



ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Διπλωματική Εργασία

**Ανάπτυξη εφαρμογής σε Android για την
εύρεση φαρμακείων και νοσοκομείων.**

Χατζηαθανασίου Κωνσταντίνος

Επιβλέπων καθηγητής: Παντελής Αγγελίδης

Κοζάνη, 2020



ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Διπλωματική Εργασία

**Ανάπτυξη εφαρμογής σε Android για την
εύρεση φαρμακείων και νοσοκομείων.**

Χατζηαθανασίου Κωνσταντίνος

Επιβλέπων καθηγητής: Παντελής Αγγελίδης

Κοζάνη, 2020

Περίληψη

Η παρούσα διπλωματική εργασία έχει ως βασικό στόχο την ανάπτυξη μιας εφαρμογής, η οποία συμβάλλει στην καθοδήγηση κατά την διάρκεια της εύρεσης νοσοκομείου ή φαρμακείου. Η εφαρμογή, σχεδιάστηκε αξιοποιώντας τεχνολογίες αιχμής, δημιουργώντας ένα εξατομικευμένο και φιλικό για κάθε χρήστη περιβάλλον, εξυπηρετώντας τις ανάγκες του για καθοδήγηση σε διάφορα μέρη, εκμεταλλευόμενος τα προηγμένα χαρακτηριστικά και τις δυνατότητες που του παρέχει η εφαρμογή μας.

Συγκεκριμένα, η εν λόγω εφαρμογή, επιτρέπει στον χρήστη να βρει τα διάφορα φαρμακεία και νοσοκομεία της πόλης, να εντοπίσει κοντινά σημεία στη θέση που βρίσκεται προκειμένου να εξυπηρετήσει τις ανάγκες του και να δει τις απαραίτητες πληροφορίες που πρέπει να γνωρίζει και τέλος, του επιτρέπει να γνωρίζει την απόσταση που απέχει από τον προορισμό για την εύκολη επιλογή του πλησιέστερο φαρμακείου ή νοσοκομείου στο σημείο που βρίσκεται, έχοντας έτσι μια ψηφιακή εικόνα για το που βρίσκεται και για το που θα ήθελε να κατευθυνθεί.

Τέλος τα βασικά στοιχεία της συγκεκριμένης διπλωματικής εργασίας επικεντρώνονται σε τρεις βασικούς άξονες. Συγκεκριμένα, στον πρώτο άξονα, γίνεται η ανάλυση του λογισμικού που χρησιμοποιήθηκε, στον δεύτερο άξονα γίνεται η ανάλυση των συγκεκριμένων τεχνολογιών που εφαρμόστηκαν στην υλοποίηση της εφαρμογής. Και στον τελευταίο η παρουσίαση του κώδικα και της εφαρμογής.

Λέξεις Κλειδιά: Android εφαρμογή, Kotlin, Νοσοκομεία, Εφημερεύοντα φαρμακεία

Abstract

The main objective of this thesis is to develop an application that contributes to guidance during hospital or pharmacy finding. The application was designed using state-of-the-art technologies, creating a personalized and user-friendly environment, serving its needs for guidance in various places, taking advantage of the advanced features and capabilities of our application.

Specifically, this application allows the user to find the various pharmacies and hospitals in the city, locate nearby locations in order to serve their needs and view the necessary information they need to know and finally, allows them to know its distance from the destination for easy selection of the nearest pharmacy or hospital where it is located, thus having a digital picture of where it is and where it would like to be directed.

Finally, the key elements of this thesis focus on three main pillars. Comparatively, in the first axis, the software used is analyzed, in the second axis the specific technologies applied in the implementation of the application are analyzed. And lastly, the presentation of the code and the application

Keywords: Android application, Kotlin, Hospitals, On duty pharmacy

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή του τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, του Πανεπιστημίου Δυτικής Μακεδονίας, κύριο Παντελή Αγγελίδη για την εξαιρετική καθοδήγηση του, για την υπομονή του, για τη συνεχή επίβλεψη και συνεργασία που είχαμε καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής εργασίας μου.

Επίσης θα ήθελα να ευχαριστήσω τους φίλους που απέκτησα κατά τη διάρκεια των σπουδών μου στο Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών, για τις όμορφες στιγμές, για την στήριξη και την άψογη συνεργασία που είχαμε όλα αυτά τα χρόνια.

Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου για την πνευματική και υλική υποστήριξη που μου πρόσφεραν όλα αυτά τα χρόνια.

Κωνσταντίνος Χατζηαθανασίου, Κοζάνη 2020

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 Ανάπτυξη εφαρμογής σε Android για την εύρεση φαρμακείων και νοσοκομείων 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. Παντελής Αγγελίδης 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) Κωνσταντίνος Χατζηαθανασίου , Παντελής Αγγελίδης, 2020 , Κοζάνη

Πίνακας Περιεχομένων

1. Εισαγωγή	11
1.1 Αντικείμενο.....	12
1.2 Οργάνωση Κεφαλαίων	12
2. Θεωρητικό Υπόβαθρο	14
2.1 Τι είναι το Android	14
2.2 Ιστορική εξέλιξη	15
2.3 Λογότυπο	16
2.4 Χαρακτηριστικά και λειτουργίες.....	17
2.5 Εκδόσεις Android	19
2.5.1 Android 1.0 και Android 1.1	19
2.5.2 Android 1.5 Cupcake.....	20
2.5.3 Android 1.6 Donut.....	21
2.5.4 Android 2.0 Eclair	21
2.5.5 Android 2.2-2.2.3 Froyo.....	22
2.5.6 Android 2.3-2.3.7 Gingerbread	23
2.5.7 Android 3.0-3.2 Honeycomb.....	24
2.5.8 Android 4.0-4.0.2 Ice Cream Sandwich.....	24
2.5.9 Android 4.1-4.3.1 Jelly Bean.....	25
2.5.10 Android 4.4-4.4.4 Kit Kat	25
2.5.11 Android 5.0 Lollipop.....	26
2.5.12 Android 6.0-6.0.1 Marshmallow	27
2.5.13 Android 7.0-7.1.1 Nougat.....	28
2.5.14 Android Oreo.....	29
2.5.15 Android Pie.....	29
2.6 Έκδοση της εφαρμογής.	30

3. Τεχνολογίες που χρησιμοποιήθηκαν	31
3.1 Kotlin	31
3.2 PHP	32
3.3 MySQL	33
3.3 RSS	34
4. Εργαλεία που χρησιμοποιήθηκαν	35
4.1 Android Studio.....	35
4.2 Xampp.....	36
4.3 Firebase.....	37
4.4 Adobe Photoshop.....	38
4.5 Zapier	38
5 Παρουσίαση της εφαρμογής Pharmacy.....	39
5.1 Δημιουργία της εφαρμογής	39
5.2 Βιβλιοθήκες	40
5.3 Κλάσεις.....	41
5.3.1 Κλάση MainActivity	41
5.3.2 Κλάση ActivityHospitals.....	44
5.3.3 Κλάση Hospital	50
5.3.4 Κλάση MyListHospital.....	50
5.3.5 Κλάση ViewPharmacyActivity	51
5.3.6 Κλάση EndPoints.....	60
5.3.7 Κλάση Pharmacy	60
5.3.8 Κλάση PharmacyList.....	60
5.3.9 Κλάση SplashScreen	64
5.3.10 Κλάση ActivityArticles	65
5.3.11 Κλάση News.....	69
5.3.12 Κλάση MyListAdapter	69

5.3.12 Κλάση NewsPost	70
6. Επίλογος.....	71
6.1 Συμπεράσματα.....	71
6.2 Μελλοντικές επεκτάσεις.....	71
7. Βιβλιογραφικές αναφορές.....	73

Πίνακας Εικονών

Εικόνα 1 - Ποσοστά χρήσης λογισμικών σε Smartphones	14
Εικόνα 2- Το λογότυπο του Android από το 2007 έως το 2014	17
Εικόνα 3- Το λογότυπο του Android από το 2014 έως το 2017	17
Εικόνα 4- Το λογότυπο του Android από το 2017	17
Εικόνα 5- Εκδόσεις του Android.....	19
Εικόνα 6- Λογότυπο της έκδοσης Android 1.0	19
Εικόνα 7- Λογότυπο της έκδοσης Cupcake 1.5	20
Εικόνα 8- Λογότυπο της έκδοσης Donut 1.6.....	21
Εικόνα 9- Λογότυπο της έκδοσης Eclair 2.0	21
Εικόνα 10- Λογότυπο της έκδοσης Froyo 2.2-2.2.3.....	22
Εικόνα 11- Λογότυπο της έκδοσης Gingerbread 2.3-2.3.7	23
Εικόνα 12- Λογότυπο της έκδοσης Honeycomb 3.0-3.2.....	24
Εικόνα 13- Λογότυπο της έκδοσης Ice Cream Sandwich 4.0-4.0.2.....	24
Εικόνα 14- Λογότυπο της έκδοσης Jelly Bean 4.1-4.3.1	25
Εικόνα 16- Λογότυπο της έκδοσης Lollipop 5.0.....	26

Εικόνα 17- Λογότυπο της έκδοσης Marshmallow 6.0-6.0.1	27
Εικόνα 18- Λογότυπο της έκδοσης Nougat 7.0-7.1.1	28
Εικόνα 19- Λογότυπο της έκδοσης Android Oreo	29
Εικόνα 20- Λογότυπο της έκδοσης Android Pie	29
Εικόνα 21- Ποσοστά χρήσης εκδόσεων σε android κινητά.	30
Εικόνα 22- Λογότυπο της γλώσσας Kotlin.	31
Εικόνα 22- Το λογότυπο της PHP	32
Εικόνα 23- Το λογότυπο της MySQL	33
Εικόνα 24- Το λογότυπο του RSS	34
Εικόνα 25- Περιβάλλον χρήσης του Android Studio	35
Εικόνα 26- Το λογότυπο του Xampp	36
Εικόνα 27- Το λογότυπο του Firebase.....	37
Εικόνα 28- Το λογότυπο του Adobe Photoshop.....	38
Εικόνα 29- Το λογότυπο του Zapier.....	38
Εικόνα 30- Επιλογές συστήματος.....	39
Εικόνα 31- Αιτήσεις πρόσβασης λειτουργιών.....	40
Εικόνα 32- Εγκατάσταση βιβλιοθηκών στο build.gradle(Module.app).....	41
Εικόνα 33- Εικόνα από την κλάση MainActivity.....	42
Εικόνα 34- Εικόνα από την κλάση MainActivity.....	42
Εικόνα 35- Εικόνα από την κλάση MainActivity.....	43
Εικόνα 36- Εικόνα από την κλάση MainActivity.....	43
Εικόνα 37- Εικόνα από την MainActivity	44
Εικόνα 38- Εικόνα από την βάση δεδομένων.	45
Εικόνα 39- Εικόνα από την ActivityHospitals	45
Εικόνα 40- Εικόνα από την ActivityHospitals	46
Εικόνα 41- Εικόνα από την ActivityHospitals	46
Εικόνα 42- Εικόνα από την ActivityHospitals	47

Εικόνα 43- Εικόνα από τα νοσοκομεία	48
Εικόνα 44- Εικόνα από την χρησιμοποίηση του τηλεφώνου από το νοσοκομείο.....	48
Εικόνα 45- Εικόνα από την χρησιμοποίηση των οδηγιών	49
Εικόνα 46- Εικόνα από την χρησιμοποίηση της ιστοσελίδας.....	49
Εικόνα 47- Εικόνα από την κλάση Hospital.....	50
Εικόνα 48- Εικόνα από την κλάση MyListHospital.....	51
Εικόνα 49- Εικόνα από την βάση δεδομένων.	52
Εικόνα 50- Στιγμιότυπο από το αρχείο Connect	53
Εικόνα 51- Στιγμιότυπο από το αρχείο DbOperation.....	53
Εικόνα 52- Στιγμιότυπο από το αρχείο DbOperation.....	54
Εικόνα 53- Στιγμιότυπο από το αρχείο index.....	54
Εικόνα 54- Εικόνα από την κλάση ViewPharmacyActivity	55
Εικόνα 55- Εικόνα από την κλάση ViewPharmacyActivity	55
Εικόνα 56- Εικόνα από την κλάση ViewPharmacyActivity	56
Εικόνα 57- Εικόνα από το UI των φαρμακείων	56
Εικόνα 58- Εικόνα από το UI των φαρμακείων.	57
Εικόνα 59- Εικόνα από το UI των φαρμακείων.	58
Εικόνα 60- Εικόνα από το UI των φαρμακείων.	59
Εικόνα 61- Εικόνα από το UI των φαρμακείων.	59
Εικόνα 62- Εικόνα από την κλάση EndPoints.....	60
Εικόνα 63- Εικόνα από την κλάση Pharmacy.	60
Εικόνα 64- Εικόνα από την κλάση PharmacyList.....	61
Εικόνα 65- Εικόνα από την κλάση PharmacyList.....	61
Εικόνα 66- Εικόνα από την κλάση PharmacyList.....	62
Εικόνα 67- Εικόνα από την κλάση PharmacyList.....	62
Εικόνα 68- Εικόνα από την κλάση PharmacyList.....	63
Εικόνα 69- Εικόνα από την κλάση PharmacyList.....	63

Εικόνα 70- Εικόνα από την κλάση SplashScreen.	64
Εικόνα71- Εικόνα με το λογότυπο της εφαρμογής.	65
Εικόνα 72- Εικόνα από της βάση δεδομένων.	66
Εικόνα 74- Εικόνα από το UI των άρθρων.....	67
Εικόνα 75- Εικόνα από την κλάση ActivityArticles	68
Εικόνα 76- Εικόνα από την κλάση ActivityArticles	69
Εικόνα 77- Εικόνα από την κλάση News.	69
Εικόνα 78- Εικόνα από την κλάση MyListAdapter.	70
Εικόνα 79- Εικόνα από την κλάση NewsPost.	70

1. Εισαγωγή

Η περιπέτεια της κινητής τηλεφωνίας ξεκίνησε αμέσως μετά τον Β' παγκόσμιο πόλεμο με τις πρώτες προσπάθειες Σουηδών, Φιλανδών και Αμερικανών να δημιουργήσουν μια συσκευή που δεν εξαρτάται από καλωδιακή σύνδεση με δίκτυο παροχής τηλεφωνίας, ούτε από κάποια τοπική ασύρματη συσκευή εκπομπής ραδιοφωνικού σήματος χαμηλής συχνότητας. Όμως ως ληξιαρχική πράξη γέννησής της θεωρείται η 3η Απριλίου 1973 με την κατασκευή και έναρξη της χρήσης του κινητού τηλεφώνου από το κοινό. Τα κινητά τηλέφωνα δεν είχαν πάντοτε οθόνη αφής, κάμερα πολλών Megapixels και τη δυνατότητα να χρησιμοποιηθούν ως ένας μικροσκοπικός Η/Υ, λύνοντας μας τα χέρια. Βέβαια η απογείωση των κινητών τηλεφώνων άρχισε την δεκαετία του 90' όπου με την ψηφιοποίηση δικτύων και συσκευών οι συσκευές έγιναν μικρότερες και ελαφρύτερες χωρώντας αντίθετα με παλαιότερα στην παλάμη ή στην τσέπη.

Η εξέλιξη αυτή οδήγησε ώστε η νοοτροπία τόσο των χρηστών αλλά και των εταιρειών ως προς τον τρόπο αντιμετώπισης τους απέναντι στα κινητά τηλέφωνα να αλλάξει και να αντιμετωπίζεται περισσότερο ως ένας φορητός υπολογιστής με την ιδιότητα και του τηλεφώνου αλλά σε μέγεθος κινητού τηλεφώνου. Για τη λειτουργία τους δημιουργήθηκαν πολλά λειτουργικά συστήματα από πολλές εταιρίες. Εκείνο όμως που κυριάρχησε είναι το λογισμικό Android. Το λογισμικό αυτό, αναπτύχθηκε και παρουσιάστηκε από την Google στα τέλη του 2007. Ο αριθμός των εφαρμογών που ο χρήστης έχει τη δυνατότητα να εγκαταστήσει στην συσκευή του, είναι κοντά στις 3.000.000 εφαρμογές. Οι εφαρμογές αναπτύσσονται σε διάφορες πλατφόρμες και ο καθένας μπορεί να αναπτύξει την δικιά του και να την προωθήσει μέσω του Play store ή κάποιου άλλου store.[1]

1.1 Αντικείμενο

Βασικός στόχος της παρούσας διπλωματικής εργασίας, αποτελεί η ανάπτυξη μιας εφαρμογής για κινητές συσκευές, η οποία θα εξυπηρετεί και θα καθοδηγεί τους χρήστες για την εύκολη αναζήτηση των φαρμακείων ή νοσοκομείων της πόλης. Για την κάλυψη μεγαλύτερου μέρους χρηστών η εφαρμογή θα αναπτυχθεί για λειτουργικό σύστημα Android. Και θα βοηθήσει στην μείωση του χρόνου αναζήτησης και ταλαιπωρίας που βιώνει για να βρει το προϊόν που θέλει. Η εφαρμογή δεν θα έχει μόνο αυτές τις λειτουργίες, ο χρήστης θα μπορεί να διαβάσει άρθρα που έχουν θέμα για την υγεία, άθληση και την διατροφή με σκοπό να μπορεί να ψυχαγωγηθεί και να ενημερωθεί. Άλλη μια λειτουργία της εφαρμογής θα είναι η σάρωση barcode και την αυτοματοποιημένη αναζήτηση του στο διαδίκτυο. Έτσι ώστε να μπορεί να βρει πληροφορίες για το φάρμακο που θέλει.

1.2 Οργάνωση Κεφαλαίων

Η παρούσα διπλωματική εργασία χωρίζεται συνολικά σε επτά κεφάλαια – θεματικές ενότητες.

