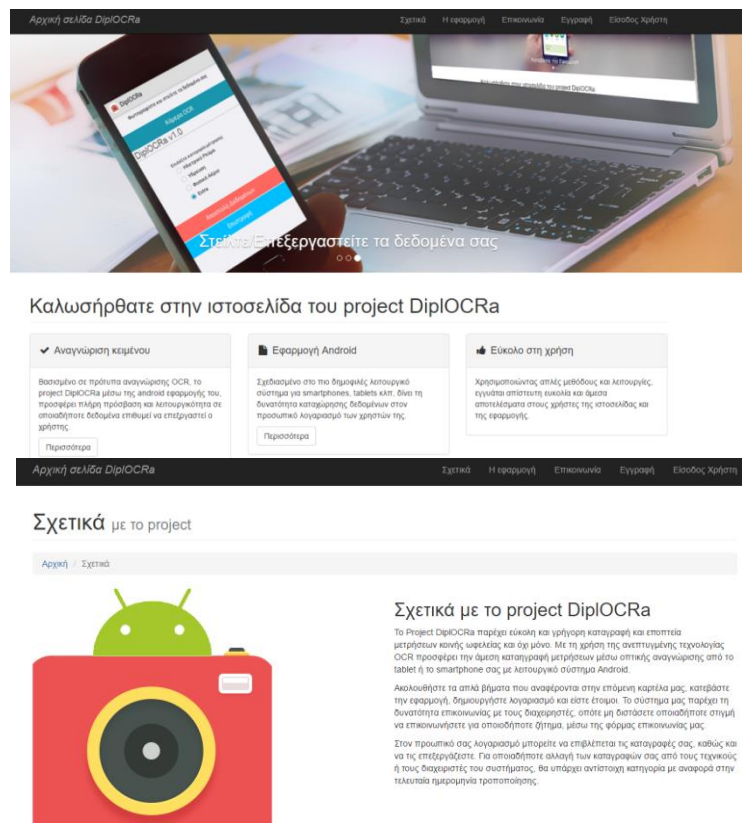
Έχοντας πλέον αναλύσει και την εφαρμογή Android που αναπτύχθηκε, θα συνεχίσουμε με την ανάπτυξη της ιστοσελίδας που πραγματοποιήθηκε. Θα γίνει αναλυτική αναφορά στις λειτουργίες της αλλά και στον τρόπο με τον οποίο αυτή συνεργάζεται και αλληλεπιδρά με τη βάση δεδομένων αλλά και με την εφαρμογή.

## 4.3 Ανάπτυξη ιστοχώρου

Η ιστοσελίδα που δημιουργήθηκε στην παρούσα διπλωματική εργασία, βασίστηκε πάνω στη δημιουργία ενός προφίλ εταιρίας. Στη συνέχεια θα αναλυθούν οι δυνατότητες που δίνει η ιστοσελίδα σε κάθε επίπεδο χρήστη που υπάρχει, καθώς και όλες οι λειτουργίες της.

### 4.3.1 Αρχική σελίδα

Δεδομένου ότι η ιστοσελίδα μας προβάλλει ένα προφίλ εταιρίας, στην αρχική της σελίδα, παρουσιάζονται γενικές πληροφορίες του project σε διάφορες καρτέλες τις οποίες μπορεί να τις περιηγηθεί οποιοσδήποτε χρήστης, εγγεγραμμένος και μη. Στην παρακάτω εικόνα (Εικόνα 40) εμφανίζονται αναφορικά η αρχική σελίδα και η καρτέλα «Σχετικά», που μπορεί να διαβάσει ένας επισκέπτης στη σελίδα.



**Εικόνα 40:** Αρχική σελίδα ιστοσελίδας

Όπως φαίνεται παραπάνω, στην πρώτη του επίσκεψη ένας απλός χρήστης παίρνει μια πρώτη εικόνα σχετικά με την ιστοσελίδα, και μπορεί να διαβάσει αν

επιθυμεί, πληροφορίες σχετικές με το project αλλά και πηγές που σχετίζονται και αναλύουν τα δομικά κομμάτια που αναπτύχθηκαν μαζί με τις λειτουργίες και δυνατότητες που προσφέρει το project αυτό.

Στη συνέχεια θα συνεχίσουμε την αναφορά στην ιστοσελίδα, παρουσιάζοντας τις λειτουργίες του διαχειριστή, και αργότερα των υπόλοιπων χρηστών.

### 4.3.2 Λειτουργίες Διαχειριστή (υπερχρήστη)

Οι λειτουργίες του διαχειριστή αφορούν τη δημιουργία νέων χρηστών, την καταχώρηση του στα επιτρεπτά επίπεδα χρήστη, την εποπτεία των μετρήσεων και την επεξεργασία των δεδομένων με διάφορους τρόπους.

Απαραίτητες προϋποθέσεις για τη δημιουργία ενός χρήστη από το διαχειριστή είναι, η εισαγωγή του Ονοματεπώνυμου του χρήστη, η διεύθυνση email του, ο κωδικός που αντιστοιχεί στο χρήστη αυτόν, η διεύθυνσή του και το επίπεδο χρήστη στο οποίο θα καταχωρηθεί. Για να πραγματοποιηθεί επιτυχώς η καταχώρηση του χρήστη με τα στοιχεία που μόλις αναφέραμε υπάρχουν 2 περιορισμοί. Η διεύθυνση email θα πρέπει να είναι έγκυρη και μοναδική στη βάση, και το επίπεδο χρήστη που θα εισαχθεί να συμβαδίζει με τον ορισμό των επιπέδων όπως αναφέρθηκαν παραπάνω. Στην παρακάτω εικόνα (Εικόνα 41) βλέπουμε τη φόρμα προσθήκης χρήστη του διαχειριστή.

## Προσθήκη Νέου Χρήστη

🏠 Πίνακας Ελέγχου / 👤 Προσθήκη Νέου Χρήστη

👤 Φόρμα Προσθήκης Χρήστη

Όνοματεπώνυμο:

Email:

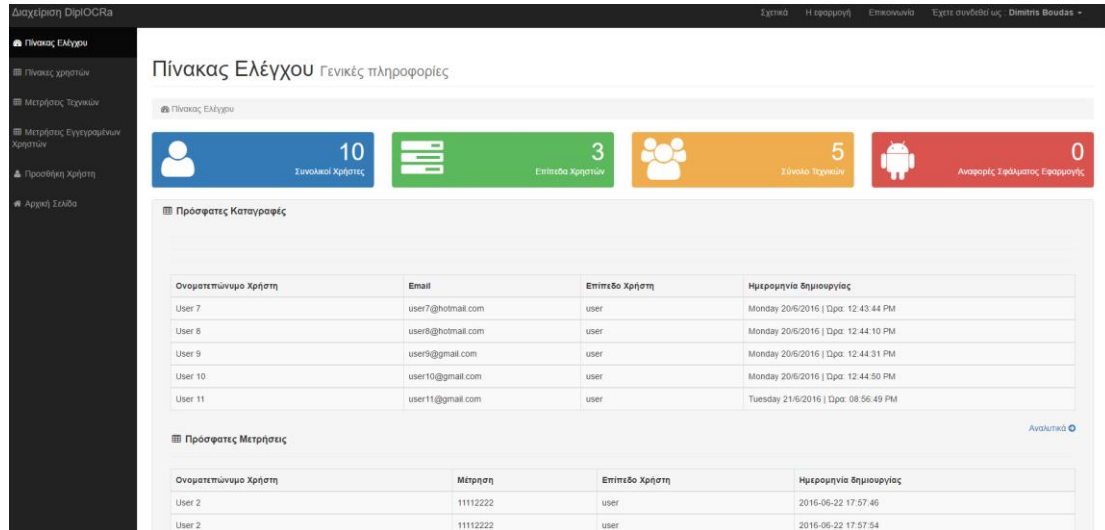
Κωδικός:

Διεύθυνση:

Επίπεδο Χρήστη:

**Εικόνα 41:** Φόρμα προσθήκης χρήστη διαχειριστή

Μόλις γίνει επιτυχής προσθήκη ενός χρήστη, εμφανίζεται στο διαχειριστή η αρχική του σελίδα, η οποία ενημερώνεται αυτομάτως για την πρόσφατη αλλαγή της προσθήκης που έγινε. Στην παρακάτω εικόνα παρουσιάζεται (Εικόνα 42) αναλυτικά το κεντρικό μενού του διαχειριστή.



**Εικόνα 42:** Κεντρικό μενού διαχειριστή

Στο μενού αυτό, εμφανίζονται στο διαχειριστή γενικές πληροφορίες όπως για παράδειγμα οι πίνακες οι οποίοι περιέχουν τις πιο πρόσφατες καταχωρήσεις χρηστών που πραγματοποιήθηκαν, καθώς και τις πιο πρόσφατες μετρήσεις, ανεξαρτήτως επιπέδου χρήστη. Στα αριστερά, υπάρχει μία λίστα επιλογών από την οποία μπορεί να περιηγηθεί ο χρήστης στις υπόλοιπες καρτέλες διαχείρισης.

Αφού λοιπόν καταχωρηθεί ένας καινούριος χρήστης, όπως περιγράφηκε παραπάνω, εμφανίζεται αμέσως στους πίνακες χρηστών που είναι διαθέσιμοι στο διαχειριστή. Στους πίνακες αυτούς παρουσιάζονται όλα τα στοιχεία των χρηστών χωρισμένα σε κατηγορίες ανάλογα με το επίπεδο τους. Εκεί, ο διαχειριστής μπορεί άμεσα να πραγματοποιήσει τροποποιήσεις, είτε αυτό σημαίνει να επεξεργαστεί και να αλλάξει τα στοιχεία ενός χρήστη, είτε ακόμα και να τον διαγράψει. Στην παρακάτω εικόνα (Εικόνα 43) παρουσιάζεται αναλυτικά ο πίνακας των χρηστών ανά επίπεδα.