Στο 1^ο κεφάλαιο γίνεται μια σύντομη ιστορική αναδρομή για το κινητό τηλέφωνο. Επιπλέον, γίνεται μια σύντομη παρουσίαση του θέματος της εργασίας και τους λόγους που έγινε η επιλογής της συγκεκριμένης εφαρμογής.

Στο 2^ο κεφάλαιο γίνεται μία εκτενέστερη ανάλυση του λειτουργικού συστήματος. Επίσης, αναλύονται οι εκδόσεις του λογισμικού και στον λόγο επιλογής της συγκεκριμένης έκδοσης.

Στο 3^ο κεφάλαιο γίνεται η ανάλυση των γλωσσών που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Και στην παρουσίαση βασικών στοιχείων που έχουν.

Στο 4^ο κεφάλαιο γίνεται μια σύντομη περιγραφή του Android Studio, Firebase, Adobe Photoshop και του Xampp.

Στο 5ο κεφάλαιο παρουσιάζεται αναλυτικά ο κώδικας και οι λειτουργίες της εφαρμογής Pharmacy που διαθέτει. Με την βοήθεια στιγμιότυπων οθόνης καταγράφεται η εμφάνιση της εφαρμογής, οι δυνατότητες που προσφέρει και παρέχεται εκτενής εξήγηση για τον τρόπο χρήσης της.

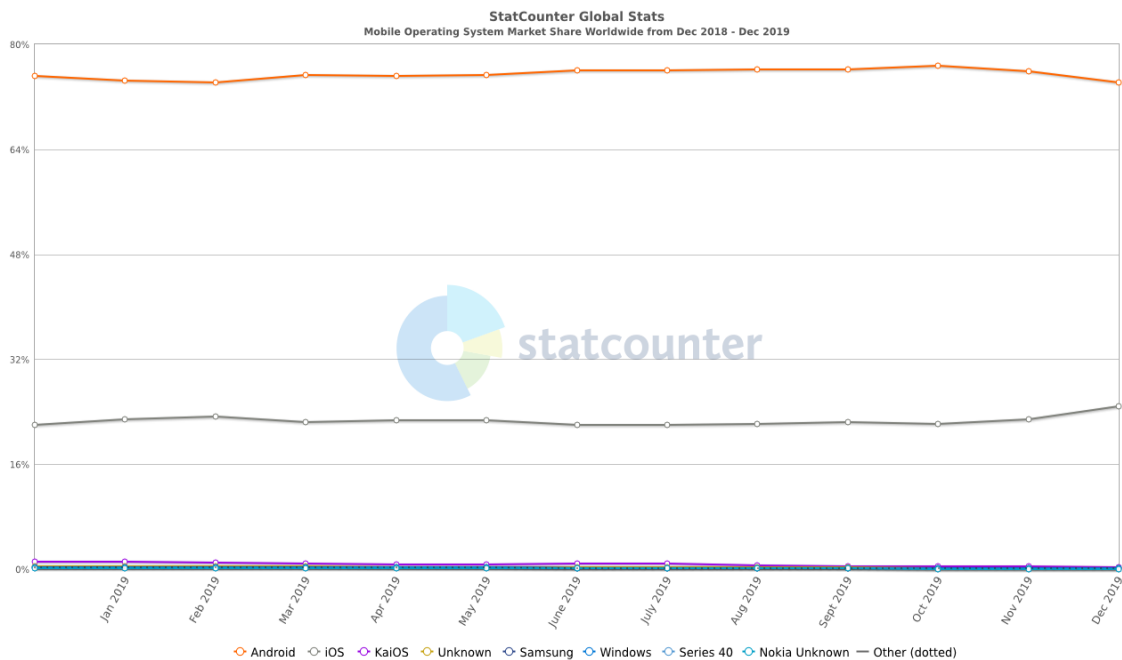
Στο 6ο κεφάλαιο γίνεται μια σύντομη επισκόπηση της διπλωματικής εργασίας και παρουσιάζονται διάφορα αποτελέσματα και συμπεράσματα που προέκυψαν. Επίσης, γίνεται αναφορά για πιθανές μελλοντικές επεκτάσεις.

Στο 7^ο κεφάλαιο παρουσιάζεται οι βιβλιογραφικές αναφορές

2. Θεωρητικό Υπόβαθρο

2.1 Τι είναι το Android

Το Android είναι ένα λειτουργικό σύστημα που ενσωματώνεται σε συσκευές κινητής τηλεφωνίας, τα οποία διαθέτουν οθόνη αφής, τρέχουν τον πυρήνα (kernel) του λειτουργικού Linux και ακόμη, επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού της Google. Το Android είναι ένα προϊόν ελεύθερου λογισμικού, για αυτό υπάρχουν πάρα πολλές συσκευές με Android, όπου βέβαια η κάθε μία έχει τα δικά της διαφορετικά χαρακτηριστικά και κατασκευάζονται από διαφορετικές εταιρίες όπου η κάθε μια δίνει διαφορετική βαρύτητα σε διαφορετικά χαρακτηριστικά[2].



Εικόνα 1 - Ποσοστά χρήσης λογισμικών σε Smartphones

Ως αναφορά το λογισμικό που χρησιμοποιούν τα έξυπνα τηλέφωνα, την πρώτη θέση παίρνουν αυτά που χρησιμοποιούν Android OS με 74.13% ενώ ακολουθούν τα iPhone χρησιμοποιούν iOS με 24.79% και ακολουθούν τα Windows OS με 0.1%. Αναλυτικότερα, σύμφωνα με έρευνα της Statcounter, που έγινε τον Δεκέμβριο του 2019 και αφορούσε την κατανομή των λογισμικών ανάμεσα στα smartphone της αγοράς, που φαίνεται στον παραπάνω πίνακα[3].

2.2 Ιστορική εξέλιξη

Το λειτουργικό σύστημα Android ιδρύθηκε στο Palo Alto της Καλιφόρνια τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris White. Ο Rubin χαρακτήρισε το έργο Android ως "τεράστιο δυναμικό για την ανάπτυξη έξυπνων κινητών συσκευών που έχουν μεγαλύτερη επίγνωση της θέσης και των προτιμήσεων του ιδιοκτήτη του". Οι πρώτες προθέσεις της εταιρείας ήταν να αναπτύξει ένα προηγμένο λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές και αυτή ήταν η βάση του πεδίου για τους επενδυτές τον Απρίλιο του 2004. Η εταιρεία αποφάσισε τότε ότι η αγορά των φωτογραφικών μηχανών δεν ήταν αρκετά μεγάλη για τους στόχους της, πέντε μήνες αργότερα είχε εκτρέψει τις προσπάθειές της και ανέβαξε το Android ως λειτουργικό σύστημα χειρός που θα ανταγωνιζόταν τα Symbian και τα Microsoft Windows Mobile.

Τον Ιούλιο του 2005, η Google απέκτησε το Android Inc. για τουλάχιστον 50 εκατομμύρια δολάρια. Προώθησε την πλατφόρμα στους κατασκευαστές και τους μεταφορείς με την υπόσχεση παροχής ενός ευέλικτου, αναβαθμίσιμου συστήματος. Η Google είχε "παρατάξει μια σειρά από εξαρτήματα υλικού και συνεργάτες λογισμικού και σημείωσε στους μεταφορείς ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας". Η Google άλλαξε αργότερα τα έγγραφα προδιαγραφών Android για να δηλώσει ότι θα υποστηρίζονται οι "Οθόνες αφής", αν και "το Προϊόν σχεδιάστηκε με την παρουσία

διακριτών φυσικών κουμπιών ως υπόθεση, επομένως μια οθόνη αφής δεν μπορεί να αντικαταστήσει πλήρως τα φυσικά κουμπιά"[23].

Στις 5 Νοεμβρίου 2007, η Open Handset Alliance, μια κοινοπραξία εταιρειών τεχνολογίας όπως η Google, κατασκευαστές συσκευών όπως HTC, Motorola και Samsung, ασύρματες εταιρείες όπως η Sprint και η T-Mobile και κατασκευαστές chipset όπως η Qualcomm και η Texas Instruments με στόχο την ανάπτυξη της "πρώτης πραγματικά ανοικτής και ολοκληρωμένης πλατφόρμας για κινητές συσκευές". Μέσα σε ένα χρόνο η Open Handset Alliance αντιμετώπισε δύο άλλους ανταγωνιστές ανοικτής πηγής, το Symbian Foundation και το LiMo Foundation, το τελευταίο αναπτύσσοντας επίσης ένα κινητό λειτουργικό σύστημα όπως το Google. Τον Σεπτέμβριο του 2007, το InformationWeek κάλυψε μια μελέτη Evaluateserve που ανέφερε ότι η Google είχε καταθέσει αρκετές αιτήσεις ευρεσιτεχνίας στον τομέα της κινητής τηλεφωνίας.

Από το 2008, το Android έχει δει πολλές αναβαθμίσεις, οι οποίες έχουν βελτιώσει σταδιακά το λειτουργικό σύστημα, προσθέτοντας νέα χαρακτηριστικά και διορθώνοντας σφάλματα σε προηγούμενες κυκλοφορίες. Κάθε μεγάλη κυκλοφορία ονομάζεται με αλφαβητική σειρά μετά από επιδόρπιο ή ζαχαρούχο σκεύασμα, με τις πρώτες εκδόσεις του Android να ονομάζονται "Cupcake", "Donut", "Eclair" και "Froyo", με αυτή τη σειρά. Κατά την ανακοίνωση του Android KitKat το 2013, η Google εξήγησε ότι "Δεδομένου ότι αυτές οι συσκευές κάνουν τη ζωή μας τόσο γλυκιά, κάθε έκδοση Android ονομάζεται μετά από ένα επιδόρπιο"[4][5].

2.3 Λογότυπο

Το λογότυπο του Android έχει αλλάξει τρεις φορές από το 2007. Το πρώτο λογότυπο φαίνεται στην παρακάτω εικόνα και χρησιμοποιήθηκε μέχρι το 2014[6].



Εικόνα 2- Το λογότυπο του Android από το 2007 έως το 2014

Όπου το 2014 έγινε ξανά αλλαγή στο λογότυπο και χρησιμοποιήθηκε για τα επόμενα 3 χρόνια. Το λογότυπο για την περίοδο 2014-2017 φαίνεται στη παρακάτω εικόνα.



Εικόνα 3- Το λογότυπο του Android από το 2014 έως το 2017

Το 2017 δέχτηκε την τελευταία του αλλαγή και παραμένει μέχρι σήμερα το ίδιο λογότυπο. Η παρακάτω εικόνα δείχνει την τωρινή του μορφή.



Εικόνα 4- Το λογότυπο του Android από το 2017

2.4 Χαρακτηριστικά και λειτουργίες

- Λειτουργίες οθόνης: Η πλατφόρμα είναι προσαρμόσιμη σε πολλές αναλύσεις οθόνης (από VGA μέχρι 4K), δισδιάστατες ψηφιακές γραφικές βιβλιοθήκες,

- τρισεδιάστατα γραφικά βασισμένα στην OpenGL ES 3.0+ έκδοση χαρακτηριστικών, καθώς και παραδοσιακές απεικονίσεις οθόνης "έξυπνων" συσκευών κινητής τηλεφωνίας.
- Αποθήκευση Δεδομένων: Χρήση βάσης δεδομένων SQLite για τις ανάγκες αποθήκευσης.
 - Συνδεσιμότητα: Το Android υποστηρίζει τεχνολογίες συνδεσιμότητας συμπεριλαμβανομένου GSM/LTE,3G,4G,CDMA,Bluetooth,NFC και Wi-Fi.
 - Αποστολή μηνυμάτων: SMS και MMS είναι οι διαθέσιμοι τρόποι ανταλλαγής μηνυμάτων.
 - Περιήγηση στο διαδίκτυο: Για την περιήγηση στον ιστό το Android διαθέτει φυλλομετρητή το Chrome. Και άλλοι φυλλομετρητές είναι διαθέσιμοι από το Google play.
 - Υποστήριξη Java: Λογισμικό γραμμένο στην Java είναι δυνατόν να μεταγλωττιστεί και να εκτελεστεί στην εικονική μηχανή Dalvik, η οποία αποτελεί εξειδικευμένη υλοποίηση εικονικής μηχανής, σχεδιασμένης για χρήση σε φορητές συσκευές.
 - Υποστήριξη υλικού: Μπορεί να συνεργαστεί με κάμερες στατικής ή κινούμενης εικόνας, οθόνες αφής, GPS, αισθητήρες επιτάχυνσης, μαγνητόμετρα, δισδιάστατους καθώς και τρισεδιάστατους επιταχυντές γραφικών.
 - Περιβάλλον ανάπτυξης λογισμικού: Περιλαμβάνει ένας προσομοιωτή συσκευής, εργαλεία για διόρθωση σφαλμάτων, μνήμη και εργαλεία ανάλυσης της απόδοσης του εκτελέσιμου λογισμικού καθώς και ένα επιπρόσθετο για το Android Studio.
 - Αγορά και εγκατάσταση εφαρμογών: Το Play Store είναι ένας κατάλογος εφαρμογών που μπορούν να μεταφορτωθούν και εγκατασταθούν στην συσκευή άμεσα μέσω ασύρματων καναλιών, χωρίς την χρήση υπολογιστή.
 - Οθόνη αφής πολλαπλών σημείων: Το λειτουργικό Android υποστηρίζει οθόνες αφής πολλαπλών σημείων.

2.5 Εκδόσεις Android

Ας δούμε τις εκδόσεις Android που κυκλοφόρησαν από το 2008 μέχρι το 2019 με χρονολογική σειρά. Συνολικά έχουν εκδοθεί 15 εκδόσεις του Android.



Εικόνα 5- Εκδόσεις του Android

2.5.1 Android 1.0 και Android 1.1



Εικόνα 6- Λογότυπο της έκδοσης Android 1.0

Με την κυκλοφορία του πρώτου Android smartphone το HTC Dream, έχουμε και την εμφάνιση της πρώτης έκδοσης 1.0 με ημερομηνία αρχικής κυκλοφορίας στις 23 Σεπτεμβρίου 2008. Ενσωματωμένες εφαρμογές που υπήρχαν ήταν το ξυπνητήρι, ο περιηγητής, η αριθμομηχανή, η κάμερα, το email, οι χάρτες και η μουσική καθώς και κάποιες άλλες.

2.5.2 Android 1.5 Cupcake



Εικόνα 7- Λογότυπο της έκδοσης Cupcake 1.5

Στις 27 Απριλίου 2009 κυκλοφόρησε η έκδοση CupCake. Και ουσιαστικά έχουμε και την αρχή των ονομάτων με γλυκά. Ήταν η έκδοση με την οποία έχουμε την υποστήριξη των widgets και ως νέα χαρακτηριστικά του ήταν η εγγραφή video και playback σε μορφή MPEG-4 και 3GP και τα εφέ κίνησης κατά την περιήγηση στις διαφορετικές οθόνες. Η έκδοση CupCake παρέχει τη δυνατότητα αυτόματης σύνδεσης ακουστικών headset σε συγκεκριμένη απόσταση, ανέβασμα εικόνων στο Picasa και βίντεο στο YouTube κατευθείαν από την κινητή συσκευή του χρήστη, ενώ παρέχει εικονικό πληκτρολόγιο με πρόβλεψη λέξεων και νέα widgets για την αρχική οθόνη. Διαθέτει κάμερα 5MP με αυτόματη εστίαση και κάρτα μνήμης micro SD.

2.5.3 Android 1.6 Donut



Donut
Android 1.6

Εικόνα 8- Λογότυπο της έκδοσης Donut 1.6

Το 15 Σεπτεμβρίου 2009 κυκλοφόρησε η έκδοση Donut. Εννοείτε ότι συμπεριλάμβανε νέες βελτιώσεις όπως ευκολότερη αναζήτηση και δυνατότητα προεπισκόπησης εφαρμογών σε όσες συσκευές είχαν Google Play, δείκτες χρήσης της μπαταρίας και αυτόματη περιστροφή οθόνης. Με το Donut δίνεται έμφαση στην αναζήτηση από την αρχική οθόνη με bookmarks, ιστορικό, επαφές αλλά και στη φωνητική αναζήτηση, ενώ υποστηρίζονται και οθόνες αναλύσεων WVGA. Τηλέφωνα της έκδοσης αυτής προσφέρει ευκολία στην εύρεση των επαφών, διαθέτει qwerty πληκτρολόγιο για γρήγορη αποστολή SMS και 3G και Wi-Fi για να είναι ο χρήστης συνδεδεμένος στο διαδίκτυο όπου και να βρίσκεται.

2.5.4 Android 2.0 Eclair



Eclair
Android 2.0/2.1

Εικόνα 9- Λογότυπο της έκδοσης Eclair 2.0

Στης 26 Οκτωβρίου 2009 κυκλοφόρησε η έκδοση Eclair 2.0, μόλις ένα μήνα μετά το Donut. Συμπεριλάμβανε αρκετές μικρές βελτιώσεις, όπως Bluetooth 2.1, κινούμενο φόντο στην οθόνη του home, και πληκτρολόγιο με έξυπνο λεξιλόγιο που μαθαίνει ανάλογα με την χρήση των λέξεων. Με το όνομα Éclair έχουμε και επόμενες επίσης γλυκές εκδόσεις, τις Éclair 2.0 και Éclair 2.1 που προχωρούν ακόμη πιο πολύ, διαθέτοντας υποστήριξη HTML5, νέο browser UI, Google Maps 3.1.2, ψηφιακό ζουμ, ενσωματωμένη υποστήριξη για flash στην κάμερα, βελτιωμένο εικονικό πληκτρολόγιο, δυνατότητα αντίληψης multi-touch, live wallpapers, και bluetooth 2.1.

2.5.5 Android 2.2-2.2.3 Froyo



Froyo
Android 2.2/2.2.3

Εικόνα 10- Λογότυπο της έκδοσης Froyo 2.2-2.2.3

Στης 20 Μαΐου 2010 κυκλοφόρησε και το ακρώνυμο της εκδόσεις αποτελεί συντόμευση της φράσης “Frozen Yogurt” (παγωμένο γιαούρτι). Είναι η πρώτη έκδοση του Android που υποστήριζε Adobe Flash. Μερικές από τις βελτιώσεις ήταν σύνδεση μέσω USB και Wi-Fi hotspot, η γρήγορη εναλλαγή γλώσσας κατά την πληκτρολόγηση και η δυνατότητα απενεργοποίησης της λειτουργίας δικτύου δεδομένων. Η έκδοση 2.2 (FroYo-Frozen Yogurt) αναβάθμισε αισθητά την ταχύτητα του OS, αλλά και τη γενικότερη απόδοση, παρέχει υποστήριξη Adobe flash 10.1 και επιλογή εγκατάστασης εφαρμογών στην κάρτα μνήμης, διαθέτει Market με δυνατότητα αυτόματων updates, ενσωματώνει τον Chrome V8 JavaScript στα browsers applications. Στη FroYo

βρίσκουμε και τη δυνατότητα χρήσης της συσκευής για διαμοιρασμό ίντερνετ μέσω Wi-Fi σε άλλες συσκευές (tethering).

2.5.6 Android 2.3-2.3.7 Gingerbread



Εικόνα 11- Λογότυπο της έκδοσης Gingerbread 2.3-2.3.7

Στης 6 Δεκεμβρίου 2010 κυκλοφόρησε η πιο πολυπληθής έκδοση του Android. Ήταν πολύ πιο γρήγορο και εύχρηστο από τις προηγούμενες εκδόσεις και έδινε στους δημιουργούς εφαρμογών μεγαλύτερες δυνατότητες. Οι βελτιώσεις περιλάμβαναν υποστήριξη πολλών καμερών στην συσκευή όπως και μεγαλύτερης ανάλυσης οθόνη. Η GingerBread (2.3), που κυκλοφόρησε υποστηρίζει πλέον πολύ μεγάλα μεγέθη οθονών και αναλύσεων, διαθέτει επανασχεδιασμένο multi-touch πληκτρολόγιο, προεγκατεστημένη υποστήριξη για τηλεφωνικές κλήσεις μέσω ίντερνετ (VoIP), download manager για κατέβασμα μεγάλων αρχείων, λειτουργίες copy-paste σε όλο το λειτουργικό, καθώς και προεγκατεστημένη υποστήριξη για πολλαπλές κάμερες.

2.5.7 Android 3.0-3.2 Honeycomb



Εικόνα 12- Λογότυπο της έκδοσης Honeycomb 3.0-3.2

Στης 22 Φεβρουαρίου 2011 κυκλοφόρησε και ήταν διαθέσιμη μόνο για tablets. Οι βελτιώσεις της περιλαμβάνουν γρήγορη πρόσβαση σε χαρακτηριστικά της κάμερας, καλύτερο πληκτρολόγιο κατάλληλο για μεγάλες οθόνες, εκτέλεση πολλαπλών λειτουργιών και εύκολη μετάβαση από την μια στην άλλη. Η έκδοση Honeycomb (3.1) προσέθεσε την επιλογή να μεταφέρεται περιεχόμενο απευθείας από συσκευές USB, ενώ τέλος η έκδοση 3.2 προσέθεσε διάφορες δυνατότητες και ευκολίες για χρήστες και developers όπως τη μεταφορά αρχείων από κάρτες SD και δυνατότητα Zoom to Fill.

2.5.8 Android 4.0-4.0.2 Ice Cream Sandwich



Εικόνα 13- Λογότυπο της έκδοσης Ice Cream Sandwich 4.0-4.0.2

Στης 18 Οκτωβρίου 2011 κυκλοφόρησε και έφερε πληθώρα αλλαγών στο λειτουργικό σύστημα. Η δυνατότητα χρήσης “μαλακών” κουμπιών δηλαδή των κουμπιών πάνω στην οθόνη (πίσω, αρχική) είναι πλέον πραγματικότητα καθώς μέχρι τότε όλα τα κινητά είχαν εξωτερικά κουμπιά για αυτές τις λειτουργίες. Μερικές άλλες δυνατότητες ήταν καλύτερη χρήση των φωνητικών εντολών, το Face Unlock, βελτίωση της ταχύτητας απόκρισης και αναδιαμορφώσει του περιβάλλοντος χρήσης. Αυτή η έκδοση διαθέτει καλύτερο web browser με tabs, ανανεωμένο γραφικό περιβάλλον με αρκετά 3D στοιχεία και ανανεωμένο εικονικό πληκτρολόγιο. Δίνει ειδική έκδοση του Gmail για tablets, δυνατότητα βιντεοκλήσεων μέσω εφαρμογής Google Talk, ανανεωμένη έκδοση Google Maps και βελτιστοποιημένη εφαρμογή για ανάγνωση Google e-books.

2.5.9 Android 4.1-4.3.1 Jelly Bean



Android 4.1 Jelly Bean

Εικόνα 14- Λογότυπο της έκδοσης Jelly Bean 4.1-4.3.1

Στης 9 Ιουλίου 2012 κυκλοφόρησε η έκδοση και αποτελεί την καλύτερη έκδοση του Android μέχρι σήμερα. Το περιβάλλον χρήσης και η απόκρισή του είναι πιο γρήγορη και καλοφτιαγμένη από ποτέ ενώ περιλαμβάνει πάρα πολλές μικρές βελτιώσεις σε όλο το σύστημα, όπως για παράδειγμα στην κάμερα και στην χρήση φωνής για υπαγόρευση κειμένου[7].

2.5.10 Android 4.4-4.4.4 Kit Kat



Εικόνα 15- Λογότυπο της έκδοσης Kit Kat 4.4-4.4.4

Στης 31 Οκτωβρίου 2013 κυκλοφόρησε η έκδοση KitKat. Αν και αρχικά ήταν να ονομαστεί "Key Lime Pie" ("KLP") κωδική ονομασία, το όνομα άλλαξε καθώς πολύ λίγοι άνθρωποι ξέρουν πραγματικά τη γεύση αυτής της πίτα. Το KitKat έκανε το ντεμπούτο του στο Nexus της Google 5 και έχει βελτιστοποιηθεί για να τρέχει σε ένα μεγαλύτερο εύρος συσκευών από τις προηγούμενες εκδόσεις του Android, αφού έχει ως συνιστώμενη ελάχιστη μνήμη RAM 512[8].

2.5.11 Android 5.0 Lollipop



Εικόνα 16- Λογότυπο της έκδοσης Lollipop 5.0

Στης 12 Νοεμβρίου 2014 ανακοινώθηκε από την Google η νέα έκδοση του Android για επιλεγμένες συσκευές που τρέχουν Android, συμπεριλαμβανομένων τις συσκευές Nexus. Η Lollipop ως βασική αλλαγή που θα παρουσιάσει είναι ένα επανασχεδιασμένο περιβάλλον εργασίας χρήστη χτισμένο γύρω από μία διαδραστική σχεδιαστική γλώσσα που αποκαλείται ως «υλικό σχεδιασμού». Επιπλέον βελτιώσεις του συστήματος θα είναι ότι το notifications system θα επιτρέπει κοινοποιήσεις που θα μπορούν να προσπελαστούν από την lockscreen, και να εμφανίζονται μαζί με εφαρμογές ως banner πάνω την κορυφή της οθόνης. Εσωτερικές αλλαγές που θα γίνουν επίσης στην πλατφόρμα, πιο συγκεκριμένα, το Android Runtime (ART) θα αντικαταστήσει το Dalvik με μία πιο βελτιωμένη έκδοση την γνωστή ως Project Volta, η οποία θα παρέχει καλύτερη απόδοση των εφαρμογών πράγμα το οποίο σημαίνει ότι θα υπάρξουν και αλλαγές που αποσκοπούν στη βελτίωση και βελτιστοποίηση της χρήσης της μπαταρίας[9].

2.5.12 Android 6.0-6.0.1 Marshmallow



Εικόνα 17- Λογότυπο της έκδοσης Marshmallow 6.0-6.0.1

Το Marshmallow κυκλοφόρησε στις 5 Οκτωβρίου 2015. Επικεντρώνεται κυρίως στη βελτίωση της συνολικής εμπειρίας του χρήστη σε σύγκριση με τον προκάτοχο του, Lollipop, εισήγαγε μια νέα αρχιτεκτονική στα δικαιώματα των εφαρμογών, νέα

APIs για συμφραζόμενα βοηθούς (που χρησιμοποιείται για πρώτη φορά από ένα νέο χαρακτηριστικό το "Now on Tap" που παρέχει το πλαίσιο με τα ευαίσθητα αποτελέσματα αναζήτησης), ένα νέο σύστημα διαχείρισης ενέργειας που μειώνει την δραστηριότητα του παρασκηνίου όταν μια συσκευή δεν χρησιμοποιείται, εγγενή υποστήριξη για την αναγνώριση δακτυλικών αποτυπωμάτων και USB τύπου-C, τη δυνατότητα να μεταφοράς των δεδομένων και τις εφαρμογές σε μια κάρτα microSD, και άλλες εσωτερικές αλλαγές[10].

2.5.13 Android 7.0-7.1.1 Nougat



Εικόνα 18- Λογότυπο της έκδοσης Nougat 7.0-7.1.1

Το Nougat κυκλοφόρησε στις 22 Αυγούστου 2016. Εισάγει σημαντικές αλλαγές στο λειτουργικό σύστημα και την πλατφόρμα ανάπτυξης του, συμπεριλαμβανομένης της δυνατότητας να εμφανίσει πολλαπλές εφαρμογές στην οθόνη με προβολή διαίρεσης της οθόνης, υποστήριξη για τις απαντήσεις ενσωματωμένη με τις κοινοποιήσεις, καθώς και περιβάλλον Java βασισμένο στο OpenJDK αλλά και υποστήριξη για τις απόδοσης γραφικών Vulkan API, και απρόσκοπτες ενημερώσεις συστήματος για υποστηριζόμενες συσκευές[11].

2.5.14 Android Oreo



Εικόνα 19- Λογότυπο της έκδοσης Android Oreo

Το Android "Oreo" (με την κωδική ονομασία Android O κατά την ανάπτυξη) είναι η όγδοη κατά σειρά έκδοση του λειτουργικού συστήματος Android. Πρώτα κυκλοφόρησε ως έκδοση ανάπτυξης στις 21 Μαρτίου 2017 και τελικά κυκλοφόρησε για το κοινό στις 21 Αυγούστου[12].

2.5.15 Android Pie



Εικόνα 20- Λογότυπο της έκδοσης Android Pie

Το Android "Pie" (γνωστό με την κωδική ονομασία Android P κατά τη διάρκεια της ανάπτυξης) είναι η ένατη κύρια ενημέρωση και η 16η έκδοση του λειτουργικού συστήματος Android. Το Android "P" κυκλοφόρησε επίσημα στις 6 Αυγούστου 2018 ως "Android 9 Pie". Όπως δηλώνει η Google, το Android 9 Pie προσαρμόζεται και μαθαίνει τις συνήθειες του χρήστη εκμεταλλευόμενο τις δυνατότητες της Τεχνητής Νοημοσύνης για μια εξυπνότερη, γρηγορότερη και καλύτερη εμπειρία χρήσης για πολύ καιρό[13].

2.6 Έκδοση της εφαρμογής.

Η κατώτερη έκδοση που πρέπει να έχει ένα κινητό για να μπορεί να εγκατασταθεί η εφαρμογή Pharmacy είναι το Lollipop 5.0. Η επιλογή της έκδοσης ήταν δύσκολη απόφαση, για αυτόν τον λόγο ορίστηκαν δύο βασικοί παράγοντες. Ο πρώτος παράγοντας είναι να μπορεί να χρησιμοποιηθεί από το μεγαλύτερο εύρος των κινητών έτσι ώστε να υπάρχουν περισσότεροι πιθανοί χρήστες της εφαρμογής. Σύμφωνα με το Android Studio το 85% χρησιμοποιεί αυτήν την έκδοση. Ο δεύτερος παράγοντας ήταν να μην καταναλώνει πολύ μπαταρία και να μην επιβαρύνει τις επιδόσεις του κινητού. Βάση αυτών, η έκδοση Lollipop ήταν η μόνη επιλογή που υπήρχε.

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99.6%
4.2 Jelly Bean	17	98.1%
4.3 Jelly Bean	18	95.9%
4.4 KitKat	19	95.3%
5.0 Lollipop	21	85.0%
5.1 Lollipop	22	80.2%
6.0 Marshmallow	23	62.6%
7.0 Nougat	24	37.1%
7.1 Nougat	25	14.2%
8.0 Oreo	26	6.0%
8.1 Oreo	27	1.1%

Εικόνα 21- Ποσοστά χρήσης εκδόσεων σε android κινητά.

3. Τεχνολογίες που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο παρουσιάζονται όλες οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Για την καλύτερη κατανόηση της περιγραφής του κώδικα που θα γίνει στα επόμενα κεφάλαια.

3.1 Kotlin



Εικόνα 22- Λογότυπο της γλώσσας Kotlin.

Η Kotlin είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που σχεδιάστηκε από την ομάδα JetBrains. Είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα (OP), γενικής χρήσης που συνδυάζει τόσο αντικειμενοστραφή όσο και συναρτησιακά χαρακτηριστικά. Ως εκ τούτου, απαντά τα περισσότερα από τα βασικά ζητήματα που εμφανίστηκαν στη Java, παρέχοντας στους προγραμματιστές διαλειτουργικότητα, ασφάλεια, σαφήνεια και υποστήριξη πολλών εργαλείων. Ο ενθουσιασμός γύρω από την Kotlin δημιουργείται από το γεγονός ότι η Google την χαρακτήρισε επίσημα ως την γλώσσα για την ανάπτυξη εφαρμογών Android. Η Kotlin είναι επίσης συνεργάσιμη με τις υπάρχουσες γλώσσες Android. Είναι μια γλώσσα με

πολλές ομοιότητες με τη Java. Όμως, σε σύγκριση με τη Java έχει πιο καθαρή σύνταξη, προσθέτει κάποια ωραία χαρακτηριστικά, παίρνει ιδέες από τον συναρτησιακό προγραμματισμό και έχει άλλες βελτιώσεις[14].

3.2 PHP



Εικόνα 22- Το λογότυπο της PHP

Η PHP είναι μια σεναριακή γλώσσα προγραμματισμού (scripting language), που είναι κατάλληλη για δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο. Όπως κάθε άλλη σεναριακή γλώσσα προγραμματισμού, επιτρέπει στους προγραμματιστές να δημιουργήσουν την λογική σε δημιουργία περιεχομένου ιστοσελίδων και να χειριστούν τα δεδομένα που επιστρέφονται από ένα πρόγραμμα περιήγησης ιστού. Η PHP περιέχει επίσης μια σειρά από επεκτάσεις που διευκολύνουν την αλληλεπίδραση με βάσεις δεδομένων, την εξαγωγή δεδομένων για την εμφάνιση τους σε μια εφαρμογή. Η PHP μπορεί να δημιουργήσει, να ανοίξει, να διαβάσει, να γράψει, να διαγράψει και να κλείσει αρχεία. Μπορεί να λειτουργήσει σε διάφορες πλατφόρμες όπως Windows, Linux, Unix, Mac OS X, είναι συμβατή με σχεδόν όλους τους servers που χρησιμοποιούνται σήμερα όπως Apache, IIS και υποστηρίζει ένα ευρύ φάσμα βάσεων δεδομένων[15].

3.3 MySQL



Εικόνα 23- Το λογότυπο της MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοιχτού κώδικα που λειτουργεί σε πολλές πλατφόρμες . Παρέχει πρόσβαση σε πολλούς χρήστες για την υποστήριξη πολλών μηχανών αποθήκευσης και υποστηρίζεται από την Oracle. Η MySQL χρησιμοποιείται από πολλές εφαρμογές που βασίζονται σε βάσεις δεδομένων. Χρησιμοποιείται επίσης από πολλούς δημοφιλείς ιστότοπους, όπως το Facebook, το Twitter, το Flickr και το YouTube. Υπάρχει ευκολία διαχείρισης διότι η λήψη του λογισμικού είναι πολύ εύκολη και χρησιμοποιεί επίσης έναν προγραμματιστή συμβάντων για να προγραμματίσει αυτόματα τις εργασίες. Διατηρεί την ιδιότητα ACID (Ατομικότητα, Συνέπεια, Απομόνωση, Ανθεκτικότητα) και επίσης επιτρέπει την κατανεμημένη υποστήριξη πολλαπλών εκδόσεων.

Η MySQL επίσης διαθέτει plugin βιβλιοθήκες για να ενσωματώνει τη βάση δεδομένων σε οποιαδήποτε εφαρμογή. Υποστηρίζει επίσης αποθηκευμένες διαδικασίες, ενεργοποιήσεις, λειτουργίες, προβολές και πολλά άλλα για την ανάπτυξη εφαρμογών. Υποστηρίζει ισχυρούς μηχανισμούς που διασφαλίζουν ότι μόνο οι εξουσιοδοτημένοι χρήστες έχουν πρόσβαση στις βάσεις δεδομένων. Σχεδιασμένο να είναι πλήρως multithreaded χρησιμοποιώντας kernel threads, για εύκολη χρήση πολλαπλών CPU, αν

είναι διαθέσιμα και χρησιμοποιεί ένα πολύ γρήγορο thread-based σύστημα κατανομής μνήμης[16].