### Διαχειριστές Συστήματος

Όνοματεπώνυμο Χρήστη	Email	Επίπεδο Χρήστη	Ημερομηνία δημιουργίας	Ημερομηνία τροποποίησης	Τροποποίηση Στοιχείων	Διαγραφή Χρήστη
Dimitris Boudas	dimitrisboudas@hotmail.com	admin	20/6/2016   Ώρα: 12:36:35 PM		<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Dimitris Boudas	dimitrisboudas@gmail.com	admin	20/6/2016   Ώρα: 12:37:07 PM	20/6/2016   Ώρα: 12:46:58 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>

### Τεχνικοί Συστήματος

Όνοματεπώνυμο Χρήστη	Email	Επίπεδο Χρήστη	Ημερομηνία δημιουργίας	Ημερομηνία τροποποίησης	Τροποποίηση Στοιχείων	Διαγραφή Χρήστη
Technikos 1	technikos1@yahoo.gr	tech	20/6/2016   Ώρα: 12:38:13 PM	20/6/2016   Ώρα: 12:47:05 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 2	technikos2@yahoo.gr	tech	20/6/2016   Ώρα: 12:38:42 PM	20/6/2016   Ώρα: 12:47:12 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 3	technikos3@gmail.gr	tech	20/6/2016   Ώρα: 12:39:13 PM	20/6/2016   Ώρα: 12:47:19 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 4	technikos4@gmail.gr	tech	20/6/2016   Ώρα: 12:39:41 PM	20/6/2016   Ώρα: 12:47:26 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 5	technikos5@gmail.com	tech	20/6/2016   Ώρα: 12:40:16 PM	20/6/2016   Ώρα: 12:47:34 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>

### Εγγεγραμμένοι Χρήστες Συστήματος

Όνοματεπώνυμο Χρήστη	Email	Επίπεδο Χρήστη	Ημερομηνία δημιουργίας	Ημερομηνία τροποποίησης	Τροποποίηση Στοιχείων	Διαγραφή Χρήστη
User 1	user1@gmail.com	user	20/6/2016   Ώρα: 12:40:51 PM		<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
User 2	user2@gmail.com	user	20/6/2016   Ώρα: 12:41:09 PM		<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
User 3	user3@gmail.com	user	20/6/2016   Ώρα: 12:41:34 PM		<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>

**Εικόνα 43:** Πίνακες χρηστών συστήματος

Όπως φαίνεται στην παραπάνω εικόνα, ο διαχειριστής έχει πρόσβαση στα στοιχεία όλων των χρηστών (Όνοματεπώνυμο, Email, Επίπεδο χρήστη). Ταυτόχρονα επισημαίνονται η ημερομηνία και η ώρα κατά την οποία δημιουργήθηκε η εγγραφή ενός χρήστη, καθώς και η ημερομηνία/ώρα από την τελευταία φορά που τροποποιήθηκε κάποιος χρήστης.

Όταν κρίνεται αναγκαίο, ο διαχειριστής μπορεί να πατήσει το κουμπί «Επεξεργασία» και να πραγματοποιήσει τροποποιήσεις στα στοιχεία ενός χρήστη. Η φόρμα επεξεργασίας ενός χρήστη παρουσιάζεται στη συνέχεια (Εικόνα 44).

## Επεξεργασία Χρήστη

[🏠 Πίνακας Ελέγχου](#) / [👤 Επεξεργασία Χρήστη](#)

---

**👤 Επεξεργασία Χρήστη**

ID Χρήστη: 8

Όνοματεπώνυμο:

Email Χρήστη:

Επίπεδο Χρήστη:

[Συμπληρώστε όλα τα πεδία!](#)

**Εικόνα 44:** Φόρμα επεξεργασίας χρήστη



Μόλις πατηθεί το κουμπί της επεξεργασίας η παραπάνω φόρμα εμφανίζεται στο διαχειριστή. Τα πεδία είναι ήδη συμπληρωμένα με τα δεδομένα στοιχεία και μπορούν να επεξεργαστούν μέσω εισαγωγής κειμένου. Να σημειωθεί πως η επεξεργασία ενός email δεν αποτελεί πάντα μία πολύ καλή επιλογή, αλλά βρίσκεται μέσα στις επιλογές της τροποποίησης και μπορεί να υποστεί αλλαγές μόνο μετά από αίτημα του δικαιούχου του λογαριασμού μετά από επικοινωνία με το διαχειριστή. Στο τέλος της φόρμας αυτής, μπορεί να πραγματοποιηθεί και αλλαγή επιπέδου σε ένα χρήστη συμπληρώνοντας ένα από τα επίπεδα τα οποία συμβαδίζουν με τη δομή της βάσης δεδομένων που σχεδιάστηκε για το σύστημα. Τέλος, ο διαχειριστής μπορεί να διαγράψει ένα χρήστη από το σύστημα, όταν αυτό κριθεί αναγκαίο, πατώντας το κουμπί «Διαγραφή». Ο χρήστης διαγράφεται και μαζί του διαγράφονται πιθανές μετρήσεις που είναι καταγεγραμμένες στο λογαριασμό του. Συνεχίζοντας στη λίστα των επιλογών του διαχειριστή, σχεδιάστηκαν και υλοποιήθηκαν 2 ακόμα πολύ σημαντικές λειτουργίες. Στην πρώτη συναντάμε την καρτέλα «Μετρήσεις Τεχνικών». Σε αυτό το πεδίο, ο διαχειριστής εποπτεύει τις μετρήσεις που καταγράφονται στο σύστημα μέσω των τεχνικών του και μπορεί να επέμβει και να τροποποιήσει τις μετρήσεις αυτές. Στην παρακάτω εικόνα (Εικόνα 45) παρουσιάζεται το περιβάλλον των μετρήσεων τεχνικών που σχεδιάστηκε.

### Μετρήσεις Τεχνικών

🔍 Πίνακας Ελέγχου / 📄 Μετρήσεις Τεχνικών

#### Τεχνικοί Συστήματος

Όνοματεπώνυμο Χρήστη	Email	Επίπεδο Χρήστη	Ημερομηνία Δημιουργίας
Technikos 1	technikos1@yahoo.gr	tech	20/6/2016   Πρα: 12:38:13 PM
Technikos 2	technikos2@yahoo.gr	tech	20/6/2016   Πρα: 12:38:42 PM
Technikos 3	technikos3@gmail.gr	tech	20/6/2016   Πρα: 12:39:13 PM
Technikos 4	technikos4@gmail.gr	tech	20/6/2016   Πρα: 12:39:41 PM
Technikos 5	technikos5@gmail.com	tech	20/6/2016   Πρα: 12:40:16 PM

#### Μετρήσεις Τεχνικών

Όνοματεπώνυμο Χρήστη	ID Τεχνικού	Μέτρηση	Είδος Μέτρησης	Ημερομηνία Δημιουργίας	Ημερομηνία τροποποίησης	Επεξεργασία μέτρησης	Διαγραφή μέτρησης
Technikos 1	3	12345687	Electricity	2016-06-22 17:55:51	2016-06-22 17:55:51	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 1	3	12344321	Natural Gas	2016-06-22 17:56:28	2016-06-22 17:56:28	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 1	3	12345678	Water	2016-06-22 17:56:38	2016-06-22 17:56:38	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 1	3	12345678	Extra	2016-06-20 13:05:42	0000-00-00 00:00:00	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
Technikos 2	4	909080706	Electricity	2016-06-22 17:56:56	2016-06-22 17:56:56	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>

**Εικόνα 45:** Ο πίνακας μετρήσεων των τεχνικών

Με την είσοδό του εδώ, ο διαχειριστής βλέπει συνοπτικά τη λίστα των ενεργών τεχνικών και στη συνέχεια τις μετρήσεις που αντιστοιχούν στον καθένα. Στον πίνακα των μετρήσεων εμφανίζονται το ονοματεπώνυμο κάθε τεχνικού, το ID που τους έχει δοθεί και πληροφορίες σχετικές με τις μετρήσεις τους. Στις πληροφορίες αυτές εμφανίζονται κατά σειρά :

- Τα αριθμητικά στοιχεία της μέτρησης
- Το είδος που ανήκει η μέτρηση που έγινε
- Η ημερομηνία και η ακριβής ώρα που πραγματοποιήθηκε η καταχώρηση
- Η ημερομηνία και η ακριβής ώρα από την τελευταία φορά που κάποια μέτρηση υπέστη επεξεργασία (αν δεν υπάρχει επεξεργασία σε κάποιο πεδίο, τότε εμφανίζεται ως μηδενικό πεδίο).

Ταυτόχρονα, μαζί με την εποπτεία των μετρήσεων δίνεται η δυνατότητα επεξεργασίας τους με την προσθήκη 2 κουμπιών σε κάθε γραμμή του πίνακα. Η τροποποίηση μετρήσεων λειτουργεί με την ίδια λογική που λειτουργεί και η επεξεργασία των χρηστών του συστήματος. Ο διαχειριστής μπορεί να πατήσει το κουμπί «Επεξεργασία» και να αλλάξει κάποια μέτρηση ή την κατηγορία στην οποία ανήκει μία μέτρηση, όταν αυτό κριθεί αναγκαίο. Στην παρακάτω εικόνα (Εικόνα 46) παρουσιάζεται η φόρμα επεξεργασίας μίας μέτρησης.

## Επεξεργασία Μέτρησης

Πίνακας Ελέγχου / Επεξεργασία Μέτρησης

Επεξεργασία Μέτρησης

ID Μέτρησης: 5

Μέτρηση: 12345687

Είδος Μέτρησης: Electricity

[Συμπληρώστε όλα τα πεδία!](#)

Αποθήκευση

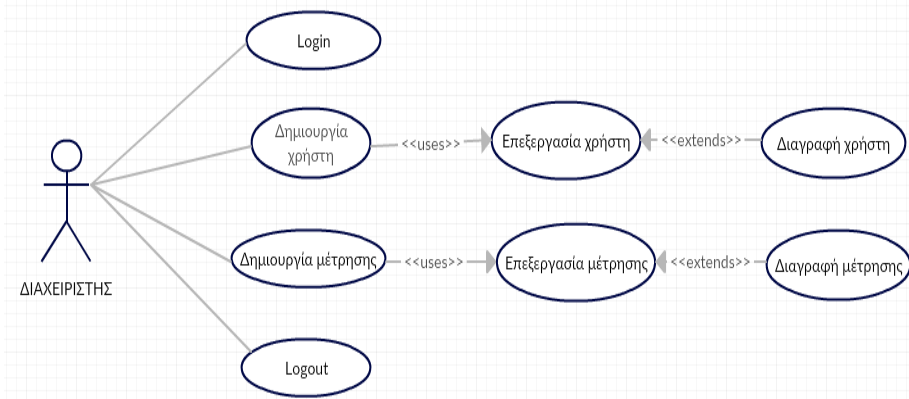
**Εικόνα 46:** Φόρμα επεξεργασίας Μέτρησης

Με την ίδια λογική λοιπόν, η φόρμα εμφανίζεται συμπληρωμένη με τα στοιχεία που έχει η μέτρηση, και ο διαχειριστής μπορεί να τα επεξεργαστεί μέσω

κειμένου. Επιπλέον μπορεί να αλλάξει κατηγορία στη μέτρηση πληκτρολογώντας ένα διαφορετικό είδος ,από αυτά που υπάρχουν, στο αντίστοιχο πεδίο. Τέλος, μπορεί να πραγματοποιηθεί διαγραφή κάποιας μέτρησης με το πάτημα του κουμπιού «Διαγραφή» που βρίσκεται σε κάθε καταχώρηση.

Στο ίδιο μοτίβο, υλοποιήθηκε άλλος ένας πίνακας που είναι διαθέσιμος στο διαχειριστή, στον οποίο μπορεί να δει, να επεξεργαστεί και να διαγράψει καταχωρημένες μετρήσεις από τους χρήστες του κατώτερου επιπέδου. Η φιλοσοφία με την οποία αναπτύχθηκε είναι ίδια με αυτήν που περιγράψαμε για τις μετρήσεις των τεχνικών.

Με βάση όλη την παραπάνω περιγραφή, παρουσιάζεται παρακάτω το διάγραμμα use-case του διαχειριστή (Εικόνα 47).



**Εικόνα 47:** Διάγραμμα use-case Διαχειριστή

Στη συνέχεια θα γίνει αναφορά στις λειτουργίες των υπόλοιπων επιπέδων των χρηστών μέσω του ιστοχώρου που αναπτύχθηκε.

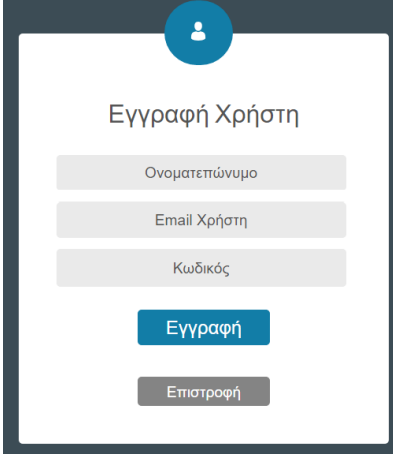
### 4.3.3 Λειτουργίες χρηστών κατώτερων επιπέδων

Οι λειτουργίες των υπόλοιπων επιπέδων των χρηστών αφορούν κατά βάση την δημιουργία απλού λογαριασμού για την εποπτεία του προσωπικού τους προφίλ, και την καταγραφή μετρήσεων.

Αρχικά, για να μπορέσει ένας χρήστης να έχει τη δυνατότητα να χρησιμοποιήσει την εφαρμογή θα πρέπει αρχικά να δημιουργήσει ένα

λογαριασμό στη σελίδα. Στις καρτέλες της ιστοσελίδας ο χρήστης μπορεί να βρει τις απαραίτητες πληροφορίες επιλέγοντας την καρτέλα «Η εφαρμογή».

Ο χρήστης μπορεί να επιλέξει την επιλογή «Εγγραφή», από την λίστα της αρχικής οθόνης της ιστοσελίδας. Με την πραγματοποίηση αυτή της επιλογής, εμφανίζεται στο χρήστη η φόρμα εγγραφής που αναπτύχθηκε. Στην παρακάτω εικόνα (Εικόνα 48) εμφανίζεται η φόρμα της εγγραφής νέου χρήστη.

The image shows a mobile application registration screen. At the top, there is a blue circular icon with a white person silhouette. Below it, the title 'Εγγραφή Χρήστη' is centered. There are three light gray input fields stacked vertically, labeled 'Όνοματεπώνυμο', 'Email Χρήστη', and 'Κωδικός'. Below these fields is a blue button with the text 'Εγγραφή'. At the bottom, there is a gray button with the text 'Επιστροφή'.

**Εικόνα 48:** Φόρμα εγγραφής χρήστη

Στη φόρμα αυτή δίνεται η προτροπή στο χρήστη να εισάγει τα προσωπικά του στοιχεία μαζί με έναν κωδικό, για να δημιουργήσει το λογαριασμό του. Είναι απαραίτητο να συμπληρωθούν όλα τα πεδία και γι αυτό υπάρχει σχετικός έλεγχος. Όταν ο έλεγχος ολοκληρωθεί επιτυχώς και η εγγραφή πραγματοποιηθεί, εμφανίζεται στο χρήστη το μήνυμα «Ο λογαριασμός με (Όνοματεπώνυμο και Email χρήστη) δημιουργήθηκε επιτυχώς».

Για να αποκτήσει πρακτικό νόημα η εγγραφή του χρήστη και να καταχωρηθεί στο σύστημα ώστε να μπορεί να χρησιμοποιήσει την εφαρμογή για την καταγραφή μετρήσεων, με την ολοκλήρωση της εγγραφής του θα πρέπει να ενημερώσει το διαχειριστή του συστήματος μέσω της φόρμας επικοινωνίας, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 49).

Στείλτε μας το μήνυμά σας

Όνοματεπώνυμο:

Τηλέφωνο:

Διεύθυνση Email:

Μήνυμα:

[Αποστολή Μηνύματος](#)

**Εικόνα 49:** Φόρμα επικοινωνίας με διαχειριστή

Στη φόρμα αυτή, θα πρέπει να συμπληρώσει τα στοιχεία του και να αναφέρει τη δημιουργία λογαριασμού, τη διεύθυνση του και το αίτημα του για αποστολή της εφαρμογής. Στη συνέχεια ο διαχειριστής αναλαμβάνει να προσθέσει τα στοιχεία του χρήστη (εάν δεν έχει πραγματοποιήσει απλή εγγραφή από μόνος του) και να του αποστείλει το σύνδεσμο για την εφαρμογή, έτσι ώστε ο χρήστης να μπορεί να πραγματοποιήσει καταγραφές μετρήσεων.

Αφού ολοκληρωθούν τα βήματα αυτά, ο χρήστης μπορεί πλέον να χρησιμοποιήσει την εφαρμογή για την καταχώρηση μετρήσεων στο λογαριασμό του. Σε περίπτωση που ο χρήστης επιθυμεί να πραγματοποιεί μετρήσεις ως τεχνικός, θα πρέπει να αποστείλει σχετικό αίτημα στο διαχειριστή μέσω της φόρμας επικοινωνίας.

Για να επισκεφτεί το προσωπικό του προφίλ ο χρήστης, πρέπει πρώτα να πραγματοποιήσει είσοδο μέσω της φόρμας εισόδου που αναπτύχθηκε για την ιστοσελίδα. Η φόρμα αυτή φαίνεται στην επόμενη εικόνα (Εικόνα 50).

**Εικόνα 50:** Η φόρμα εισόδου χρήστη

Ο χρήστης εισάγει τα στοιχεία του στη φόρμα για να πραγματοποιήσει είσοδο. Είναι απαραίτητο να συμπληρωθούν και τα 2 πεδία και γι αυτό υπάρχει σχετικός έλεγχος από τη φόρμα. Αφού εισαχθούν τα σωστά στοιχεία, προβάλλεται το αντίστοιχο μήνυμα επιτυχίας και η φόρμα στέλνει το χρήστη στο προσωπικό του προφίλ.

Μόλις ο χρήστης αρχίσει να πραγματοποιεί μετρήσεις, το προσωπικό του προφίλ ενημερώνεται αυτόματα και εμφανίζονται σε μορφή πίνακα οι μετρήσεις που πραγματοποιεί. Εκεί ο χρήστης μπορεί να επιβλέπει τις λεπτομέρειες των μετρήσεων του και να πραγματοποιεί τροποποιήσεις. Στην παρακάτω εικόνα (Εικόνα 51) εμφανίζεται το προφίλ ενός χρήστη.

Αρχική σελίδα DiPiOCRa

Σχετικά Η εφαρμογή Επικοινωνία Έχετε συνδεθεί ως : **Τεχνικός 1**

### Κεντρικό Μενού ΤΕΧΝΙΚΟΥ

Αρχική / Κεντρικό μενού τεχνικού

Δεδομένα Τεχνικού

Μέτρηση	Ημερομηνία δημιουργίας	Είδος μέτρησης	Ημερομηνία τροποποίησης	Επεξεργασία Μέτρησης	Διαγραφή Μέτρησης
12345687	Wednesday 22/6/2016   Ώρα: 05:55:51 PM	Electricity	Wednesday 22/6/2016   Ώρα: 05:55:51 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
12344321	Wednesday 22/6/2016   Ώρα: 05:56:28 PM	Natural Gas	Wednesday 22/6/2016   Ώρα: 05:56:28 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
12345678	Wednesday 22/6/2016   Ώρα: 05:56:38 PM	Water	Wednesday 22/6/2016   Ώρα: 05:56:38 PM	<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
12345678	Monday 20/6/2016   Ώρα: 01:05:42 PM	Extra		<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>
9898900000	Wednesday 22/6/2016   Ώρα: 06:09:03 PM	Electricity		<a href="#">Επεξεργασία</a>	<a href="#">Διαγραφή</a>

Πελάτες Τεχνικού

Όνοματεπώνυμο	Email	Διεύθυνση
User 1	user1@gmail.com	Street One
User 2	user2@gmail.com	Street One
User 3	user3@gmail.com	Street One

COPYRIGHT © Boudas Dimitris  
SUPERVISOR DR. Minas Dasygenis

**Εικόνα 51:** Το προφίλ χρήστη

Ο εγγεγραμμένος χρήστης (τεχνικός) μπορεί να παρακολουθεί τις μετρήσεις του από τον παραπάνω πίνακα. Σε αυτόν προβάλλονται πληροφορίες σχετικές με τις μετρήσεις που έχει κάνει, καθώς και το είδος της μέτρησης, ακολουθούμενο από ημερομηνίες δημιουργίας και τροποποίησης. Ο χρήστης μπορεί να επεξεργαστεί και να διαγράψει τις μετρήσεις του με παρόμοιο τρόπο με αυτόν του διαχειριστή.

Ακριβώς κάτω από τις μετρήσεις του τεχνικού υπάρχουν καταχωρημένοι σε έναν πίνακα οι εγγεγραμμένοι χρήστες (πελάτες) τους οποίους έχει οριστεί να εξυπηρετεί ο συγκεκριμένος τεχνικός. Η ανάθεση πελατών γίνεται από το

διαχειριστή και βασίζεται πάντα σε κατηγοριοποίηση ανάλογα με τη διεύθυνση των πελατών.