3.3 RSS



Εικόνα 24- Το λογότυπο του RSS

Το ακρωνύμιο RSS, από τον αγγλικό όρο Rich Site Summary (Σύνοψη Πλουσίας Σελίδας), αναφέρεται σε μία προτυποποιημένη μέθοδο ανταλλαγής ψηφιακού περιεχομένου διαμέσου του διαδικτύου, στηριγμένη στην πρότυπη, καθιερωμένη και ευρέως υποστηριζόμενη γλώσσα σήμανσης XML. Ένας χρήστης μπορεί έτσι να ενημερώνεται αυτομάτως για γεγονότα και νέα από όσες ιστοσελίδες υποστηρίζουν RSS, αρκεί να έχει εγγραφεί ο ίδιος συνδρομητής στην αντίστοιχη υπηρεσία της εκάστοτε ιστοσελίδας. Οι εν λόγω ενημερώσεις περιέχουν τα πλήρη δεδομένα, σύνοψη των δεδομένων, σχετικά μεταδεδομένα, ημερομηνία έκδοσης, ενώ αποστέλλονται αυτομάτως στον συνδρομητή μέσω διαδικτύου.

Το πρότυπο RSS υπάρχει από το 1999, ωστόσο παλαιότερες, όχι τόσο επιτυχημένες τεχνολογίες παρόμοιου σκοπού (οι οποίες βέβαια δεν αξιοποιούσαν την καθιερωμένη σήμερα γλώσσα XML, καθώς αυτή δεν είχε εμφανιστεί πριν το 1998) κυκλοφορούσαν από τα μέσα της δεκαετία του 1990.

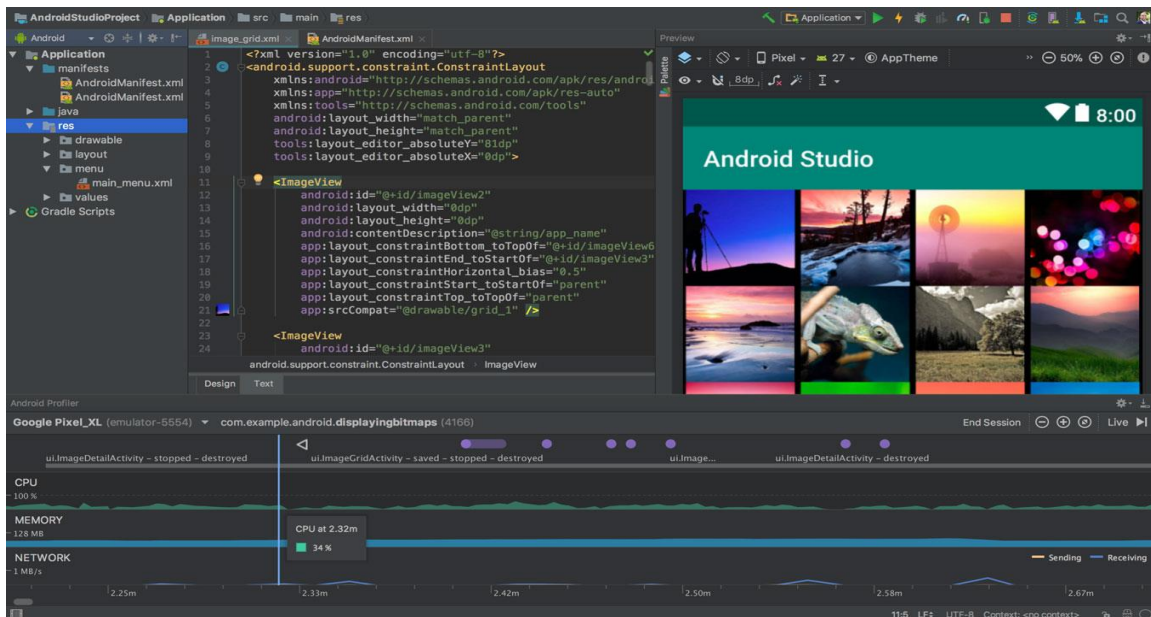
Συνήθως η διαδικασία απαιτεί ένα ειδικό λογισμικό «αναγνώστη» (το οποίο μπορεί να είναι ενσωματωμένο στον πλοηγό Web του χρήστη), όπου κάποιος πληκτρολογεί τη διεύθυνση URL της ζητούμενης ροής μιας ιστοσελίδας, ώστε να εγγραφεί σε αυτήν ως συνδρομητής. Ο αναγνώστης ελέγχει τακτικά για ενημερώσεις περιεχομένου τις καταγεγραμμένες ροές, από όλες τις διαφορετικές πηγές στις οποίες είναι εγγεγραμμένος ο συνδρομητής, αναλαμβάνοντας από μόνος του να μεταφέρει τις εν

λόγω ενημερώσεις όποτε χρειάζεται και να τις συγκεντρώνει στον υπολογιστή του χρήστη. Ο αναγνώστης παρέχει στον εκάστοτε χρήστη μία κοινή και εξατομικευμένη διασύνδεση για όλες τις ροές που ο ίδιος έχει επιλέξει να παρακολουθεί.

4. Εργαλεία που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο παρουσιάζονται όλα τα λογισμικά και πλατφόρμες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Το Android studio χρησιμοποιήθηκε για την δημιουργία της εφαρμογής Pharmacy, το Champp για την διαχείριση των στοιχείων της βάσης δεδομένων. Η Firebase χρησιμοποιήθηκε για την δημιουργία μιας βάσης που θα δέχεται δεδομένα και θα τα αποθηκεύει. Επίσης χρησιμοποιήθηκε το Photoshop για την επεξεργασία των εικονιδίων που υπάρχουν στην εφαρμογή και το Zapier για να γίνει η αυτοματοποίηση των νέων άρθρων.

4.1 Android Studio



Εικόνα 25- Περιβάλλον χρήσης του Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (Integrated Development Environment - IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Επίσης είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0 και έχει αναπτυχθεί με βάση το IntelliJ IDEA. Παρέχει στους προγραμματιστές έναν επεξεργαστή κειμένου, δυνατότητα αποσφαλμάτωσης και προσομοιώνει λειτουργίες, όπως η τοποθεσία GPS, η καθυστέρηση δικτύου, οι αισθητήρες κίνησης και η δυνατότητα πολλαπλής αφής. Επιπλέον, παρέχει emulator δίνοντας στους προγραμματιστές την δυνατότητα δοκιμής των εφαρμογών τους σε διάφορα κινητά, tablets, Android Wear και συσκευές Android TV[18].

4.2 Xampp



XAMPP

Εικόνα 26- Το λογότυπο του Xampp

Το XAMPP είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, λογισμικού ανοικτού κώδικα και ανεξαρτήτου πλατφόρμας, το οποίο περιέχει το εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl. Επίσης είναι ελεύθερο λογισμικό ως προς την χρήση και τρέχει σε Microsoft Windows, Linux, Solaris, and Mac OS X[19]. Το XAMPP είναι ακρωνύμιο και αναφέρεται στα παρακάτω αρχικά:

- **X** αναφέρεται στο «cross-platform» που σημαίνει λογισμικό ανεξάρτητο πλατφόρμας

- Apache HTTP εξυπηρετητής
- MySQL
- PHP
- Perl

4.3 Firebase



Εικόνα 27- Το λογότυπο του Firebase

Το Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητά και ιστού που αναπτύχθηκε από την Firebase το 2011 και στη συνέχεια εξαγοράστηκε από την Google το 2014. Η βάση δεδομένων Firebase Realtime είναι μια βάση δεδομένων NoSQL που φιλοξενεί το cloud και μας επιτρέπει να αποθηκεύουμε και να συγχρονίζουμε τους χρήστες σε πραγματικό χρόνο. Η βάση δεδομένων Realtime είναι ένα μεγάλο αντικείμενο JSON το οποίο οι προγραμματιστές μπορούν να διαχειριστούν σε πραγματικό χρόνο [20].

4.4 Adobe Photoshop



Εικόνα 28- Το λογότυπο του Adobe Photoshop

Το Adobe Photoshop είναι ένα πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems. Αυτή τη στιγμή αποτελεί ηγέτη της αγοράς των προγραμμάτων επεξεργασίας εικόνων, και είναι το προϊόν - σήμα κατατεθέν της Adobe Systems [21].

4.5 Zapier



Εικόνα 29- Το λογότυπο του Zapier

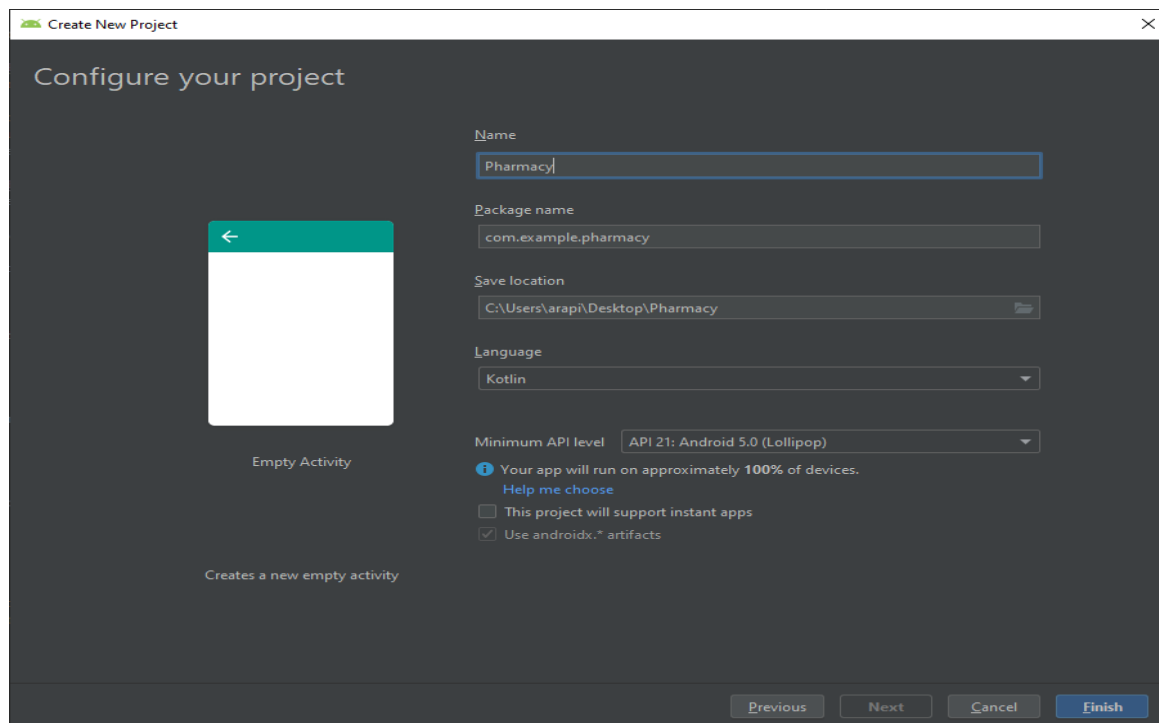
Το Zapier είναι ένα εργαλείο ηλεκτρονικής αυτοματοποίησης που συνδέει τις εφαρμογές, όπως το Gmail, το Slack, το Mailchimp και πολλά άλλα. Μπορείς να συνδέσεις δύο ή περισσότερες εφαρμογές για να αυτοματοποιήσεις επαναλαμβανόμενες εργασίες χωρίς να κωδικοποιήσεις. Είναι αρκετά εύκολο να μπορεί κανείς να δημιουργήσει τις δικές του ροές εργασίας εφαρμογών με μερικά μόνο κλικ και να ενημερώνεται όταν υπάρξει αλλαγή από την τρέχουσα κατάσταση[22].

5 Παρουσίαση της εφαρμογής Pharmacy

Σε αυτό το κεφάλαιο θα γίνει η παρουσίαση και ανάλυση της εφαρμογής με σχετικές εικόνες και σχόλια για την κατανόηση του κώδικα και πως λειτουργεί η εφαρμογή.

5.1 Δημιουργία της εφαρμογής

Αρχικά θα πρέπει να δημιουργήσουμε το project και να επιλέξουμε την κατώτερη έκδοση στην οποία θα λειτουργεί η εφαρμογή. Επίσης επιλέγουμε το όνομα της εφαρμογής και την γλώσσα προγραμματισμού στην οποία θα προγραμματίσουμε. Η γλώσσα που θεώρησα καλύτερη για την υλοποίηση της εφαρμογής είναι η Kotlin, με έκδοση λειτουργικού συστήματος το Lollipop. Και για όνομα της εφαρμογής διάλεξα το Pharmacy. Γιατί σαν όνομα για την είναι αντιπροσωπευτικό για τις λειτουργίες της εφαρμογής και δεν ξεχνιέται εύκολα.



Εικόνα 30- Επιλογές συστήματος

Η εφαρμογή για να μπορεί να βρει την τοποθεσία του κινητού, να καλέσει και για να έχει πρόσβαση στο internet θα πρέπει να ζητήσει την άδεια από τον χρήστη. Αυτό γίνεται στο AndroidManifest.xml με το παρακάτω κώδικα.

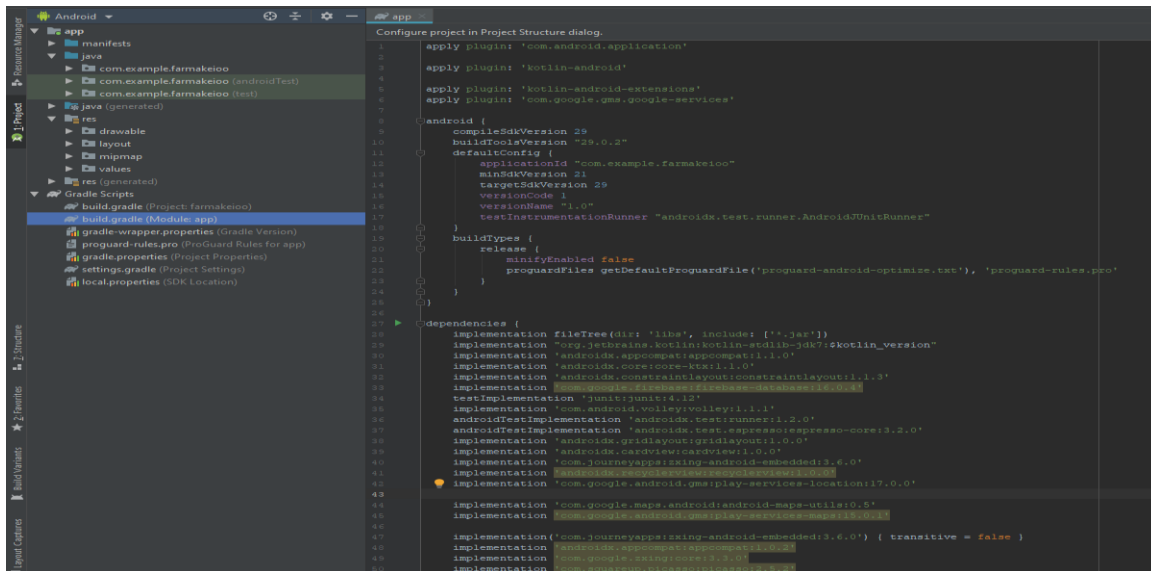
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Εικόνα 31- Αιτήσεις πρόσβασης λειτουργιών

- `Android.permission.INTERNET` ζητάει δικαίωμα χρήσης internet είτε από δεδομένα κινητής τηλεφωνίας είτε από χρήση του Wi-Fi.
- `Android.permission.CALL_PHONE` ζητάει δικαιώματα χρήσης για να μπορεί να καλέσει έναν αριθμό.
- `Android.permission.ACCESS_COARSE_LOCATION` ζητάει δικαίωμα για πρόσβαση στην κατά προσέγγιση τοποθεσία της συσκευής.
- `Android.permission.ACCESS_FINE_LOCATION` ζητάει δικαίωμα για πρόσβαση στην ακριβή τοποθεσία της συσκευής.

5.2 Βιβλιοθήκες

Το Android Studio έχει τοποθετήσει τις βασικές βιβλιοθήκες για την λειτουργία της εφαρμογής. Αυτές βρίσκονται στο `build.gradle(Module.app)`, όπου μπορείς να προσθέσεις καινούργιες βιβλιοθήκες που υπάρχουν ήδη για να χρησιμοποιήσεις τις λειτουργίες τους.



Εικόνα 32- Εγκατάσταση βιβλιοθηκών στο build.gradle(Module.app)

Στην εφαρμογή έχουν προστεθεί οι παρακάτω βιβλιοθήκες:

- Picasso για να μπορεί να εμφανίσει τις φωτογραφίες από την βάση δεδομένων.
- Firebase για να μπορεί να συγχρονισθεί με την βάση δεδομένων.
- RecyclerView για να μπορεί να χρησιμοποιήσει recyclerview λίστα.
- Zxing για να χρησιμοποιήσει το barcode scanner.