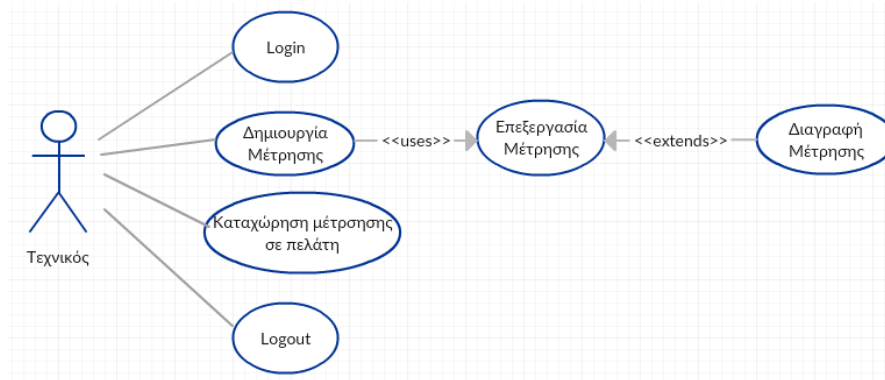
Για τους πελάτες που έχουν ανατεθεί στον τεχνικό, υπάρχει σε μορφή πίνακα μέσα στο προφίλ του, ένας κατάλογος που του εμφανίζει λεπτομερώς τις μετρήσεις που έχουν πραγματοποιήσει οι πελάτες που του έχουν ανατεθεί. Ο σχετικός πίνακας εμφανίζεται στην επόμενη εικόνα (Εικόνα 52).

Δεδομένα Πελατών

Όνοματεπώνυμο	ID Πελάτη	Μέτρηση Πελάτη	Είδος Μέτρησης	ID Μέτρησης	Ημερομηνία Δημιουργίας	Ημερομηνία Τροποποίησης	Επεξεργασία Μέτρησης	Διαγραφή Μέτρησης
User 1	8	9823888	Electricity	1	2016-06-22 17:57:24	2016-06-22 17:57:24	Επεξεργασία	Διαγραφή
User 1	8	9823486	Natural Gas	2	2016-06-22 17:57:31	2016-06-22 17:57:31	Επεξεργασία	Διαγραφή
User 1	8	9823486	Water	3	2016-06-22 17:57:39	2016-06-22 17:57:39	Επεξεργασία	Διαγραφή
User 1	8	9823486	Extra	4	2016-06-20 13:03:44	0000-00-00 00:00:00	Επεξεργασία	Διαγραφή
User 2	9	1112222	Electricity	13	2016-06-22 17:57:46	2016-06-22 17:57:46	Επεξεργασία	Διαγραφή
User 2	9	1112222	Natural Gas	14	2016-06-22 17:57:54	2016-06-22 17:57:54	Επεξεργασία	Διαγραφή
User 2	9	1112222	Water	15	2016-06-22 17:58:00	2016-06-22 17:58:00	Επεξεργασία	Διαγραφή
User 2	9	1112222	Extra	16	2016-06-20 13:07:01	0000-00-00 00:00:00	Επεξεργασία	Διαγραφή

**Εικόνα 52:** Πίνακας πελατών τεχνικού

Στον πίνακα αυτόν εμφανίζονται όλες οι απαραίτητες λεπτομέρειες που πρέπει να γνωρίζει ο τεχνικός για τις μετρήσεις των πελατών του. Επιπλέον έχει και εδώ το δικαίωμα επεξεργασίας και διαγραφής των μετρήσεων χρήστη όπου αυτό κρίνεται αναγκαίο. Το διάγραμμα use-case ενός τεχνικού παρουσιάζεται παρακάτω (Εικόνα 53).



**Εικόνα 53:** Διάγραμμα use-case τεχνικού

Επίσης να σημειωθεί πως, για τους εγγεγραμμένους χρήστες κατώτατου επιπέδου (απλοί χρήστες), αναπτύχθηκε περιβάλλον λογαριασμού, που

ακολουθεί την ίδια φιλοσοφία με αυτό των τεχνικών. Βέβαια όπως είναι φυσικό, ο απλός χρήστης έχει πρόσβαση και δικαιώματα επεξεργασίας μόνο στις προσωπικές του καταχωρήσεις. Οι καταχωρήσεις του απλού χρήστη θα ελεγχθούν από τον τεχνικό που του αντιστοιχεί και θα επεξεργαστούν ώστε να ανταποκρίνονται στην πραγματικότητα.

#### 4.4 Στοιχεία σύνδεσης διακομιστή-εφαρμογής-ιστοσελίδας

Στην παράγραφο αυτή θα γίνει μία σύντομη αναφορά στον κώδικα που δημιουργήθηκε ώστε να συνεργάζονται σωστά η εφαρμογή Android και η ιστοσελίδα, με το διακομιστή. Μέσα από την ανάλυση της υλοποίησης, έγινε αναφορά σε αιτήματα τύπου POST προς το διακομιστή. Για την πλήρη κατανόηση του συστήματος θα εξηγήσουμε αυτούς τους συνδέσμους.

Η εφαρμογή Android για να πραγματοποιήσει τις κυριότερες λειτουργίες της, επικοινωνεί με τη βάση δεδομένων μέσω του διακομιστή. Για την εγγραφή ενός χρήστη, για την είσοδο ενός χρήστη, για την καταχώρηση μετρήσεων στη βάση, αλλά και για την παρουσίαση αποτελεσμάτων από τη βάση στην οθόνη της εφαρμογής, δημιουργήθηκε κώδικας σε γλώσσα PHP. Ο κώδικας αυτός επικοινωνεί με τη βάση και «απαντάει» στα ερωτήματα της εφαρμογής, αφού αρχικά γίνουν οι απαραίτητες μετατροπές για τη χρήση αντικειμένων JSON. Για να δημιουργηθεί λοιπόν αυτή η «γέφυρα» μεταξύ βάσης και εφαρμογής δημιουργήθηκαν 4 php αρχεία, για κάθε μία από τις διεργασίες της εφαρμογής. Η αντιστοίχιση τους είναι η εξής :

- Τα POST της εφαρμογής στο σύνδεσμο *URL\_REGISTER* αντιστοιχούν στο αρχείο του διακομιστή *android/register.php*
- Τα POST της εφαρμογής στο σύνδεσμο *URL\_LOGIN* αντιστοιχούν στο αρχείο του διακομιστή *android/login.php*
- Τα POST της εφαρμογής στο σύνδεσμο *insertUrl* αντιστοιχούν στο αρχείο του διακομιστή *android/insertData.php*
- Τα POST της εφαρμογής στο σύνδεσμο *url* αντιστοιχούν στο αρχείο του διακομιστή *android/measures.php*



Στον κατάλογο android του διακομιστή, υπάρχουν επίσης και οι λειτουργίες και συναρτήσεις τις οποίες καλούν τα παραπάνω αρχεία.

Οι μέθοδοι και οι συναρτήσεις που χρησιμοποιήθηκαν για την σωστή λειτουργία της εφαρμογής με τη βάση, είναι παρόμοιες με αυτές που χρησιμοποιήθηκαν για τη σύνδεση και περάτωση ενεργειών της ιστοσελίδας με τη βάση. Για επιπλέον λειτουργίες και ενέργειες που προσφέρει το περιβάλλον της ιστοσελίδας που αναπτύχθηκε, δημιουργήθηκαν επιπλέον συναρτήσεις βασισμένες στην ίδια φιλοσοφία με τις προηγούμενες και προσθέτοντας τα απαραίτητα επιπλέον στοιχεία.

#### **4.5 Στοιχεία κώδικα**

Για την υλοποίηση του ιστοχώρου δημιουργήθηκαν 40 αρχεία php συνολικού μεγέθους 171KB και 6397 γραμμών κώδικα. Επιπλέον δημιουργήθηκαν 5 αρχεία JavaScript μεγέθους 236 KB και 922 γραμμές κώδικα. 4 αρχεία css μεγέθους 90KB και 1012 γραμμές κώδικα. Επιπλέον προστέθηκαν 9 εικόνες τύπου PNG συνολικού μεγέθους 1815KB.

Για την υλοποίηση της εφαρμογής Android δημιουργήθηκαν 10 αρχεία java συνολικού μεγέθους 40,1KB και 1545 γραμμών κώδικα. Επίσης 8 αρχεία xml συνολικού μεγέθους 15,9KB και 481 γραμμών κώδικα. Επίσης προστέθηκαν 4KB αρχείων gradle για την εξωτερική βιβλιοθήκη tess-two, καθώς και 2,19MB με τα γλωσσικά στοιχεία αναγνώρισης του αρχείου eng.traineddata.

#### **4.6 Ασφάλεια συστήματος**

Η ασφάλεια συστημάτων λογισμικού είναι ένας πάρα πολύ σημαντικός παράγοντας στην σωστή λειτουργία όλων των ενοτήτων που υλοποιήθηκαν στην παρούσα διπλωματική εργασία. Δεδομένου ότι στο σύστημα που αναπτύχθηκε υπάρχει συνεχής επικοινωνία εφαρμογής, βάσης και ιστοσελίδας, κρίθηκε αναγκαίο να υλοποιηθούν τα απαραίτητα μέτρα ασφαλείας σε κάθε περίπτωση.

Ξεκινώντας από τα βασικά σημεία, στη βάση δεδομένων που αναπτύχθηκε, ο κάθε χρήστης αποθηκεύει ευαίσθητα προσωπικά δεδομένα. Για το λόγο αυτό υπάρχει το απαραίτητο πεδίο εισαγωγής κωδικού πρόσβασης. Βέβαια ένας απλός κωδικός θα ήταν πολύ εύκολη υπόθεση να βρεθεί από μία επίθεση κακόβουλου χρήστη. Έτσι λοιπόν ο προσωπικός κωδικός των χρηστών, μέσα από τις λειτουργίες δημιουργίας λογαριασμού, από την εφαρμογή, από την ιστοσελίδα αλλά ακόμα και από την προσθήκη χρήστη χειροκίνητα από το διαχειριστή, δέχεται επεξεργασία που έχει ως αποτέλεσμα την παροχή πολύ υψηλού επιπέδου ασφάλειας. Συγκεκριμένα κάθε κωδικός που εισάγεται περνά από μία ειδική συνάρτηση. Στη συνάρτηση αυτή τοποθετείται αρχικά μία επεξεργασία sha1 και αφού δημιουργηθεί το κατάλληλο εργαλείο κρυπτογράφησης (salt), ο ήδη διαμορφωμένος κωδικός περνά σε επεξεργασία κωδικοποίησης base64. Η base64 θεωρείται μία από τις ισχυρότερες κωδικοποιήσεις για αυτό κι επιλέχθηκε στη συγκεκριμένη λειτουργία. Με ακόμα περισσότερη λεπτομέρεια στην ασφάλεια, το «εργαλείο» κρυπτογράφησης που αναφέρθηκε, αποθηκεύεται ξεχωριστά από τον κρυπτογραφημένο κωδικό και χρησιμοποιείται με συγκεκριμένο τρόπο μέσα από συνάρτηση ειδικά φτιαγμένη για αυτό. Το αποτέλεσμα της διεργασίας που μόλις αναφέρθηκε είναι η καταχώρηση ενός πολύ ισχυρού κωδικού, που παρέχει υψηλά επίπεδα ασφαλείας στους χρήστες του.

Ένα άλλο σημαντικό κομμάτι στην ασφάλεια του συστήματος είναι το γεγονός ότι καμία ενέργεια δεν μπορεί να πραγματοποιηθεί από μη εγγεγραμμένους χρήστες. Η εφαρμογή δεν επιτρέπει την πραγματοποίηση ενεργειών αν δεν έχει γίνει είσοδος μέσω ενός καταχωρημένου λογαριασμού. Επίσης, στην ιστοσελίδα που δημιουργήθηκε, οι επισκέπτες δεν μπορούν να συνδεθούν ή να δουν κανένα από τα λειτουργικά στοιχεία αν δεν έχουν ένα καταχωρημένο λογαριασμό. Σε αυτό συμβάλει και το γεγονός ότι ο μόνος τρόπος για να εγκαταστήσει κάποιος την εφαρμογή είναι να του δοθεί ο αντίστοιχος σύνδεσμος έπειτα από επικοινωνία με το διαχειριστή.

Ακόμα, η άμεση πρόσβαση μέσω URL στα αρχεία της σελίδας είναι μπλοκαρισμένη. Έχουν δημιουργηθεί επιπλέον αρχεία και έλεγχοι, έτσι ώστε

οποιοσδήποτε μη εγκεκριμένος χρήστης προσπαθήσει να εισέλθει σε σελίδα για την οποία δεν έχει δικαιώματα, να απομακρύνεται κατευθείαν από την σελίδα αυτή και να επιστρέφει στην αρχική σελίδα.

Σε όλες τις φόρμες στις οποίες καταχωρούνται και αποστέλλονται στοιχεία προς τη βάση δεδομένων, χρησιμοποιούνται μέθοδοι φιλτραρίσματος για την αποφυγή εισαγωγής ανεπιθύμητων χαρακτήρων σε οποιοδήποτε σημείο. Η συνάρτηση που χρησιμοποιήθηκε είναι η `filter_var` [19], και εισήχθη στον κώδικα με τη χρήση των ανάλογων `filter` και `flags` για κάθε περίπτωση. Ένα παράδειγμα της χρήσης τους εμφανίζεται στην εικόνα που ακολουθεί (Εικόνα 54).

```
$name = filter_var($_POST['name'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW);
$email = filter_var($_POST['email'], FILTER_SANITIZE_EMAIL);
$password = filter_var($_POST['password'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW);
$role = filter_var($_POST['role'], FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_LOW);
```

**Εικόνα 54:** Χρήση φίλτρων μέσω της συνάρτησης `filter_var`

Η συνάρτηση λειτουργεί με βάση τον τύπο `filter_var(var, filtername, options)`. Το `var` αντιστοιχεί στην μεταβλητή η οποία θα περάσει από φίλτρο και είναι απαραίτητη προϋπόθεση να είναι συμπληρωμένη. Το `filtername` αντιστοιχεί στο ID ή το όνομα του φίλτρου που θα χρησιμοποιηθεί όπως για παράδειγμα το «`FILTER_SANITIZE_EMAIL`» της εικόνας. Η επιλογή του είναι προαιρετική. Τέλος το `options` αποτελεί την διευκρίνιση ενός η περισσοτέρων `flags` που θα χρησιμοποιηθούν στο φίλτρο και είναι επίσης προαιρετικό. Κάθε ID φίλτρου λειτουργεί με συγκεκριμένα `flags`, που βρίσκονται καταχωρημένα στο εγχειρίδιο της PHP.

Συνεχίζοντας στον τομέα της ασφάλειας, σε κάθε ερώτημα ή διεργασία που αλληλεπιδρά με τη βάση δεδομένων χρησιμοποιήθηκε η επέκταση MySQLi [20] (αναβάθμιση έκδοσης MySQL). Τα κυριότερα χαρακτηριστικά της επέκτασης αυτής είναι :

- Αντικειμενοστραφές περιβάλλον
- Υποστήριξη προετοιμασμένων statements
- Υποστήριξη πολλαπλών statements
- Υποστήριξη transactions

- Ενισχυμένες δυνατότητες debugging
- Ενσωματωμένη υποστήριξη του διακομιστή

Το αντικειμενοστραφές περιβάλλον σημαίνει τη δυνατότητα χρήσης αντικειμένων δεδομένων PHP (PHP Data Objects) ή απλά PDO. Επίσης τα προετοιμασμένα statements συμβάλουν σημαντικά στην ασφάλεια απέναντι σε sql injections δεδομένου ότι δεν εμφανίζουν τα δεδομένα και δεν εκτελούνται άμεσα, παρά μόνο όταν ζητηθεί από σχετική κλήση. Η μορφή ενός statement περιγράφεται από τον τύπο : "INSERT INTO tablename (value1, value2) VALUES (?, ?)". Στη θέση των «(?,?)» πραγματοποιείται bind\_param των τιμών που επιθυμεί ο χρήστης, σε διαφορετική γραμμή του κώδικα.

Με βάση λοιπόν όλες αυτές τις δικλίδες ασφαλείας, όταν κάποιος χρήστης εισέρχεται επιτυχώς στο λογαριασμό του, δημιουργούνται κατάλληλες μεταβλητές session. Στις μεταβλητές αυτές αποθηκεύονται απαραίτητες πληροφορίες, σχετικές με το προφίλ του χρήστη. Με βάση αυτές ο χρήστης κατευθύνεται στο κατάλληλο προφίλ επιπέδου και παραμένει αυστηρά σε αυτό, χωρίς να μπορεί να επεξεργαστεί τίποτα περισσότερο από τις δυνατότητες που του παρέχει το επίπεδό του. Όταν ο χρήστης πραγματοποιήσει τις ενέργειες που επιθυμεί, μπορεί να εξέλθει από το λογαριασμό του. Με την πραγματοποίηση αυτής της λειτουργίας ενεργοποιείται σχετικός κώδικας που διαγράφει όλα τα στοιχεία session που είναι αποθηκευμένα, αποτρέποντας έτσι την αναπαραγωγή τους από κάποιο άλλο χρήστη του ίδιου υπολογιστή.

Σε παρόμοιο μοτίβο, όπως αναφέρθηκε νωρίτερα, η εφαρμογή Android διευκολύνει μεν ένα χρήστη αποθηκεύοντας τα στοιχεία του, αλλά όταν ο χρήστης πραγματοποιήσει έξοδο, όλα τα δεδομένα και τα στοιχεία διαγράφονται, καθιστώντας έτσι την εφαρμογή ασφαλή για την χρήση πολλαπλών χρηστών.

Επιπλέον πραγματοποιήθηκαν ενέργειες για την επικοινωνία της εφαρμογής Android με την ιστοσελίδα μέσω HTTPS και SSL [21], έτσι ώστε να υπάρχει ακόμα μεγαλύτερη προσοχή στην ασφάλεια. Για την πραγματοποίηση του μέτρου αυτού υπήρξαν απαραίτητες κάποιες βασικές αλλαγές στον κώδικα του Android μετά από κάποια updates της έκδοσης του Android Studio.

Επίσης πρέπει να σημειωθεί πως, με την εισαγωγή αδειών στο Manifest του Android, υπάρχουν οι απαραίτητες ενδείξεις σε οποιοδήποτε χρήστη επιχειρήσει να εγκαταστήσει την εφαρμογή, όπως για παράδειγμα το γεγονός ότι η εφαρμογή θα χρησιμοποιήσει την κάμερα της συσκευής, σύνδεση στο internet, πιθανή χρήση εξωτερικών χώρων αποθήκευσης μέσων και άλλα.

## 4.7 Δοκιμές

Δεδομένου ότι στην παρούσα διπλωματική αναπτύχθηκαν 3 σημαντικά μέρη, στην πορεία, και ειδικότερα μετά την ολοκλήρωση τους πραγματοποιήθηκαν δοκιμές για να επαληθευτεί η σωστή λειτουργία όλων των τμημάτων του συστήματος που αναπτύχθηκε. Στον τομέα της εφαρμογής Android, πραγματοποιήθηκαν πάρα πολλές δοκιμές μέσω του Android Virtual Device [22] (AVD) που προσφέρει το Android Studio, μέχρι το σημείο όπου ολοκληρώθηκε και το κομμάτι της λειτουργίας της κάμερας της συσκευής. Από το σημείο εκείνο και έπειτα χρησιμοποιήθηκε κανονική συσκευή smartphone, εγκαθιστώντας την εφαρμογή και πραγματοποιώντας δοκιμαστικές λήψεις φωτογραφιών πάνω σε κείμενο. Γενικότερα, το πλάνο των δοκιμών που έλαβαν χώρα πραγματοποιήθηκε ως εξής :

Πραγματοποιήθηκαν ενέργειες για τον έλεγχο της σωστής λειτουργίας όλων των δραστηριοτήτων και των κουμπιών της εφαρμογής. Κάθε φορά που γινόταν εκτέλεση κάποιας λειτουργίας υπήρχε αντίστοιχο μήνυμα σφάλματος η επιτυχίας στην οθόνη της κονσόλας του προσομοιωτή του Android Studio.

Έγιναν δοκιμές σχετικά με τη δημιουργία ενός λογαριασμού, με την πραγματοποίηση εισόδου σε έναν λογαριασμό και με τη διαχείριση των δεδομένων του λογαριασμού αυτού. Υπήρξαν δοκιμές σχετικά με τον έλεγχο των στοιχείων στην περίπτωση επιτυχούς σύνδεσης, εμφανίζοντας τα στοιχεία και πάλι στην οθόνη της κονσόλας του προσομοιωτή.