5.3 Κλάσεις

5.3.1 Κλάση MainActivity

Αρχικά στην κλάση βρίσκει την τοποθεσία του χρήστη με την συνάρτηση getLastLocation. Η συνάρτηση onActivityResult είναι για την λειτουργία του barcode scanner όταν ο χρήστης θέλει να το χρησιμοποιήσει. Μόλις σκανάρει ένα barcode το αναζητάει στην ηλεκτρονική σελίδα www.galinos.gr, όπου υπάρχουν όλες οι πληροφορίες για το φάρμακο.

```

1 package com.example.farmakeioo
2
3 import ...
4
5 class MainActivity : AppCompatActivity() {
6
7     lateinit var mFusedLocationClient: FusedLocationProviderClient
8
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12
13         mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
14         getLastLocation()
15
16         hospital.setOnClickListener { @View()
17             val intent= Intent ( packageContext: this,ActivityHospitals::class.java)
18             startActivity(intent)
19         }
20         pharmacy.setOnClickListener { @View()
21             val intent= Intent ( packageContext: this,ViewPharmacyActivity::class.java)
22             startActivity(intent)
23         }
24         articles.setOnClickListener { @View()
25             val intent= Intent ( packageContext: this,ActivityArticles::class.java)
26             startActivity(intent)
27         }
28         scanner.setOnClickListener { @View()
29             val scanner = IntentIntegrator ( activity: this)
30             scanner.initiateScan()
31         }
32     }
33
34     //barcode scanner
35     override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
36         if (resultCode== Activity.RESULT_OK)
37         {
38             val result :IntentResult? = IntentIntegrator.parseActivityResult(requestCode, resultCode, data)
39             if (result != null) {
40                 if (result.contents == null) {
41                     Toast.makeText ( context: this, text: "Cancelled", Toast.LENGTH_LONG).show()
42                 } else {
43                     val url1 = "https://www.galinos.gr/web/drugs/main/packages/"
44                     startActivity(Intent(Intent.ACTION_VIEW, Uri.parse( urlString: url1+result.getContents())))
45                 }
46             } else {
47                 super.onActivityResult(requestCode, resultCode, data)
48             }
49         }
50     }
51 }

```

Εικόνα 33-Εικόνα από την κλάση MainActivity

```

73     }
74 }
75
76 fun pro1(glo1:Double)
77 {
78     pla=glo1
79     platos_hospital=glo1
80 }
81 fun pro2(glo2:Double)
82 {
83     mka=glo2
84     mhkos_hospital=glo2
85 }
86 fun getLastLocation() {
87
88     if (checkPermissions()) {
89         Log.d( tag: "checkPermissions",checkPermissions().toString())
90         if (isLocationEnabled()) {
91             Log.d( tag: "isLocationEnabled",isLocationEnabled().toString())
92             mFusedLocationClient.getLastLocation().addOnCompleteListener(this){ task ->
93                 var location: Location? = task.result
94                 if (location != null) {
95                     requestNewLocationData()
96                     Log.d( tag: "requestNewLocationData",requestNewLocationData().toString())
97                 } else {
98                     var globalVar1 :Double =location.latitude
99                     var globalVar2 :Double =location.longitude
100
101                     pro1(globalVar1)
102                     pro2(globalVar2)
103                 }
104             } else
105             {
106                 Toast.makeText ( context: this, text: "Turn on location", Toast.LENGTH_LONG).show()
107                 val intent = Intent (Settings.ACTION_LOCATION_SOURCE_SETTINGS)
108                 startActivity(intent)
109             }
110         } else
111         {
112             requestPermissions()
113             Log.d( tag: "requestPermissions",requestPermissions().toString())
114         }
115     }
116 }
117 private fun requestNewLocationData() {
118
119     var mLocationRequest = LocationRequest()
120     mLocationRequest.priority = LocationRequest.PRIORITY_HIGH_ACCURACY
121     mLocationRequest.interval = 0
122     mLocationRequest.fastestInterval = 0

```

Εικόνα 34-Εικόνα από την κλάση MainActivity

```
124         mLocationRequest.numUpdates = 1
125
126         mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this)
127         mFusedLocationClient.requestLocationUpdates(
128             mLocationRequest, mLocationCallback,
129             Looper.myLooper()
130         )
131     }
132     private val mLocationCallback: LocationCallback = object : LocationCallback() {
133     of override fun onLocationResult(locationResult: LocationResult) {
134         var mLastLocation: Location = locationResult.lastLocation
135         var globalVar1 : Double = mLastLocation.latitude
136         var globalVar2 : Double = mLastLocation.longitude
137         pro1(globalVar1)
138         pro2(globalVar2)
139     }
140     }
141 }
142
143 private fun isLocationEnabled(): Boolean {
144     var locationManager: LocationManager = getSystemService(Context.LOCATION_SERVICE) as LocationManager
145     return locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) || locationManager.isProviderEnabled(
146         locationManager.NETWORK_PROVIDER
147     )
148 }
149 private fun checkPermissions(): Boolean {
150     if (ActivityCompat.checkSelfPermission(
151         context: this, Manifest.permission.ACCESS_COARSE_LOCATION) == PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
152         context: this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED
153     ) {
154         return true
155     }
156     return false
157 }
158 private fun requestPermissions() {
159     ActivityCompat.requestPermissions(
160         activity: this,
161         arrayOf(
162             Manifest.permission.ACCESS_COARSE_LOCATION,
163             Manifest.permission.ACCESS_FINE_LOCATION
164         ),
165         PERMISSION_ID
166     )
167 }
168
169 of override fun onRequestPermissionsResult(
170     requestCode: Int,
171     permissions: Array<String>,
172     grantResults: IntArray
173 ) {
```

Εικόνα 35-Εικόνα από την κλάση MainActivity

```
174         if (requestCode == PERMISSION_ID) {
175             if ((grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED))
176                 getLastLocation()
177         }
178     }
179 }
180
181 }
```

Εικόνα 36-Εικόνα από την κλάση MainActivity

Στην συνέχεια υπάρχει στιγμιότυπο από την εφαρμογή, όπου ο χρήστης μπορεί να διαλέξει ποια λειτουργία της εφαρμογής θα χρησιμοποιήσει.

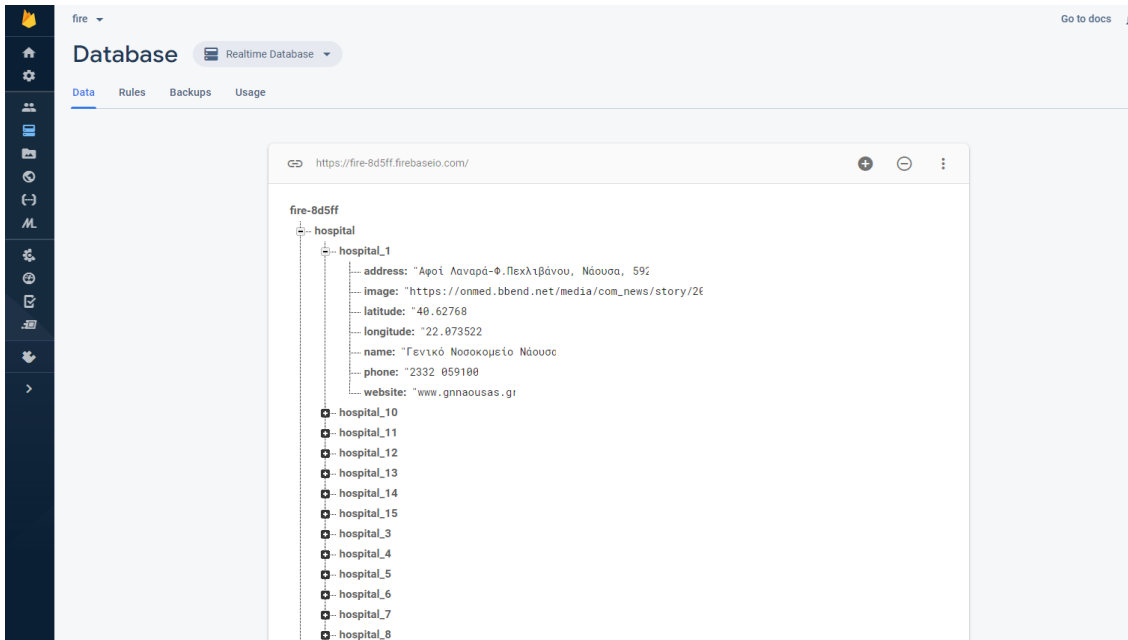


Εικόνα 37-Εικόνα από την MainActivity

5.3.2 Κλάση ActivityHospitals

Στην ActivityHospitals η συνάρτηση loadDatabase φορτώνει τα στοιχεία των νοσοκομείων στον πίνακα products. Αυτό γίνεται με την βοήθεια της data κλάσης Hospital και της MyListHospital η οποία είναι η κλάση που προβάλλει τα στοιχεία των

νοσοκομείων. Στις επόμενες εικόνες θα παρουσιαστούν η βάση δεδομένων, ο κώδικας για να προβληθούν τα νοσοκομεία και στιγμιότυπα από την χρήση της εφαρμογής.



Εικόνα 38-Εικόνα από την βάση δεδομένων.

```
1 package com.example.farmakeioo
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import com.google.firebase.database.*
6 import kotlinx.android.synthetic.main.activity_articles.*
7 import kotlin.math.round
8
9 class ActivityHospitals : AppCompatActivity() {
10     private lateinit var database: DatabaseReference
11     companion object
12     {
13         var platos_hospital = 0.0
14         var mhkos_hospital = 0.0
15     }
16
17     val products : ArrayList<Hospital> = arrayListOf<Hospital>()
18     val name : MutableList<String> = mutableListOf<String>()
19     val address : MutableList<String> = mutableListOf<String>()
20     val imageId : MutableList<String> = mutableListOf<String>()
21     val phone : ArrayList<String> = arrayListOf<String>()
22     val website : ArrayList<String> = arrayListOf<String>()
23     val latitude : ArrayList<String> = arrayListOf<String>()
24     val longitude : ArrayList<String> = arrayListOf<String>()
25     val distance : ArrayList<String> = arrayListOf<String>()
26
27
28     override fun onCreate(savedInstanceState: Bundle?) {
29         super.onCreate(savedInstanceState)
30         setContentView(R.layout.activity_hospitals)
31         database = FirebaseDatabase.getInstance().reference
32
33         loadDatabase()
34     }
35 }
```

Εικόνα 39-Εικόνα από την ActivityHospitals

```

36 private fun loadDatabase() {
37
38     val ref : DatabaseReference = FirebaseDatabase.getInstance().getReference("hospital")
39     ref.addValueEventListener(object : ValueEventListener {
40     override fun onDataChange(dataSnapshot: DataSnapshot) {
41         products.clear()
42         name.clear()
43         address.clear()
44         imageId.clear()
45         phone.clear()
46         website.clear()
47         latitude.clear()
48         longitude.clear()
49
50         for (productSnapshot : DataSnapshot in dataSnapshot.children) {
51             val product : Hospital? = productSnapshot.getValue(Hospital::class.java)
52             products.add(product!!)
53         }
54
55         var i=0
56
57         while (i<products.size)
58         {
59             name.add( products[i].name)
60             address.add(products[i].address)
61             imageId.add(products[i].image)
62             phone.add(products[i].phone)
63             website.add(products[i].website)
64             latitude.add(products[i].latitude)
65             longitude.add(products[i].longitude)
66             distance.add(distance(latitude[i].toDouble(),longitude[i].toDouble()), platos hospita, mhkos hospital).toString())
67
68             i++
69         }
70     }

```

Εικόνα 40-Εικόνα από την ActivityHospitals

```

71
72     for (i :Int in 0 until distance.size)
73     for (j :Int in 0 until distance.size)
74     if (distance[i]<distance[j])
75     {
76         var t :String =distance[i]
77         distance[i]=distance[j]
78         distance[j]=t
79
80         t=name[i]
81         name[i]=name[j]
82         name[j]=t
83
84         t=address[i]
85         address[i]=address[j]
86         address[j]=t
87
88         t=imageId[i]
89         imageId[i]=imageId[j]
90         imageId[j]=t
91
92         t=phone[i]
93         phone[i]=phone[j]
94         phone[j]=t
95
96         t=website[i]
97         website[i]=website[j]
98         website[j]=t
99
100        t=latitude[i]
101        latitude[i]=latitude[j]
102        latitude[j]=t
103
104        t=longitude[i]
105        longitude[i]=longitude[j]
106        longitude[j]=t
107    }
108
109

```

Εικόνα 41-Εικόνα από την ActivityHospitals


```

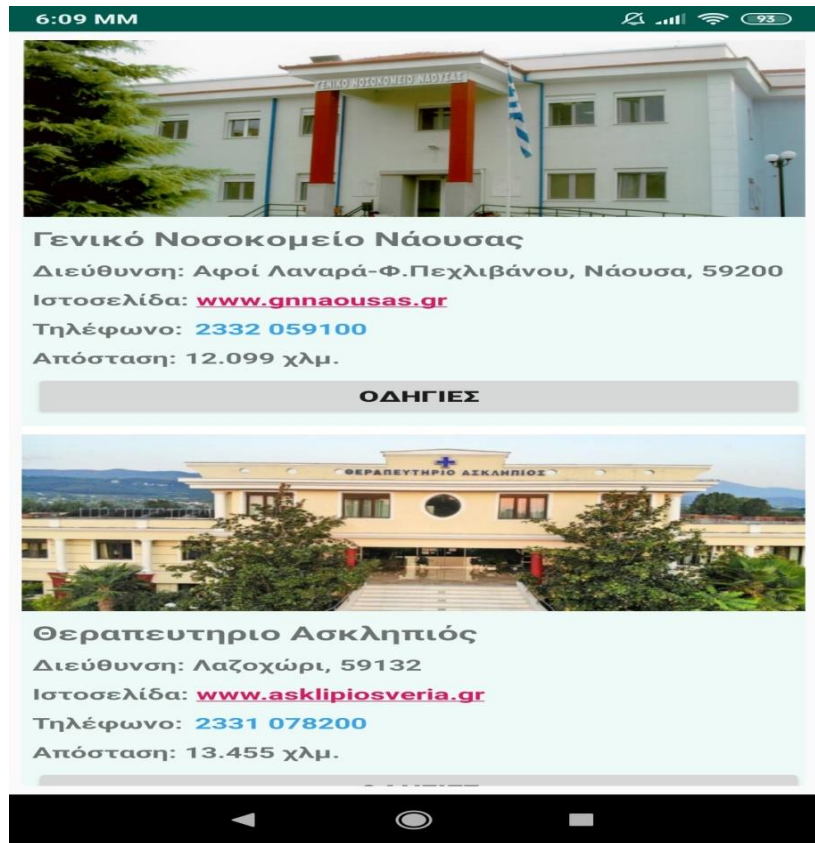
110         val myListAdapter = MyListHospital(context: this@ActivityHospitals, name, address, imageId, phone, website, latitude, longitude, distance)
111         listView.adapter = myListAdapter
112     }
113     override fun onCancelled(databaseError: DatabaseError) {
114         throw databaseError.toException()
115     }
116 }
117 }
118 }
119 private fun distance(lat1: Double, lon1: Double, lat2: Double, lon2: Double): Double {
120     val theta :Double = lon1 - lon2
121     var dist :Double = Math.sin(Math.toRadians(lat1)) * Math.sin(Math.toRadians(lat2)) + Math.cos(
122         Math.toRadians(lat1)
123     ) * Math.cos(Math.toRadians(lat2)) * Math.cos(Math.toRadians(theta))
124     dist = Math.acos(dist)
125     dist = Math.toDegrees(dist)
126     dist = dist * 60 * 1.1515
127     dist = dist * 1.609344
128     dist = round(( * dist*1000) /1000
129     return (dist)
130 }
131 }

```

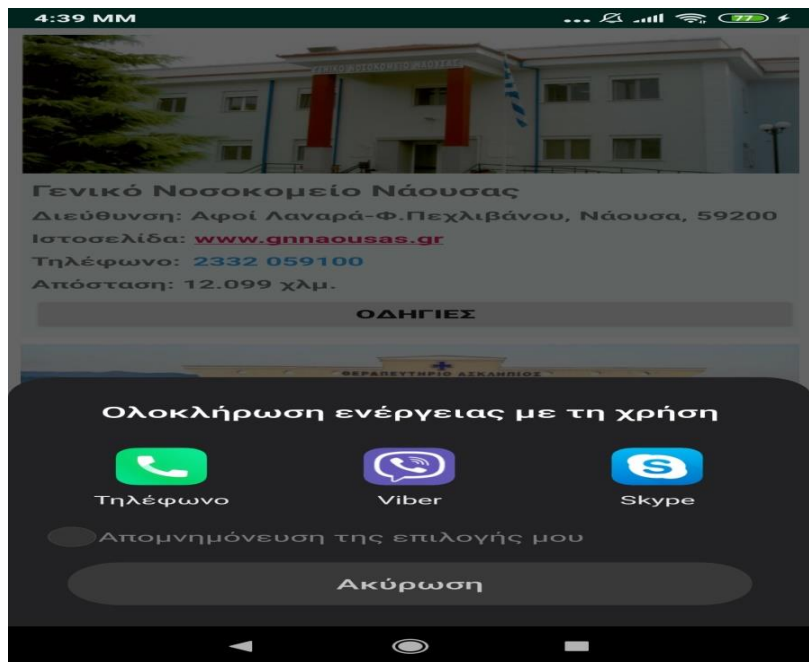
Εικόνα 42-Εικόνα από την ActivityHospitals

Η λειτουργία για τα νοσοκομεία φαίνεται στην παρακάτω εικόνα όπου ο χρήστης μπορεί να βρει νοσοκομεία του νομού Μακεδονίας. Οι πληροφορίες που μπορεί να μάθει ο χρήστης είναι οι εξής:

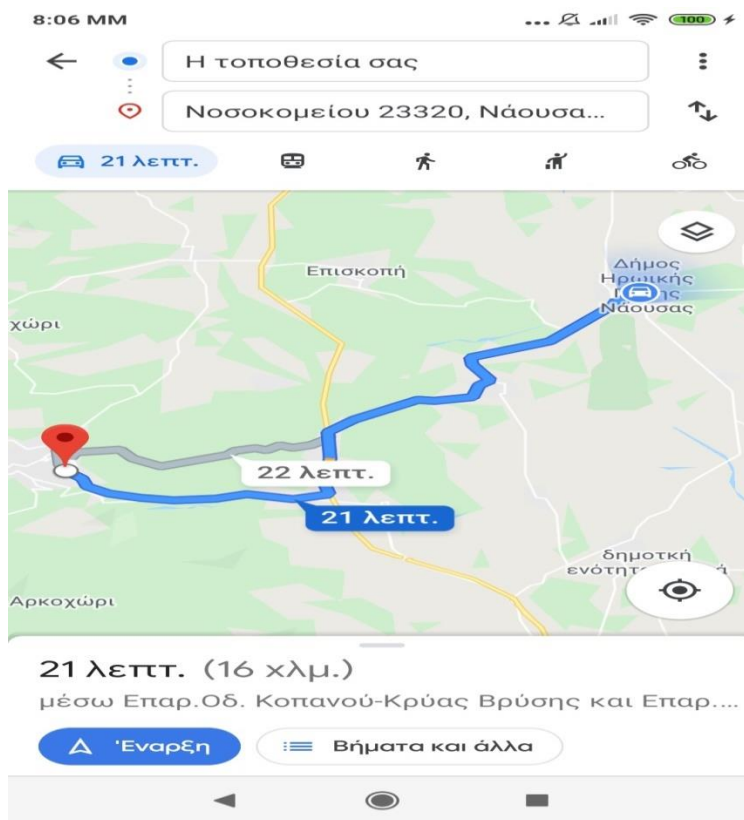
- Εικόνα νοσοκομείου.
- Διεύθυνση.
- Πόλη.
- Ταχυδρομικός κώδικας.
- Ιστοσελίδα του νοσοκομείου, η οποία αν πατήσει σε αυτήν ο χρήστης του ανοίγει την ιστοσελίδα τους.
- Τηλέφωνο, στο οποίο αν πατήσει πάνω του ο χρήστης τον οδηγεί στην κλήση του νοσοκομείου.
- Απόσταση, η οποία προσδιορίζεται αυτόματα βάση την τοποθεσία που βρίσκεται ο χρήστης και η λίστα με τα νοσοκομεία είναι ταξινομημένοι με την κοντινότερο νοσοκομείο.
- Οδηγίες για το πώς μπορεί να πάει στο νοσοκομείο με την χρήση του Google map.