Ταυτόχρονα, υπήρξαν και επιτηδευμένες λανθασμένες εισαγωγές στοιχείων για να γίνει ξεκάθαρο ότι λειτουργούσαν όλες οι παράμετροι και εμφανίζονται τα κατάλληλα μηνύματα στην οθόνη, ακόμα και στις περιπτώσεις λάθους.

Έπειτα όταν ολοκληρώθηκε το κομμάτι της αναγνώρισης και της χρήσης της κάμερας, πραγματοποιήθηκαν δοκιμές σε απλό κείμενο για την αναπαραγωγή αποτελεσμάτων. Στη συνέχεια πραγματοποιήθηκαν λήψεις κατευθείαν από ρολόγια μετρητών (συγκεκριμένα ρολόγια Ηλεκτρικού ρεύματος και ύδρευσης). Επεκτείνοντας αυτές τις δοκιμές, έγιναν απόπειρες φωτογράφισης υπό δύσκολες συνθήκες για να διαπιστωθεί η χρήση των λειτουργιών της κάμερας από την εφαρμογή. Αυτές οι συνθήκες ήταν για παράδειγμα, λήψη φωτογραφιών με ελλιπή φωτισμό για την ενεργοποίηση του φλας, αλλά και λήψη από διαφορετική απόσταση για την ενεργοποίηση του auto-focus της κάμερας.

Επιπλέον με την εμφάνιση εύστοχων αποτελεσμάτων, επιχειρήσαμε την καταχώρηση τους στα σωστά πεδία, ανάλογα με το χρήστη που είναι συνδεδεμένος και πραγματοποιεί την αποστολή των δεδομένων.

Ακόμα, δοκιμάστηκε και επαληθεύτηκε μέσω της βάσης δεδομένων η σωστή εμφάνιση αποτελεσμάτων την οθόνη του χρήστη σχετικά με τις μετρήσεις που έχει κάνει μόνο αυτός.

Στο κομμάτι του ιστοχώρου, πραγματοποιήθηκαν πολλές και ποικίλες λειτουργίες, με την εφαρμογή πολλών και διαφορετικών κλήσεων προς τη βάση δεδομένων.

Για το λόγο αυτό έγιναν δοκιμές κάθε φορά που επιχειρήθηκε κάποια λειτουργία. Συγκεκριμένα έγινε έλεγχος της σωστής λειτουργίας και κλήσης όλων των καρτελών που συμπεριλαμβάνονται στην ιστοσελίδα. Στη συνέχεια πραγματοποιήθηκαν δοκιμές με τη φόρμα επικοινωνίας, αφήνοντας κενά όλα τα απαραίτητα πεδία για τον έλεγχο εμφάνισης των κατάλληλων μηνυμάτων και την αποφυγή αποστολής τους. Παράλληλα όμως δημιουργήθηκαν κανονικές φόρμες επικοινωνίας, και εστάλησαν δοκιμαστικά για την επαλήθευση της σωστής λειτουργίας και εμφάνισής τους στο email του διαχειριστή.

Στη συνέχεια πραγματοποιήθηκαν έλεγχοι και δοκιμές με την εισαγωγή σωστών και λανθασμένων στοιχείων στις φόρμες εγγραφής και εισόδου χρήστη αντίστοιχα, για την εμφάνιση των ανάλογων μηνυμάτων σε κάθε περίπτωση.

Επιπλέον πραγματοποιήθηκαν δοκιμές επαλήθευσης ανακατεύθυνσης των χρηστών στο σωστό προφίλ ανάλογα με το επίπεδο, καθώς και δοκιμές σχετικά με την ορθή αποθήκευση των μεταβλητών session.

Στο περιβάλλον κάθε επιπέδου χρήστη ξεχωριστά έγινε έλεγχος για την προβολή των σωστών πινάκων δεδομένων και ταυτόχρονα δοκιμάστηκαν οι λειτουργίες επεξεργασίας και διαγραφής στοιχείων, επαληθεύοντας τα αποτελέσματα άμεσα μέσω της βάσης δεδομένων.

Ειδικότερα στο προφίλ του διαχειριστή πραγματοποιήθηκαν δοκιμές προσθήκης και επεξεργασίας χρηστών, ελέγχοντας αν οι προσθήκες αυτές λειτουργούν σωστά, αποθηκεύοντας τα στοιχεία σωστά στις σωστές κατηγορίες. Παράδειγμα στην εισαγωγή ενός χρήστη, ελέγχθηκε αν εισάγονται κανονικά ο κρυπτογραφημένος κωδικός μαζί με το salt καθώς και η αποθήκευση timestamp ακριβείας στην λίστα ημερομηνίας και ώρας καταχώρησης.

Επίσης δοκιμάστηκαν πολλαπλές επεξεργασίες και διαγραφές χρηστών και μετρήσεων, οι οποίες επαληθεύτηκαν από τα δεδομένα που εμφανίζονταν στους πίνακες διαχείρισης αλλά και στη βάση δεδομένων.

## Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό έγινε παρουσίαση και περιγραφή της υλοποίησης του λογισμικού μέρους του συστήματος. Έγινε αναλυτική αναφορά σε όλα τα μέρη λογισμικού, στις λειτουργίες τους και στον τρόπο επικοινωνίας μεταξύ τους μέσω διαγραμμάτων, εικόνων και ανάλυσης κώδικα. Επίσης παρουσιάστηκαν παραδείγματα χρήσης με βάση τις απαιτήσεις που υπήρχαν από το σύστημα όπως αυτές είχαν οριστεί νωρίτερα. Επιπλέον έγινε αναφορά στα μέτρα ασφάλειας που χρησιμοποιήθηκαν στην εφαρμογή και παρουσιάστηκαν οι δοκιμές που πραγματοποιήθηκαν με την ολοκλήρωση του συστήματος.





## Κεφάλαιο 5

### Συμπεράσματα

Το σύστημα που δημιουργήθηκε στην παρούσα διπλωματική εργασία αποτελείται από 3 σημαντικούς πυλώνες. Την ιστοσελίδα, την εφαρμογή σε Android και τη βάση δεδομένων που αποθηκεύει όλα τα στοιχεία και επικοινωνεί με τη σελίδα και την εφαρμογή ταυτόχρονα. Η βασική λειτουργία του συστήματος είναι η καταγραφή μετρήσεων κοινής ωφελείας μέσω συσκευής κινητού τηλεφώνου και η διαχείριση όλων των μετρήσεων αυτών από την εφαρμογή αλλά και την ιστοσελίδα, κατηγοριοποιημένων σε αντίστοιχα επίπεδα χρηστών. Πραγματοποιήθηκαν ξεχωριστά προφίλ, για να εξυπηρετήσουν το κάθε επίπεδο χρήστη ανάλογα με τις απαιτήσεις και τις λειτουργίες που υπήρχαν, όπως αυτές αναλύθηκαν σε προηγούμενο κεφάλαιο. Ο

διαχειριστής του συστήματος δημιουργεί χρήστες (τεχνικούς και απλούς χρήστες) και παρέχει άδεια για τη χρήση της εφαρμογής όπου αυτό θεωρείται επιτρεπτό. Έχει στη διάθεση του και επεξεργάζεται όλα τα στοιχεία του συστήματος, με δυνατότητες μετατροπών και διαγραφών, χρηστών και μετρήσεων. Επιπλέον μπορεί να προάγει ή υποβαθμίσει οποιονδήποτε χρήστη στο αντίστοιχο επίπεδο. Ακόμα μπορεί να ομαδοποιήσει και να αναθέσει απλούς χρήστες σε τεχνικούς με αντίστοιχα κριτήρια.

Οι τεχνικοί του συστήματος είναι υπεύθυνοι για την πραγματοποίηση καταγραφών μετρήσεων και την ενημέρωση του συστήματος με αυτές. Διαθέτουν δικό τους ξεχωριστό γραφικό περιβάλλον στο οποίο μπορούν να ενημερωθούν για όλες τις μετρήσεις που έχουν πραγματοποιήσει οι ίδιοι και να ελέγξουν την αντιστοιχία τους με αυτές των απλών χρηστών που τους έχουν ανατεθεί. Ταυτόχρονα μπορούν να επιβλέπουν και τις μετρήσεις των χρηστών

που έχουν υπό την εποπτεία τους. Επίσης, έχουν και αυτοί τις δυνατότητες επεξεργασίας και διαγραφής μετρήσεων, μόνο όμως στα πεδία που αναφέραμε.

Οι εγγεγραμμένοι χρήστες (απλοί χρήστες) του συστήματος, μπορούν να δημιουργήσουν λογαριασμό, και μετά από σχετική άδεια μπορούν να χρησιμοποιήσουν την εφαρμογή για να πραγματοποιούν τις προσωπικές τους καταγραφές. Στη συνέχεια μπορούν να επιβλέπουν τις μετρήσεις αυτές μέσω της εφαρμογής, ή να επισκεφθούν το ειδικά διαμορφωμένο γραφικό περιβάλλον που σχεδιάστηκε στην ιστοσελίδα.

Η εφαρμογή δοκιμάστηκε σε φυσιολογικές για τα ζητούμενα συνθήκες και τα αποτελέσματα ήταν αρκετά ικανοποιητικά.

Η γενικότερη ιδέα υλοποίησης αυτού του συστήματος πραγματοποιήθηκε σε ένα ικανοποιητικό επίπεδο, κάνοντας χρήση τεχνολογιών που είναι πολύ προσιτές σε ευρύ κοινό, σε συνδυασμό με καινοτόμες ενέργειες με γνώμονα την εύκολη εξυπηρέτηση και απλότητα χρήσης τους από τη μεριά των χρηστών κάθε επιπέδου στα οποία απευθύνεται.

Βέβαια όπως οι περισσότερες καινοτόμες ιδέες, έτσι και η παρούσα υλοποίηση συμπεριλαμβάνει πλεονεκτήματα και μειονεκτήματα, δυνατά σημεία και αδυναμίες.