Εικόνα 43-Εικόνα από τα νοσοκομεία



Εικόνα 44-Εικόνα από την χρησιμοποίηση του τηλεφώνου από το νοσοκομείο



Εικόνα 45-Εικόνα από την χρησιμοποίηση των οδηγιών



Εικόνα 46-Εικόνα απο την χρησιμοποίηση της ιστοσελίδας

5.3.3 Κλάση Hospital

Η κλάση Hospital έχει σκοπό να κρατάει τα δεδομένα που έχουν συλλεχθεί από την βάση δεδομένων. Επίσης κάθε δεδομένο που συλλέγει αρχικοποιείτε με ένα συγκεκριμένο όνομα και μια κενή τιμή, έτσι ώστε στην συνέχεια να αποθηκευτούν σε ένα πίνακα με τα στοιχεία.

```
1 package com.example.farmakeioo
2
3
4 data class Hospital(
5     val address: String = " ",
6     val image: String = " ",
7     var latitude: String = " ",
8     var longitude: String = " ",
9     var name: String = " ",
10    var phone: String = " ",
11    var website: String = " ")
12    {constructor() : this( address: " ", image: " ", latitude: " ", longitude: " ", name: " ", phone: " ", website: " ") }
```

Εικόνα 47-Εικόνα από την κλάση Hospital

5.3.4 Κλάση MyListHospital

Στην κλάση MyListHospital έχουν περάσει σαν παράμετροι οι πίνακες που έχουν τα στοιχεία των νοσοκομείων τα οποία τα διαχειρίζεται και τα προβάλλει στον χρήστη.

```

1 package com.example.farmakeio
2
3 import android.content.Intent
4 import android.net.Uri
5 import android.view.View
6 import android.view.ViewGroup
7 import android.widget.AdapterView
8 import android.widget.AdapterView.OnItemClickListener
9 import android.widget.AdapterView.OnItemClickListener
10 import com.squareup.picasso.Picasso
11
12
13 class MyListHospital(private val context: ActivityHospitals, private val name: MutableList<String>, private val address: MutableList<String>, private val imageId: MutableList<String>,
14 private val phone: MutableList<String>, private val website: MutableList<String>, private val latitude: MutableList<String>, private val longitude: MutableList<String>,
15 private val distance: MutableList<String>)
16 : ArrayAdapter<String>(context, R.layout.hospital_view, name) {
17
18 override fun getView(position: Int, view: View?, parent: ViewGroup): View {
19 val inflater :LayoutInflater = context.layoutInflater
20 val rowView :View = inflater.inflate(R.layout.hospital_view, root null, attachToRoot true)
21 val text_name :TextView = rowView.findViewById(R.id.text_name) as TextView
22 val image :ImageView = rowView.findViewById(R.id.image_view) as ImageView
23 val text_address :TextView = rowView.findViewById(R.id.text_address) as TextView
24 val text_website :TextView = rowView.findViewById(R.id.text_website) as TextView
25 val text_distance :TextView = rowView.findViewById(R.id.text_distance) as TextView
26 val text_phone :TextView = rowView.findViewById(R.id.text_phone) as TextView
27 val text_direction :TextView = rowView.findViewById(R.id.text_direction) as TextView
28 val text_ph :TextView = rowView.findViewById(R.id.text_ph) as TextView
29 var eikona :String =imageId[position]
30
31
32 text_address.text="Διεύθυνση: "+address[position]
33 text_website.text="Ιστοσελίδα: "+website[position]
34 text_phone.text=phone[position]
35 text_ph.text="Τηλέφωνο: "
36 text_distance.text="Απόσταση: "+distance[position]+" χιλ."
37 text_name.text=name[position]
38 text_phone.setOnClickListener { #View:
39 val intent = Intent(Intent.ACTION_DIAL, Uri.parse("tel:#{phone[position]}"))
40 context.startActivity(intent)
41 }
42 Picasso.with(context).load(eikona).resize( targetWidth 1000, targetHeight 455).centerCrop().into(image)
43 text_direction.setOnClickListener { #View:
44 val url :String =
45 "https://www.google.com/maps/dir/?api=1&destination=" + latitude[position] + "," + longitude[position] + "&travelmode=driving"
46 val intent = Intent(Intent.ACTION_VIEW, Uri.parse(url))
47 context.startActivity(intent)
48 }
49 return rowView
50 }
51 }

```

Εικόνα 48-Εικόνα από την κλάση MyListHospital

5.3.5 Κλάση ViewPharmacyActivity

Η κλάση ViewPharmacyActivity εμφανίζει τα φαρμακεία που υπάρχουν για τις πόλεις της Βέροιας, Νάουσας, Καστοριάς, Κοζάνης και Πτολεμαΐδας. Για να μπορέσει να γίνει αυτό χρησιμοποιούνται οι κλάσεις PharmacyList, Pharmacy και EndPoints. Που θα γίνει η ανάλυση αυτών των κλάσεων στην συνέχεια. Στις επόμενες εικόνες θα δούμε την βάση δεδομένων των κώδικα σε php για να ενώσει την βάση με την εφαρμογή , τον κώδικα μαζί με τις κλάσεις που απαιτούνται για την υλοποίηση του και στιγμιότυπα από την χρήση της εφαρμογής.

Τα στοιχεία που παρέχονται στο χρήστη για το κάθε φαρμακείο είναι τα εξής:

- Όνομα φαρμακείου.
- Διεύθυνση.
- Πόλη.
- Ταχυδρομικός κώδικας.
- Ώρες λειτουργίας στις οποίες υπάρχουν οι επιπλέον ώρες που το φαρμακείο παραμένει ανοιχτό.
- Τηλέφωνο, στο οποίο πατώντας το μπορεί να πάρει τηλέφωνο στο φαρμακείο.
- Απόσταση από την τοποθεσία του χρήστη.
- Κατάσταση του φαρμακείου, δηλαδή αν είναι ανοιχτό, κλειστό ή εφημερεύει.
- Οδηγίες για το πως μπορεί να πάει ο χρήστης στο φαρμακείο με την βοήθεια του Google maps.

ID	onoma	dieuthinsh	nomos	periochi	thlefono	L_k	platos	mhkos
1	Μηλίτσια Κων/να - Γρηγόρης	Μαμάτσιο 2	Κοζάνης	Κοζάνης	24610 49094	50100	40 307641	21 786867
2	Φλωρος Δημήτριος	Μεθώνης 1	Κοζάνης	Κοζάνης	24610 38003	50100	40 301489	21 793119
3	Καραστογιάννη Ι Αίλικα	Κωστή Παλαμά 12	Κοζάνης	Κοζάνης	24610 22752	50100	40 298081	21 789857
4	Χαραμή Μαρία - Θηρεσία	11ης Οκτωβρίου 49	Κοζάνης	Κοζάνης	24610 28797	50100	40 299208	21 797615
8	Μασούρα - Ζαμάνη Αλεξάνδρα Α	Βενζέλου 26	Ημοθίας	Βέροια	23310 24123	59132	40 525422	22 204198
9	Καπετανόπουλος Αθανάσιος	Ζαφειράκη 24	Ημοθίας	Νάουσα	23320 20203	59200	40 63131	22 068939
10	Αντράκος Χ. Γεώργιος	Καμπτή 14	Ημοθίας	Νάουσα	23320 52992	59200	40 629181	22 070087
11	Δίλιας Ιωάννης	28ης Οκτωβρίου 17	Ημοθίας	Νάουσα	23320 23030	59200	40 631655	22 067862
12	Τσιόγκα Ν Αναστασία	Πλατεια Καρατάσου 18	Ημοθίας	Νάουσα	23320 24220	59200	40 630805	22 069822
13	Γκέρας Μ Γεώργιος	Καμπτή 5	Ημοθίας	Νάουσα	23320 22800	59200	40 629963	22 069875
14	Μπάτση-Φειδοπούλου Βασιλική	17ης Οκτωβρίου	Ημοθίας	Νάουσα	23320 26507	59200	40 631615	22 068013
15	Μάντισος Ιωάννης	Κωνσταντινίδου 1	Ημοθίας	Νάουσα	23320 27000	59200	40 630273	22 069336
17	Ευάγγελου Μιχαηλίου	11ης Νοεμβρίου 4	Καστοριά	Καστοριά	2467 022529	52100	40 522284	21 262288
18	Κωνσταντίνος Μπαϊρακτάρης	Μεγ. Αλεξάνδρου 35	Καστοριά	Καστοριά	2467 029700	52100	40 520155	21 263171
19	Σαμαράς Γεώργιος	Ιουστινιανού 7	Καστοριά	Καστοριά	2467 028866	52100	40 522345	21 263858
20	Παυλοπούλου Αλεξάνδρα Κ	Λευφόρος Κόκκων 2	Καστοριά	Καστοριά	2467 081087	52100	40 519981	21 257477
21	Γιοβανόπουλου Ειρήνη /Λευκός	Βασ. Κωνσταντίνου 4	Καστοριά	Καστοριά	2467 022660	52100	40 519628	21 266722
23	Ιωάννης Παπαδόπουλος - Μαρία Παπαδοπούλου	Γράμμου 18	Καστοριά	Καστοριά	2467 083364	52100	40 522225	21 260823
24	Πυροβέτης Πέτρος & Σία	Μητροπόλεως 2	Καστοριά	Καστοριά	2467 026166	50100	40 521344	21 263712
25	Εφη Παπαδοπούλου	Παύλου Μελά 44	Κοζάνης	Κοζάνης	2461 031996	50100	40 304482	21 789152
26	Λαζάρου Στέφανου-Αρίσταρχου Στεφάνου	Πλατεια Νικης	Κοζάνης	Κοζάνης	2461 031783	50100	40 301215	21 788267
27	Δακίς Γεώργιος	Ερμού 1	Κοζάνης	Κοζάνης	2461 038639	50100	40 300277	21 789419
28	Σηφκα Ελένη Κ	Πλατεια Λαοσάνη 16	Κοζάνης	Κοζάνης	24610 40682	50100	40 298662	21 790103
29	Μουμουλίδου Μαρία	Μοναστηρίου 30B	Κοζάνης	Κοζάνης	24610 41961	50100	40 307382	21 787502
30	Παπαντώνης Ιωάννης	Χαρίση Παύλου 20	Κοζάνης	Κοζάνης	24610 22452	50100	40 300341	21 785828

Εικόνα 49-Εικόνα από την βάση δεδομένων.

```

1 <?php
2
3 class DbConnect
4 {
5
6     private $con;
7
8
9     function connect()
10 {
11
12     include_once dirname(__FILE__) . '/Constants.php';
13
14
15     $this->con = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_NAME);
16
17     if (mysqli_connect_errno()) {
18
19         echo "Failed to connect to MySQL: " . mysqli_connect_error();
20         return null;
21
22     }
23     return $this->con;
24 }
25 }

```

Εικόνα 50-Στιγμιότυπο από το αρχείο Connect

```

1 <?php
2 class DbOperation
3 {
4     private $con;
5
6     function __construct()
7     {
8         require_once dirname(__FILE__) . '/DbConnect.php';
9         $db = new DbConnect();
10        $this->con = $db->connect();
11    }
12
13    public function getPharmacy($genrel,$day){
14
15        mysqli_set_charset($this->con,"utf8");
16
17        if ($day=="1")
18        {
19            $stmt = $this->con->prepare("SELECT ID, onoma, perioxh,dieuthinsh, t_k, thilefono, platos, mkkos , kuriakh,hmeres_dihneureuseis, hmeres_dianuktereuseis FROM farmakeio,efhmereuon_hmeres,efhmeries_fa WHERE perioxh='Sgenrel' AND efhmereuon_hmeres.ID_farmakeiwn
20                =farmakeio.ID AND farmakeio.ID=efhmeries_fa.id_f ");
21        }
22        else if ($day=="2")
23        {
24            $stmt = $this->con->prepare("SELECT ID, onoma, perioxh,dieuthinsh, t_k, thilefono, platos, mkkos , deuthera,hmeres_dihneureuseis, hmeres_dianuktereuseis FROM farmakeio,efhmereuon_hmeres,efhmeries_fa WHERE perioxh='Sgenrel' AND efhmereuon_hmeres.ID_farmakeiwn
25                =farmakeio.ID AND farmakeio.ID=efhmeries_fa.id_f ");
26        }
27        else if ($day=="3")
28        {
29            $stmt = $this->con->prepare("SELECT ID, onoma, perioxh,dieuthinsh, t_k, thilefono, platos, mkkos , trith,hmeres_dihneureuseis, hmeres_dianuktereuseis FROM farmakeio,efhmereuon_hmeres,efhmeries_fa WHERE perioxh='Sgenrel' AND efhmereuon_hmeres.ID_farmakeiwn
30                =farmakeio.ID AND farmakeio.ID=efhmeries_fa.id_f ");
31        }
32        else if ($day=="4")
33        {
34            $stmt = $this->con->prepare("SELECT ID, onoma, perioxh,dieuthinsh, t_k, thilefono, platos, mkkos , tetarth,hmeres_dihneureuseis, hmeres_dianuktereuseis FROM farmakeio,efhmereuon_hmeres,efhmeries_fa WHERE perioxh='Sgenrel' AND efhmereuon_hmeres.ID_farmakeiwn
35                =farmakeio.ID AND farmakeio.ID=efhmeries_fa.id_f ");
36        }
37        else if ($day=="5")
38        {
39            $stmt = $this->con->prepare("SELECT ID, onoma, perioxh,dieuthinsh, t_k, thilefono, platos, mkkos , penyth,hmeres_dianuktereuseis,hmeres_dihneureuseis FROM farmakeio,efhmereuon_hmeres,efhmeries_fa WHERE perioxh='Sgenrel' AND efhmereuon_hmeres.ID_farmakeiwn
40                =farmakeio.ID AND farmakeio.ID=efhmeries_fa.id_f ");
41        }
42        else

```

Εικόνα 51-Στιγμιότυπο από το αρχείο DbOperation

```

42-  {
43-      $stmt = $this->con->prepare("SELECT ID, onoma, perioch,eleuthinsh, st_x, sthiefono, platos, wkhos , sabbato,hmeres_ilmneureusis, hmeres_dianaktereusis FROM 'farmakio' LEFT JOIN efhmeruson_hmeres ON farmakio.ID=efhmeruson_hmeres.ID_farmakioin LEFT JOIN
44-      efhmeries_fa ON farmakio.ID=efhmeries_fa.ID_f_ HMERIE farmakio.perioch='Sporoi'");
45-  }
46-  $stmt->execute();
47-  $stmt->bind_result($ID, $onoma, $perioch,$eleuthinsh,$st_x, $sthiefono, $platos,$wkhos,$sestraweres,$hmeres_ilmneureusis,$hmeres_dianaktereusis);
48-  $pharmacy = array();
49-  while($stmt->fetch()){
50-      $step = array();
51-      $step['ID'] = $ID;
52-      $step['onoma'] = $onoma;
53-      $step['perioch'] = $perioch;
54-      $step['eleuthinsh'] = $eleuthinsh;
55-      $step['st_x'] = $st_x;
56-      $step['sthiefono'] = $sthiefono;
57-      $step['platos'] = $platos;
58-      $step['wkhos'] = $wkhos;
59-      $step['sestraweres'] = $sestraweres;
60-      $step['hmeres_dianaktereusis'] = $hmeres_dianaktereusis;
61-      $step['hmeres_ilmneureusis'] = $hmeres_ilmneureusis;
62-      array_push($pharmacy, $step);
63-  }
64-  }
65-  return $pharmacy;
66-  }
67-  }
68-  }

```

Εικόνα 52-Στιγμιότυπο από το αρχείο DbOperation

```

1  <?php
2  require_once '../includes/DbOperation.php';
3
4  $response = array();
5
6  if(isset($_GET['op'])){
7
8      switch($_GET['op']){
9
10         case 'getpharmacy':
11
12             $genre1=$_POST['kal'];
13             $day=$_POST['day'];
14             $db = new DbOperation();
15             $pharmacy = $db->getPharmacy($genre1,$day);
16             if(count($pharmacy)<=0){
17                 $response['error'] = true;
18                 $response['message'] = 'Nothing found in the database';
19             }else{
20                 $response['pharmacy'] = $pharmacy;
21             }
22
23             break;
24         default:
25             $response['error'] = true;
26             $response['message'] = 'No operation to perform';
27
28     }
29 }
30 }else{
31     $response['error'] = false;
32     $response['message'] = 'Invalid Request';
33 }
34
35 print(json_encode($response,JSON_UNESCAPED_UNICODE));
36

```

Εικόνα 53-Στιγμιότυπο από το αρχείο index

Τα παραπάνω αρχεία είναι γραμμένα σε Php και η λειτουργία τους είναι για να συνδεθούν και να διαχειριστούν την βάση δεδομένων και να συλλέξουν τα σωστά δεδομένα για να μεταφερθούν στην εφαρμογή.


```

1 package com.example.farmakeiio
2 import android.graphics.Color
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import android.widget.*
7 import com.android.volley.Response
8 import com.android.volley.toolbox.StringRequest
9 import com.android.volley.toolbox.Volley
10 import org.json.JSONException
11 import org.json.JSONObject
12 import android.widget.Toast
13 import androidx.core.content.ContextCompat
14 import kotlinx.android.synthetic.main.activity_view_pharmacy.*
15 import java.util.*
16
17
18 class ViewPharmacyActivity : AppCompatActivity() {
19     private var listView: ListView? = null
20     private var pharmacyList: MutableList<Pharmacy?> = null
21     lateinit var option: Spinner
22     lateinit var ka : String
23
24
25     override fun onCreate(savedInstanceState: Bundle?) {
26         super.onCreate(savedInstanceState)
27         setContentView(R.layout.activity_view_pharmacy)
28         listView = findViewById(R.id.listViewPharmacy) as ListView
29         option = findViewById(R.id.sp_option) as Spinner
30         pharmacyList = mutableListOf()
31         option = findViewById(R.id.sp_option) as Spinner
32
33         swipe_container.setOnRefreshListener {
34             loadPharmacy()
35             swipe_container.isRefreshing = false
36             swipe_container.setProgressBackgroundColorSchemeColor(ContextCompat.getColor(context=this, R.color.colorPrimary))
37             swipe_container.setColorSchemeColors(Color.WHITE)
38         }
39     }
40
41     val options :Array<String> = arrayOf("Διάλεξε τοποθεσία ", "Βέροια", "Καστοριά", "Κοζάνη", "Νέουδα", "Πολιγαίσε")
42     option.adapter = ArrayAdapter<String>(context=this, android.R.layout.select_dialog_item, options)
43     option.onItemSelectedListener=object : AdapterView.OnItemSelectedListener{
44         override fun onNothingSelected(parent: AdapterView<*>?) {
45

```

Εικόνα 54-Εικόνα από την κλάση ViewPharmacyActivity

```

46     }
47
48     override fun onItemSelected(parent: AdapterView<*>?, view: View?, position: Int, id: Long) {
49         ka =options.get(position)
50
51         loadPharmacy()
52     }
53
54 }
55
56
57
58 private fun loadPharmacy() {
59     pharmacyList!!.clear()
60     val stringRequest =object :StringRequest(Method.POST, EndPoints.URL_GET_PHARMACY,Response.Listener<String> { s ->
61         try {
62             val obj = JSONObject(s)
63
64             val array :JSONArray = obj.getJSONArray( name: "pharmacy")
65
66             for (i :Int in 0..array.length()-1) {
67                 val objectArtist :JSONObject! = array.getJSONObject(i)
68                 val ph = Pharmacy(
69                     objectArtist.getString( name: "onoma"),
70                     objectArtist.getString( name: "periokh"),
71                     objectArtist.getString( name: "dieuchinsh"),
72                     objectArtist.getInt( name: "t_k"),
73                     objectArtist.getString( name: "thlefono"),
74                     objectArtist.getDouble( name: "platos"),
75                     objectArtist.getDouble( name: "mhkos"),
76                     objectArtist.getString( name: "extraweres"),
77                     objectArtist.getString( name: "hmeres_dianuktereuseis"),
78                     objectArtist.getString( name: "hmeres_dihmeureuseis")
79                 )
80             }
81             pharmacyList!!.add(ph)
82             val adapter = PharmacyList(context=this@ViewPharmacyActivity, pharmacyList!!)
83             listView!!.adapter = adapter
84         } catch (e: JSONException) {
85             e.printStackTrace()
86         }
87     }, Response.ErrorListener { volleyError -> Toast.makeText(applicationContext, volleyError.message, Toast.LENGTH_LONG).show() })
88
89 }
90

```

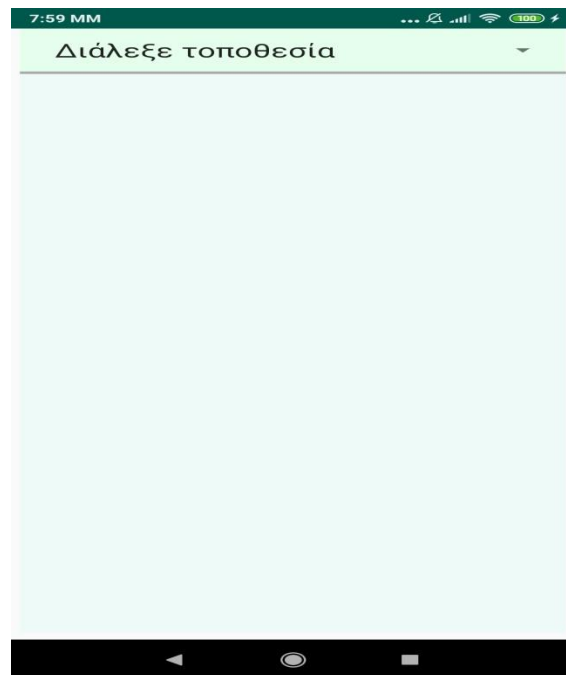
Εικόνα 55-Εικόνα από την κλάση ViewPharmacyActivity

Η συνάρτηση loadPharmacy έχει σκοπό να συνδεθεί με την βάση και να αποθηκεύσει τα στοιχεία των φαρμακείων στην pharmacyList. Επίσης η συνάρτηση getParams έχει σκοπό να στείλει την ημέρα και την επιλογή της πόλης, έτσι ώστε να ανακτηθούν τα σωστά δεδομένα από την βάση δεδομένων.

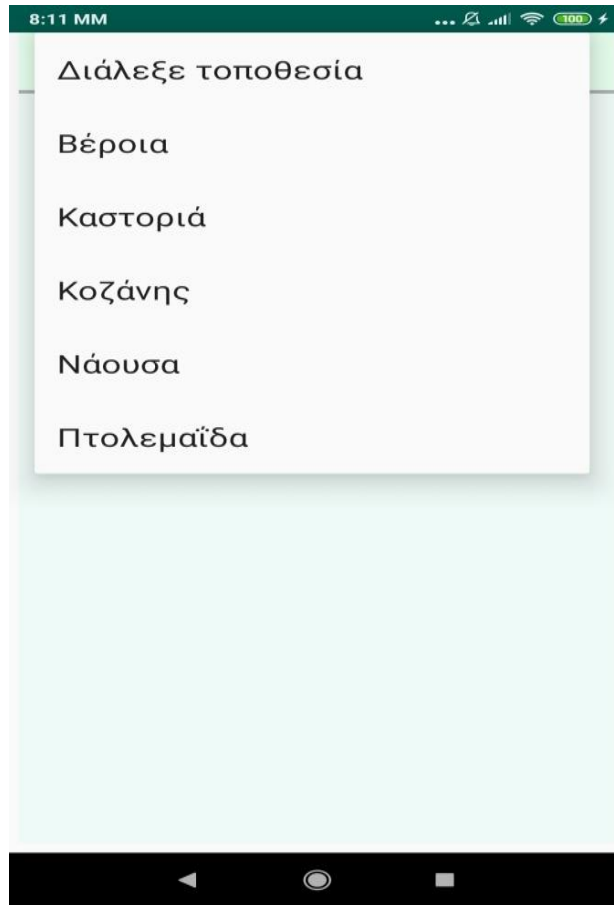
```
91 {
92     override fun getParams(): Map<String, String> {
93         val day :Int = Calendar.getInstance().get(Calendar.DAY_OF_WEEK)
94         val params = HashMap<String, String>()
95         params.put("kal", ka)
96         params.put("day", day.toString())
97         return params
98     }
99 }
100 val requestQueue :RequestQueue! = Volley.newRequestQueue( context: this)
101 requestQueue.add<String>(stringRequest)
102 }
103 }
104 }
105 }
```

Εικόνα 56-Εικόνα από την κλάση ViewPharmacyActivity

Στιγμιότυπα από το UI(Used interface design) που βλέπει ο χρήστης όταν επιλέξει να βρεί φαρμακείο φαίνεται στις παρακάτω εικόνες.

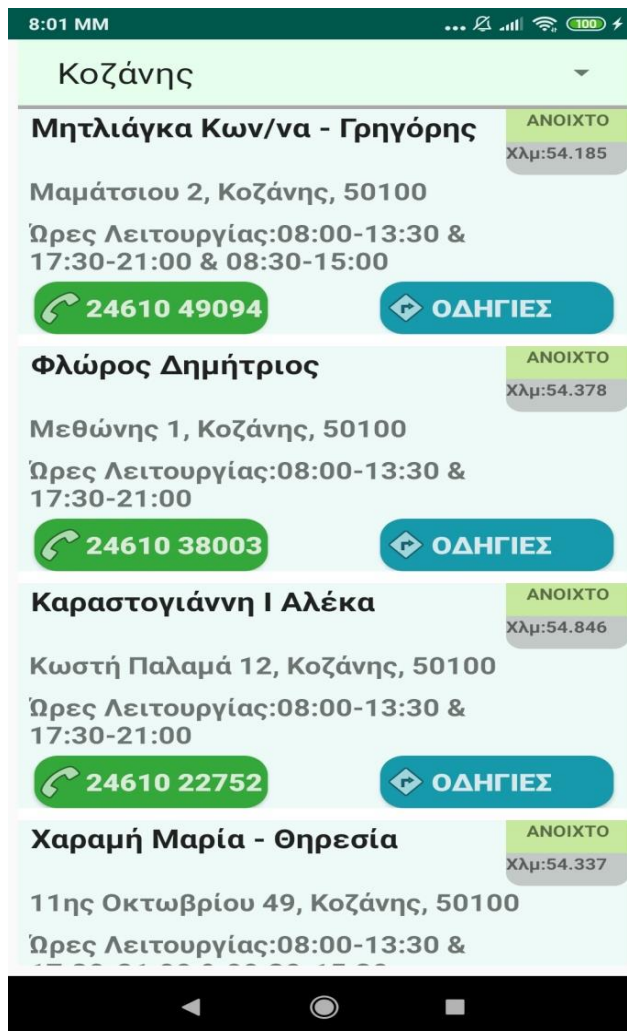


Εικόνα 57-Εικόνα από το UI των φαρμακείων.



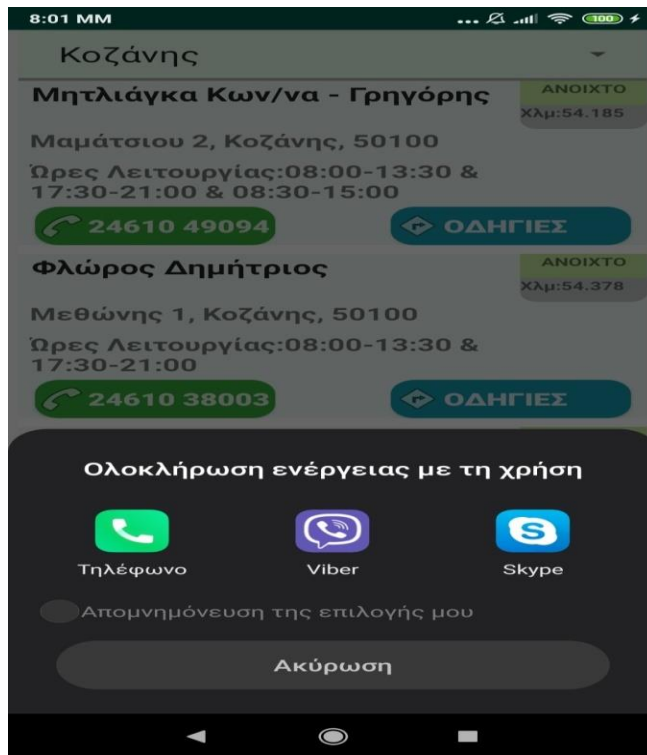
Εικόνα 58-Εικόνα απο το UI των φαρμακείων.

Ο χρήστης αρχικά θα πρέπει να διαλέξει πόλη για να εμφανιστούν τα φαρμακεία που υπάρχουν. Στην συνέχεια του εμφανίζονται όλα τα φαρμακεία όπως φαίνεται στην παρακάτω εικόνα. Ο χρήστης μπορεί να διαλέξει το πλησιέστερο φαρμακείο, ξέροντας αν είναι ανοιχτό και για πόση ώρα θα είναι. Επίσης μπορεί να πάρει τηλέφωνο ή να το αποθηκεύσει στο κινητό του για μελλοντική χρήση.

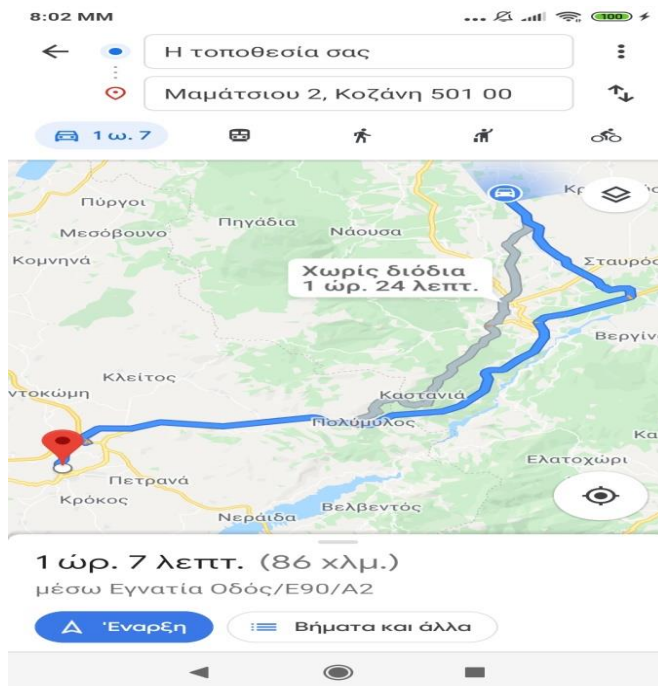


Εικόνα 59-Εικόνα από το UI των φαρμακείων.

Οι επόμενες δυο εικόνες δείχνουν τι γίνεται αν ο χρήστης επιθυμεί να πάρει τηλέφωνο το φαρμακείο και πατήσει πάνω στο κουμπί με τον αριθμό του φαρμακείου και όταν θελήσει να του δοθούν οδηγίες και πατήσει το κουμπί οδηγίες.



Εικόνα 60-Εικόνα από το UI των φαρμακείων.



Εικόνα 61-Εικόνα από το UI των φαρμακείων.

5.3.6 Κλάση EndPoints

Η κλάση EndPoints έχει το σύνδεσμο που βρίσκεται η βάση δεδομένων.

```
1 package com.example.farmakeiio
2
3 object EndPoints {
4     private val URL_ROOT = "https://victorious-attacker.000webhostapp.com/db_pharmacy/v1/?op=getpharmacy"
5     val URL_GET_PHARMACY :String = URL_ROOT
6 }
```

Εικόνα 62-Εικόνα από την κλάση EndPoints.

5.3.7 Κλάση Pharmacy

Η κλάση Pharmacy έχει σκοπό να ορίσει την ονομασία των δεδομένων, έτσι ώστε να μπορούν να χρησιμοποιηθούν κατάλληλα.

```
1 package com.example.farmakeiio
2
3
4
5 class Pharmacy(val onoma: String, val perioxh: String, val dieuthinh: String, val t_k: Int,
6               val thlefono: String, val platos: Double, val mhkos: Double, val extraweres: String,
7               val hmeres dianuktereuseis:String, val hmeres dihmeureuseis:String)
```

Εικόνα 63-Εικόνα από την κλάση Pharmacy.