Στα πλεονεκτήματα του συστήματος που υλοποιήθηκε, συμπεριλαμβάνεται η αμεσότητα με την οποία μπορούν να αποθηκευτούν σημαντικά δεδομένα στη βάση δεδομένων. Αυτό επιτυγχάνεται φυσικά με την οπτική αναγνώριση μέσω της εφαρμογής Android. Οποιοσδήποτε χρήστης μπορεί μέσα σε λίγα δευτερόλεπτα να πραγματοποιήσει μια καταγραφή, να την κατηγοριοποιήσει και να την αποθηκεύσει στο λογαριασμό του. Στη συνέχεια μπορεί άμεσα να ελέγξει την καταχώρηση του μέσω της εφαρμογής. Ακόμα, τα δεδομένα του είναι άμεσα διαθέσιμα οποιαδήποτε στιγμή στον προσωπικού του λογαριασμό στην ιστοσελίδα.

Στα μειονεκτήματα του συστήματος, διακρίνουμε τις περιπτώσεις όπου τα αποτελέσματα της αναγνώρισης μπορεί να μην είναι 100% ακριβή για διάφορους λόγους (π.χ. κακώς συντηρημένος μετρητής, αρκετά χαμηλής ευκρίνειας κάμερα συσκευής). Για το λόγο αυτό, υπάρχει η δυνατότητα

πρόχειρης επεξεργασίας αν η αναγνώριση δεν ήταν ακριβής, αλλά αυτό μας φέρνει και πάλι στα μειονεκτήματα καθώς η διαδικασία γίνεται περισσότερο χρονοβόρα. Ακόμα, όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, επιλέχθηκε πολύ υψηλή ανάλυση εικόνας στον αλγόριθμο αναγνώρισης, πράγμα που μπορεί να βοηθήσει στην καλύτερη αναγνώριση, όμως την ίδια στιγμή, αυξάνει κατά πολύ το μέγεθος του αρχείου που αποθηκεύεται στη συσκευή. Για το λόγο αυτό επιλέχθηκε μόνο η αποθήκευση της τελευταίας κατά σειρά, μέτρησης.

Στα δυνατά σημεία του συστήματος, συναντήσαμε την περίπτωση της βελτιστοποίησης της οπτικής αναγνώρισης και την ευελιξία που προσφέρει. Με την προσθήκη της βιβλιοθήκης tess-two, ανοίξαμε το δρόμο σε μία τεράστια λίστα επιλογών σχετικά με το κείμενο που θα αναγνωρίζει η εφαρμογή. Αυτό επιτυγχάνεται με την ύπαρξη χιλιάδων δηλωμένων αρχείων γλωσσών και αλφαριθμητικών στοιχείων σε καταλόγους της Google, πράγμα που σημαίνει ότι μπορεί να πραγματοποιηθεί εν μέρει αναβάθμιση στην αναγνώριση, με την απλούστατη διαδικασία του «drag and drop» του κατάλληλου αρχείου στον κατάλογο της εφαρμογής. Επιπλέον, υπάρχει δυνατότητα «εκπαίδευσης» του αλγορίθμου που ήδη χρησιμοποιείται, μέσω της εξάσκησης της εφαρμογής με λήψη εικόνων και ορίζοντας κάθε φορά τις γωνίες και τις καμπύλες των στοιχείων της εικόνας για καλύτερα αποτελέσματα.

Βέβαια η παραπάνω διαδικασία μπορεί να αποτελέσει και ένα αδύναμο σημείο του συστήματος, καθώς η τελειοποίηση του αλγορίθμου είναι μία πάρα πολύ δύσκολη, αν όχι αδύνατη, διαδικασία που απαιτεί σημαντική επένδυση χρόνου. Ένας ακόμα παράγοντας που αποτελεί αδυναμία, είναι το γεγονός ύπαρξης διαφορετικών τύπων μετρητών, με την έννοια των παλαιών και των καινούριων κατασκευών που υπάρχουν. Με αυτό το δεδομένο, ο αλγόριθμος αναγνώρισης, μπορεί να δυσκολευτεί πάρα πολύ να εξάγει σωστά αποτελέσματα καθώς παρατηρούνται σημαντικές διαφορές στις είδος της γραμματοσειράς του κάθε μετρητή. Ακόμα ένα αδύναμο σημείο του συστήματος είναι το γεγονός ότι η εφαρμογή έχει ως απαραίτητη προϋπόθεση την σύνδεση στο internet για την είσοδο του χρήστη και την αποστολή και ανάγνωση δεδομένων στη βάση. Αυτό σημαίνει ότι σε περίπτωση που ένας μετρητής βρίσκεται για παράδειγμα σε

κάποιο υπόγειο, ή γενικότερα σε σημείο που δεν υπάρχει δυνατότητα σύνδεσης της συσκευής, π.χ. λόγω εμβέλειας, είναι αδύνατο να πραγματοποιηθεί είσοδος του χρήστη και κατ' επέκταση η πραγματοποίηση καταγραφής μέτρησης.

## Μελλοντικές επεκτάσεις

Στο σύστημα που υλοποιήθηκε στην παρούσα διπλωματική εργασία, είναι βέβαιο ότι μπορούν να προστεθούν πάρα πολλές προδιαγραφές και μελλοντικές επεκτάσεις, καθώς αποτελεί λογισμικό με πάρα πολλές δυνατότητες και τεράστια ευελιξία.

Από τις κυριότερες προσθήκες που μπορούν να πραγματοποιηθούν είναι η δυνατότητα “offline” σύνδεσης στην εφαρμογή και η αποθήκευση δεδομένων καταγραφής μετρήσεων σε ένα προσωρινό κατάλογο ή βάση, σε περίπτωση που δεν είναι δυνατή η σύνδεση της συσκευής που χρησιμοποιείται, στο internet. Με την προσωρινή αυτή αποθήκευση η εφαρμογή θα αποστέλλει τα δεδομένα που καταγράφηκαν όσο βρισκόταν εκτός σύνδεσης, αυτόματα μόλις συνδεθεί και θα ενημερώνει σχετικά το χρήστη της συσκευής.

Ένα άλλο χαρακτηριστικό παράδειγμα μελλοντικής επέκτασης αποτελεί η προσθήκη της δυνατότητας στο χρήστη της εφαρμογής Android να επιλέγει δυναμικά μέσω μίας λίστας, την κατάλληλη γραμματοσειρά αλφαριθμητικών δεδομένων, για κάθε περίπτωση μετρητή. Με αυτόν τον τρόπο θα επιταχύνεται η καταγραφή μετρήσεων και η αποφυγή λαθών σε πολύ μεγάλο ποσοστό.

Επιπλέον, στον τομέα της οπτικής αναγνώρισης, μπορεί να προστεθεί η δυνατότητα δημιουργία ετικετών QR code για κάθε μετρητή μετά την πρώτη καταγραφή, ώστε να ελαχιστοποιηθεί κατά πολύ ο χρόνος και η διαδικασία καταχώρησης μετρήσεων ανά ιδιοκτήτη μετρητή.

Ακόμα μπορεί να πραγματοποιηθεί μία λειτουργία η οποία θα αποστέλλει στο διακομιστή την ακριβή εικόνα από την οποία εξήχθη μία μέτρηση, γλιτώνοντας έτσι σημαντικό αποθηκευτικό χώρο στη συσκευή. Με την ίδια λογική η εικόνα αυτή θα καταχωρείται ξεχωριστά σε κάθε μέτρηση που πραγματοποιήθηκε και θα δημιουργείται ένας σύνδεσμος επάνω στην κάθε

μέτρηση ενός πίνακα, ο οποίος θα ανακατευθύνει και θα προβάλλει στο χρήστη την εικόνα, ώστε να επαληθευτεί η καταχωρημένη μέτρηση.

Στον τομέα της ιστοσελίδας, θα μπορεί να αναπτυχθεί ένας αλγόριθμος ο οποίος θα ενημερώνει τους χρήστες κάθε επιπέδου με αντίστοιχα μηνύματα για την συνολική εικόνα των μετρήσεων τους. Ταυτόχρονα θα επισημαίνει, κυρίως στους τεχνικούς, που είναι απαραίτητο να πραγματοποιηθούν μετρήσεις σε συγκεκριμένο χρονικό διάστημα.

Στο γενικότερο πλάνο του συστήματος, θα μπορούσε να αποτελέσει επέκταση, η εισαγωγή περισσότερων επιπέδων χρηστών. Με την ύπαρξη των 3 βασικών επιπέδων τη δεδομένη χρονική στιγμή, η προσθήκη και άλλων επιπέδων θα μπορούσε να διευκολύνει κατά πολύ τις λειτουργίες του συστήματος, και να ανοίξει δρόμους για επεκτάσεις όπως αυτές που αναφέρθηκαν.

## Οδηγίες εγκατάστασης

Στο παράρτημα αυτό θα περιγραφεί η διαδικασία εγκατάστασης της εφαρμογής Android αλλά και της ιστοσελίδας που δημιουργήθηκαν στην παρούσα διπλωματική εργασία, σε διαφορετικό σύστημα διατηρώντας τις δυνατότητες και τις λειτουργίες που προσφέρει η υλοποίησή του.

### Εγκατάσταση αρχείων σε διακομιστή

Σε πρώτο στάδιο θα δοθούν οδηγίες εγκατάστασης των αρχείων σε διαφορετικό διακομιστή. Η εγκατάσταση αυτή έχει πολύ μεγαλύτερη σημασία διότι συμπεριλαμβάνει και τα αρχεία με τα οποία επικοινωνεί η εφαρμογή Android με τη βάση δεδομένων.

Αρχικά βασικές προϋποθέσεις της μεταφοράς, αποτελούν η υποστήριξη ύπαρξης βάσεων δεδομένων MySQL και η υποστήριξη έκδοσης PHP 5.1 και άνω, για τη σωστή λειτουργία όλων των ερωτημάτων μορφής PDO που υπάρχουν στα αντίστοιχα αρχεία.

Με δεδομένες τις προϋποθέσεις αυτές, για την μεταφορά των αρχείων πραγματοποιείται αντιγραφή όλων των αρχείων του καταλόγου «diplwmatikh». Στο διακομιστή αυτόν τα δικαιώματα όλων των αρχείων είναι 755. Οπότε σε περίπτωση μεταφοράς σε αντίστοιχο διακομιστή θα πρέπει να γίνουν οι απαραίτητες ενέργειες αλλαγής δικαιωμάτων.

Για τη δημιουργία της βάσης δεδομένων ανοίγουμε την κονσόλα του διακομιστή τύπου command prompt, και πραγματοποιούμε σύνδεση στη MySQL με δικαιώματα διαχειριστή. Πραγματοποιούμε εγκατάσταση της βάσης με την εντολή `mysql -u username -D database_name -p < APPDB.sql`. Με την επιτυχή εκτέλεσή του θα δημιουργηθούν όλοι οι πίνακες και οι συσχετίσεις τους για τη σωστή λειτουργία του συστήματος. Η εγκατάσταση αυτή παρέχει ως μοναδική εγγραφή τον admin στον πίνακα users κωδικό `adminkwdikos#`.

Εάν όλα τα παραπάνω βήματα εκτελέστηκαν επιτυχώς τότε η ιστοσελίδα και η βάση δεδομένων είναι πλήρως λειτουργική και μένει να πραγματοποιηθεί

εγκατάσταση της εφαρμογής Android, για την αξιοποίηση όλων των δυνατοτήτων του συστήματος

## Εγκατάσταση εφαρμογής Android

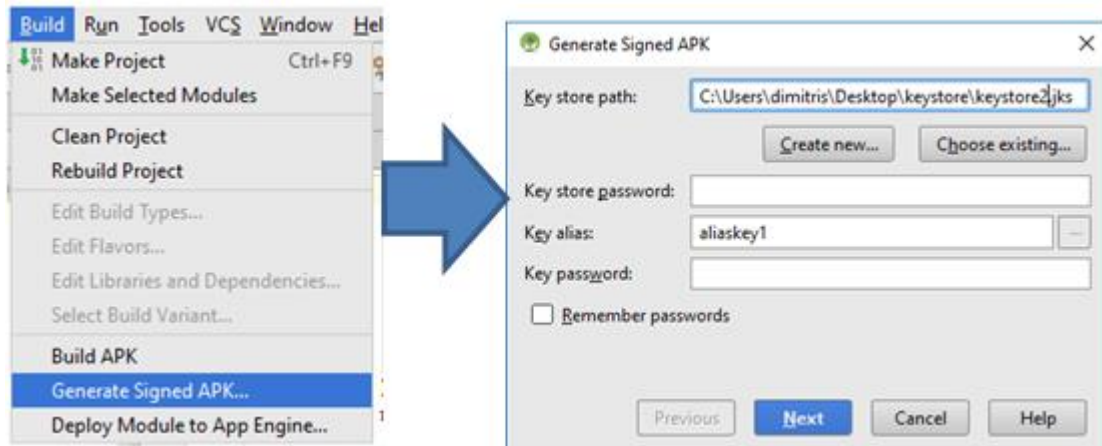
Για την εγκατάσταση της εφαρμογής Android σε κάποιο άλλο σύστημα θα πρέπει να πραγματοποιηθεί αρχικά εισαγωγή όλου του project σε ένα καινούριο κατάλογο του Android Studio. Στη συνέχεια θα πρέπει να πραγματοποιηθεί η διαδικασία εγκατάστασης της εξωτερικής βιβλιοθήκης tess-two, ακριβώς όπως παρουσιάστηκε στο κεφάλαιο 2.

Με δεδομένο την αλλαγή του διακομιστή στον οποίον βρίσκονται τα rhp αρχεία με τα οποία επικοινωνεί η εφαρμογή, θα πρέπει να ανατρέξουμε την εφαρμογή και να πραγματοποιήσουμε τροποποιήσεις στους συνδέσμους. Συγκεκριμένα θα πρέπει να πραγματοποιηθούν αλλαγές στα παρακάτω σημεία:

- Αλλαγή του path URL\_REGISTER μέσα στη διεργασία Appconfig της εφαρμογής με το αντίστοιχο του καινούριου διακομιστή
- Αλλαγή του path URL\_LOGIN μέσα στη διεργασία Appconfig της εφαρμογής με το αντίστοιχο του καινούριου διακομιστή
- Αλλαγή του path insertUrl μέσα στη διεργασία AndroidOCRActivity της εφαρμογής με το αντίστοιχο του καινούριου διακομιστή
- Αλλαγή του path url μέσα στη διεργασία MeasurementsActivity της εφαρμογής με το αντίστοιχο του καινούριου διακομιστή.

Επίσης θα πρέπει να πραγματοποιηθεί αλλαγή του συνδέσμου <http://zafora.ict.e.uowm.gr/~ictest00182/mainsite/android/app-release.apk> με τον αντίστοιχο του καινούριου διακομιστή.

Μόλις τα βήματα αυτά πραγματοποιηθούν επιτυχώς, θα πρέπει να εξαχθεί από το Android Studio η εκτελέσιμη μορφή της εφαρμογής που σχεδιάστηκε. Για να γίνει αυτό, πρέπει να επιλεγεί η επιλογή “Generate Singed APK”, και στη συνέχεια να εισαχθούν στοιχεία και κωδικοί για τη δημιουργία ασφαλούς keystore για την εφαρμογή. Η διαδικασία αυτή εμφανίζεται στην επόμενη εικόνα (Εικόνα 55).



**Εικόνα 55 :** Δημιουργία εκτελέσιμου αρχείου εφαρμογής

Αφού ο επιλεγθεί ο κατάλογος αποθήκευσης του keystore και δημιουργηθεί το εκτελέσιμο αρχείο, μπορεί πλέον να μεταφερθεί στη συσκευή προς εγκατάσταση.

Να σημειωθεί εδώ, πως είναι επιθυμητό οι κωδικοί που θα εισαχθούν να είναι ισχυροί. Επιπλέον και αυτοί και τα στοιχεία που θα τοποθετηθούν θα πρέπει να απομνημονευτούν με κάποιο τρόπο για μελλοντική χρήση.

Έχοντας το αρχείο λοιπόν μέσα στη συσκευή, μπορούμε να ξεκινήσουμε την εγκατάσταση του. Σε περίπτωση που η επιλογή αποφυγής εγκατάστασης εφαρμογών από εξωτερικές πηγές είναι ενεργοποιημένη, θα πρέπει να απενεργοποιηθεί. Μόλις η εγκατάσταση ολοκληρωθεί, η εφαρμογή είναι πλήρως λειτουργική και έτοιμη για να χρησιμοποιηθεί.





## Βιβλιογραφία

- [1] Nuance. (2015). OmniPage Standard 18 . : <http://ocr-software-review.toptenreviews.com/omnipage-standard-review.html>. Μάιος 2016.
- [2] Adobe Acrobat . (-). Adobe Pro DC . : <http://ocr-software-review.toptenreviews.com/acrobat-review.html>. Μάιος 2016.
- [3] ABBYY. (). ABBYY FineReader Professional 12. : <http://ocr-software-review.toptenreviews.com/acrobat-review.html>. Μάιος 2016.
- [4] Wikipedia. (). HTML . : <https://en.wikipedia.org/wiki/HTML>. Ιούνιος 2016.
- [5] Wikipedia. (). HTML . : [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Ιούνιος 2016.
- [6] Wikipedia. (). PHP . : <https://en.wikipedia.org/wiki/PHP>. Ιούνιος 2016.
- [7] Wikipedia. (). SQL . : <https://en.wikipedia.org/wiki/SQL>. Ιούνιος 2016.
- [8] Wikipedia. (). JavaScript . : <https://en.wikipedia.org/wiki/JavaScript>. Ιούνιος 2016.
- [9] Wikipedia. (). AJAX . : <https://en.wikipedia.org/wiki/Ajax>. Ιούνιος 2016.
- [10] Android Studio . (). Android Studio . : <https://developer.android.com/studio/index.html>. Ιούνιος 2016
- [11] Rmtheis. (). Tesseract . d: <https://github.com/tesseract-ocr>. d d Ιούνιος 2016
- [12] Leptonica . (). Leptonica . : <http://tpgit.github.io/UnOfficialLeptDocs/leptonica/>. Ιούνιος 2016

- [13] NDK . (). NDK . : <https://developer.android.com/ndk/index.html>. Ιούνιος 2016
- [14] rmtheis. (). tess-two . : <https://github.com/rmtheis/tess-two>. Ιούνιος 2016
- [15] Notepad++ . (). : <https://notepad-plus-plus.org/>. Ιούνιος 2016
- [16] Wikipedia . (). ER Model –Μοντέλο Οντοτήτων Συσχετίσεων . : [https://en.wikipedia.org/wiki/Entity%E2%80%93relationship\\_model](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model). Ιούνιος 2016.
- [17] Base64. (). Base64\_encode. : <https://www.base64decode.org/>. Ιούνιος 2016.
- [18] MySQL. (). ENUM . d: <http://dev.mysql.com/doc/refman/5.7/en/enum.html>. Ιούνιος 2016.
- [19] php. (). Filter\_var . : <http://php.net/manual/en/filter.filters.php> . Ιούνιος 2016.
- [20] Wikipedia. (). MySQLi . : <https://en.wikipedia.org/wiki/MySQLi>. Ιούνιος 2016.
- [21] Android SSL . (). Android SSL . : <https://developer.android.com/training/articles/security-ssl.html>. Ιούνιος 2016.
- [22] Android Studio . (). AVD . : <https://developer.android.com/studio/run/managing-avds.html>. Ιούνιος 2016.
- [23] Android Studio . (). Introduction. : <https://www.raywenderlich.com/120508/beginning-android-development-tutorial-android-studio>. Ιούνιος 2016
- [24] Stackoverflow. (). Stackoverflow. : <http://stackoverflow.com/>. Ιούνιος 2016.

- [25] Androidhive. (). Androidhive. : <http://www.androidhive.info/>. Ιούνιος 2016.
- [26] Vogella. (). Android Studio Tutorials . : <http://www.vogella.com/tutorials/Android/article.html>. Ιούνιος 2016.
- [27] Vogella. (). Camera API . : <http://www.vogella.com/tutorials/AndroidCamera/article.html>. Ιούνιος 2016.
- [28] w3schools. (). w3s. : <http://www.w3schools.com/>. Ιούνιος 2016.
- [29] Donn Felker (December 2010). Android Application Development For Dummies. US: ISBN: 978-0-470-77018-4. 384 pages.
- [30] Sams Teach Yourself (2014). HTML, CSS and JavaScript All in One. USA: Pearson Education. 89.
- [31] Android Magazine. Issue 21 Magazine Digital Edition. Ιούνιος 2016.
- [32] Android Magazine. Issue 32 Magazine Digital Edition. Ιούνιος 2016.
- [33] Android Magazine. Issue 44 Magazine Digital Edition. Ιούνιος 2016.
- [34] Android Magazine. Issue 45 Magazine Digital Edition. Ιούνιος 2016.