5.3.8 Κλάση PharmacyList

Η κλάση PharmacyList έχει σκοπό να χρησιμοποιήσει τα δεδομένα που της έχουν περάσει για να μπορέσει να τα προβάλει και να είναι εύκολα στην κατανόηση από τον χρήστη. Οι επόμενες εικόνες περιέχουν τον κώδικα αυτής της κλάσης

```

1 package com.example.farmakeioo
2
3 import android.app.Activity
4 import android.content.Intent
5 import android.graphics.Color
6 import android.net.Uri
7 import android.view.View
8 import android.view.ViewGroup
9 import android.widget.ArrayAdapter
10 import android.widget.TextView
11 import android.widget.Button
12 import java.util.*
13 import kotlin.math.round
14
15

```

Εικόνα 64-Εικόνα από την κλάση PharmacyList.

```

16 class PharmacyList(private val context: Activity, internal var pharmacy: List<Pharmacy>) : ArrayAdapter<Pharmacy>(context, R.layout.layout_list_pharmacy, pharmacy) {
17
18     var mhkos:Double = 0.0
19     var platos:Double = 0.0
20
21     companion object
22     {
23         var pla = 0.0
24         var mka=0.0
25     }
26
27     override fun getView(position: Int, convertView: View?, parent: ViewGroup): View {
28         val inflater :LayoutInflater = context.layoutInflater
29         val listViewItem :View = inflater.inflate(R.layout.layout_list_pharmacy, root null, attachToRoot true)
30         val textViewName :TextView = listViewItem.findViewById(R.id.textViewName) as TextView
31         val textViewNomos :TextView = listViewItem.findViewById(R.id.textViewNomos) as TextView
32         val textViewWtrario :TextView = listViewItem.findViewById(R.id.textViewWtrario) as TextView
33         val textViewThlefono :Button =listViewItem.findViewById(R.id.textViewThlefono) as Button
34         val directions :Button =listViewItem.findViewById(R.id.directions) as Button
35         val leitourgial :TextView =listViewItem.findViewById(R.id.leitourgia) as TextView
36         val apostash :TextView =listViewItem.findViewById(R.id.apostash) as TextView
37         var wrario=""
38         var leitourgia=""
39         var meros1=""
40         var meros2=""
41         var meros1a=""
42         var meros1b=""
43         var meros2a=""
44         var meros2b=""
45         var day :Int = Calendar.getInstance().get(Calendar.DAY_OF_WEEK)
46         var hour :Int = Calendar.getInstance().get(Calendar.HOUR_OF_DAY)
47         var minutes :Int = Calendar.getInstance().get(Calendar.MINUTE)
48         var seconds :Int = Calendar.getInstance().get(Calendar.SECOND)
49         var dayMonth :Int = Calendar.getInstance().get(Calendar.DAY_OF_MONTH)
50

```

Εικόνα 65-Εικόνα από την κλάση PharmacyList.

```

51     val kpharmacy : Pharmacy = pharmacy[position]
52     meros1=kpharmacy.extraweres.take( n: 4)
53     meros2=kpharmacy.extraweres.takeLast( n: 4)
54     if (meros1=="null")
55     {
56         meros1="0000"
57         meros2="0000"
58     }
59     meros1a=meros1.take( n: 2)
60     meros1b=meros1.takeLast( n: 2)
61     meros2a=meros2.take( n: 2)
62     meros2b=meros2.takeLast( n: 2)
63     mhkos = kpharmacy.mhkos
64     platos=kpharmacy.platos
65
66
67     if ( (day==3) || (day==5) || (day==6))
68     {
69         wvario="08:00-13:30 & 17:30-21:00"
70
71         if (((hour>=8) && (hour<13) || (hour==13) && (minutes<30)) || ((hour>17) && (hour<21)) || ((hour==17) && (minutes>30) && (seconds>1)) || ((hour==20)
72             && (minutes<59) && (seconds<59)) || ((hour>=meros1a.toInt()) && (hour<=meros2a.toInt()) || (hour==meros1a.toInt() && minutes==meros1b.toInt()
73             || (hour==meros2a.toInt() && minutes<=meros2b.toInt())) )
74         {
75             leitourgia="ΑΝΟΙΧΤΟ"
76         }
77         else
78         {
79             leitourgia = "ΚΛΕΙΣΤΟ"
80         }
81     }
82     else if ((day==2) || (day==4))
83     {
84         wvario="08:00-14:30"
85         if ((hour>=8) && (hour<14) || (hour==14) && (minutes<30) && ((hour>meros1a.toInt()) && (hour<=meros2a.toInt()) || (hour==meros1a.toInt())
86             && (minutes>=meros1b.toInt()) || (hour==meros2a.toInt()) && (minutes<=meros2b.toInt())) )
87         {
88             leitourgia="ΑΝΟΙΧΤΟ"
89         }
90         else
91         {
92             leitourgia = "ΚΛΕΙΣΤΟ"
93         }
94     }

```

Εικόνα 66-Εικόνα από την κλάση PharmacyList.

```

94     }
95     if (day==7)
96     {
97         if (((hour>=meros1a.toInt()) && (hour<=meros2a.toInt()) || (hour==meros1a.toInt() && minutes>=meros1b.toInt() || (hour==meros2a.toInt()
98             && minutes<=meros2b.toInt()))))
99         {
100             leitourgia="ΑΝΟΙΧΤΟ"
101         }
102         else
103         {
104             leitourgia = "ΚΛΕΙΣΤΟ"
105         }
106     }
107     if (day == 1)
108     {
109         leitourgia = "ΚΛΕΙΣΤΟ"
110     }
111     if (leitourgia=="ΚΛΕΙΣΤΟ")
112     {
113         leitourgial.setBackgroundColor(Color.rgb( red: 236, green: 160, blue: 160))
114     }
115     else
116     {
117         leitourgial.setBackgroundColor(Color.rgb( red: 198, green: 233, blue: 158))
118     }
119     textViewWvario.text="Ώρα Λειτουργίας: "+wvario + " & " +meros1a+" "+meros1b+" "+meros2a + " "+meros2b
120     if (meros1.toInt()==0 )
121     {
122         textViewWvario.text="Ώρα Λειτουργίας: "+wvario
123     }
124     leitourgial.text=leitourgia
125     var la :Double = distance(kpharmacy.platos, kpharmacy.mhkos, pla, mka)
126     apostash.text="Απόσταση: "+ la.toString()
127
128     var ef_meros1 :String = kpharmacy.hmeres_dianuktereuseis.take( n: 4)
129     var ef_meros2 :String = kpharmacy.hmeres_dianuktereuseis.takeLast( n: 4)
130     if (ef_meros1=="null")
131     {
132         ef_meros1="0000"
133         ef_meros2="0000"
134     }

```

Εικόνα 67-Εικόνα από την κλάση PharmacyList.


```

135     var efi_1 :String = ef_meros1.take( n: 2)
136     var efi_2 :String = ef_meros1.takeLast( n: 2)
137     var efi_3 :String = ef_meros2.take( n: 2)
138     var efi_4 :String = ef_meros2.takeLast( n: 2)
139
140     if (efi_1.toInt() == dayMonth||efi_2.toInt() == dayMonth||efi_3.toInt() == dayMonth||efi_4.toInt() == dayMonth) {
141
142         textViewWrrario.text = "Ώρες Λειτουργίας: 08:00-14:30 & 17:30-08:00 ΕΠΟΜΕΝΗΣ"
143         leitourgial.text = "ΕΦΗΜΕΡΕΥΕΙ"
144         leitourgial.setBackgroundColor(Color.rgb( red: 198, green: 233, blue: 158))
145         if ((kpharmacy.perioixh == "Κοζάνης" || kpharmacy.perioixh == "Πτολεμαΐδα") && day == 7) {
146             textViewWrrario.text = "Ώρες Λειτουργίας: 17:30-08:00 ΕΠΟΜΕΝΗΣ"
147         }
148         if (kpharmacy.perioixh == "Βέροια") {
149             textViewWrrario.text = "Ώρες Λειτουργίας: 08:30-14:30 & 19:00-08:00 ΕΠΟΜΕΝΗΣ"
150         }
151     }
152
153     ef_meros1= kpharmacy.hmeres_dihmeureuseis.take( n: 4)
154     ef_meros2= kpharmacy.hmeres_dihmeureuseis.takeLast( n: 4)
155     if (ef_meros1=="null")
156     {
157         ef_meros1="0000"
158         ef_meros2="0000"
159     }
160     efi_1= ef_meros1.take( n: 2)
161     efi_2= ef_meros1.takeLast( n: 2)
162     efi_3= ef_meros2.take( n: 2)
163     efi_4= ef_meros2.takeLast( n: 2)
164
165     if (efi_1.toInt() == dayMonth||efi_2.toInt() == dayMonth||efi_3.toInt() == dayMonth||efi_4.toInt() == dayMonth) {
166         textViewWrrario.text = "Ώρες Λειτουργίας: 08:00-21:00"
167         leitourgial.text = "ΔΙΗΜΕΡΕΥΕΙ"
168         leitourgial.setBackgroundColor(Color.rgb( red: 198, green: 233, blue: 158))
169         if (kpharmacy.perioixh == "Βέροια" && (day == 3 || day == 5 || day == 6)) {
170             textViewWrrario.text = "Ώρες Λειτουργίας: 08:00-17:30"
171         }
172         if (kpharmacy.perioixh == "Βέροια" && (day == 7 || day == 1)) {
173             textViewWrrario.text = "Ώρες Λειτουργίας: 08:00-14:30"
174         }
175     }
176 }
177

```

Εικόνα 68-Εικόνα από την κλάση PharmacyList.

```

178     textViewName.text = kpharmacy.onoma
179     textViewNomos.text = kpharmacy.dieuthinsh+ ", "+kpharmacy.perioixh +", "+kpharmacy.t_k
180     textViewThlefono.text=kpharmacy.thlefono
181
182     // κουμπί για να μπορείς να παρεις thlefono
183     textViewThlefono.setOnClickListener{ it.View!
184
185         val intent = Intent(Intent.ACTION_DIAL, Uri.parse( UriString: "tel:$ {kpharmacy.thlefono}"))
186         context.startActivity(intent)
187     }
188
189     directions.setOnClickListener{ it.View!
190         val url :String =
191             "https://www.google.com/maps/dir/?api=1&destination=" + kpharmacy.platos + ", " + kpharmacy.mhkos + "&travelmode=driving"
192         val intent = Intent(Intent.ACTION_VIEW, Uri.parse(url))
193         context.startActivity(intent)
194     }
195     return listViewItem
196 }
197
198 private fun distance(lat1: Double, lon1: Double, lat2: Double, lon2: Double): Double {
199     val theta :Double = lon1 - lon2
200     var dist :Double = Math.sin(Math.toRadians(lat1)) * Math.sin(Math.toRadians(lat2)) + Math.cos(
201         Math.toRadians(lat1)
202     ) * Math.cos(Math.toRadians(lat2)) * Math.cos(Math.toRadians(theta))
203     dist = Math.acos(dist)
204     dist = Math.toDegrees(dist)
205     dist = dist * 60 * 1.1515
206     dist = dist * 1.609344
207     dist= round( * dist*1000) /1000
208     return (dist)
209 }
210
211

```

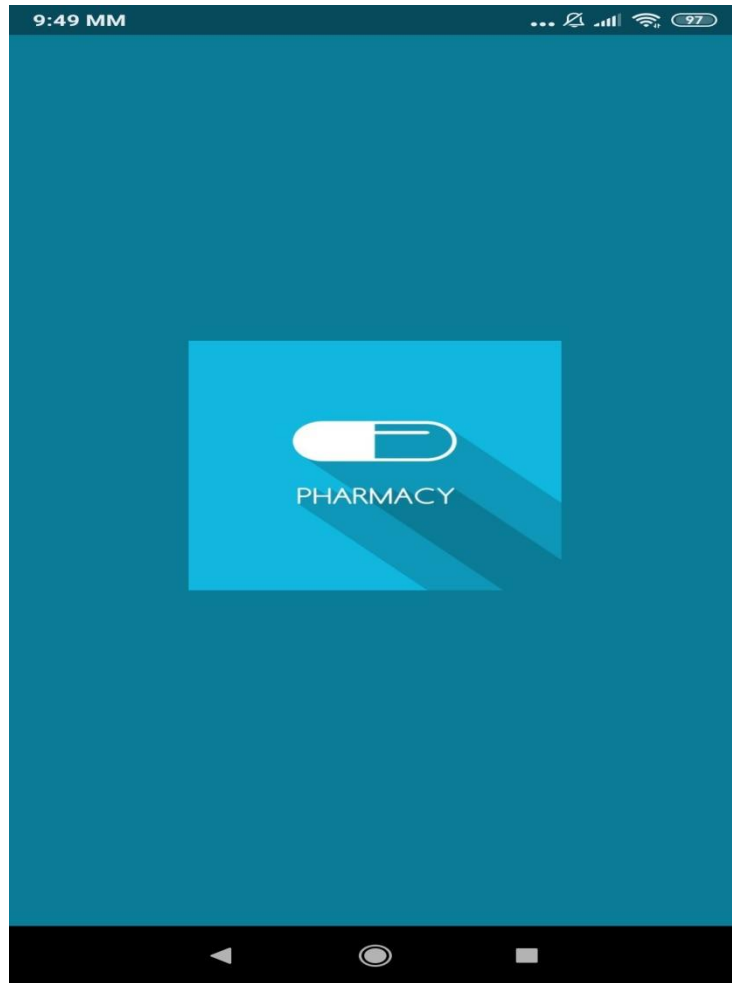
Εικόνα 69-Εικόνα από την κλάση PharmacyList.

5.3.9 Κλάση SplashScreen

Η κλάση αυτή είναι για να προβάλει το λογότυπο της εφαρμογής και για να μην γίνεται αντιληπτό η καθυστέρηση που υπάρχει όταν ο χρήστης ανοίγει μια εφαρμογή. Το λογότυπο παραμένει για 1.6 δευτερόλεπτα στην οθόνη του κινητού. Ο κώδικας και UI φαίνεται στις παρακάτω εικόνες.

```
1 package com.example.farmakeioo
2
3
4 import android.content.Intent
5 import android.os.Bundle
6 import android.os.Handler
7 import androidx.appcompat.app.AppCompatActivity
8 import kotlinx.android.synthetic.main.splash_screen.*
9
10
11 class SplashScreen : AppCompatActivity() {
12
13     override fun onCreate(savedInstanceState: Bundle?) {
14         super.onCreate(savedInstanceState)
15         setContentView(R.layout.splash_screen)
16
17         splash_logo.setImageResource(R.drawable.pharm)
18
19         val SPLASH_TIME_OUT = 1600
20         val homeIntent = Intent(packageContext, MainActivity::class.java)
21         Handler().postDelayed({
22             //Do some stuff here, like implement deep linking
23             startActivity(homeIntent)
24             finish()
25         }, SPLASH_TIME_OUT.toLong())
26     }
27
28 }
```

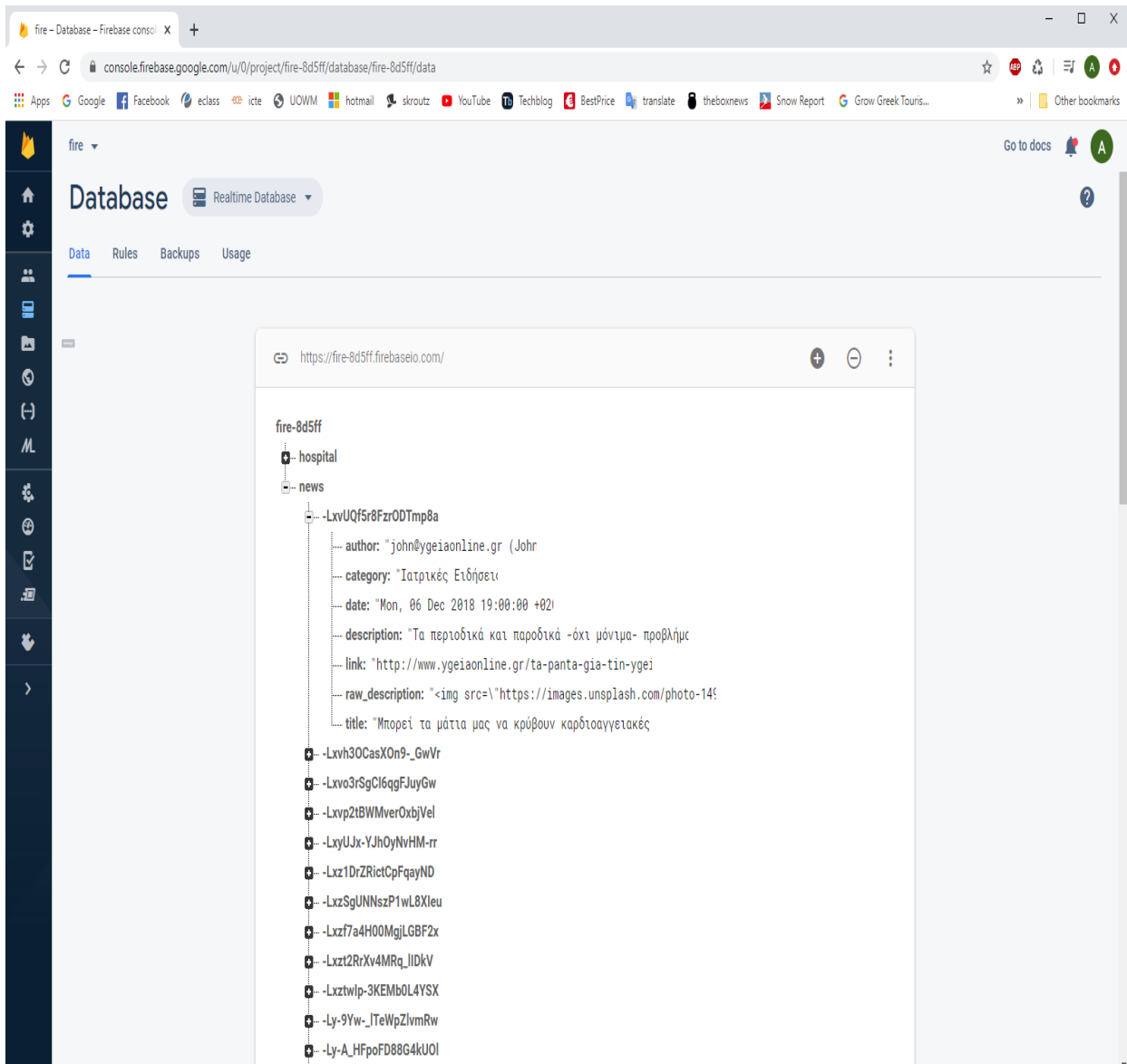
Εικόνα 70-Εικόνα από την κλάση SplashScreen.



Εικόνα71-Εικόνα με το λογότυπο της εφαρμογής.

5.3.10 Κλάση ActivityArticles

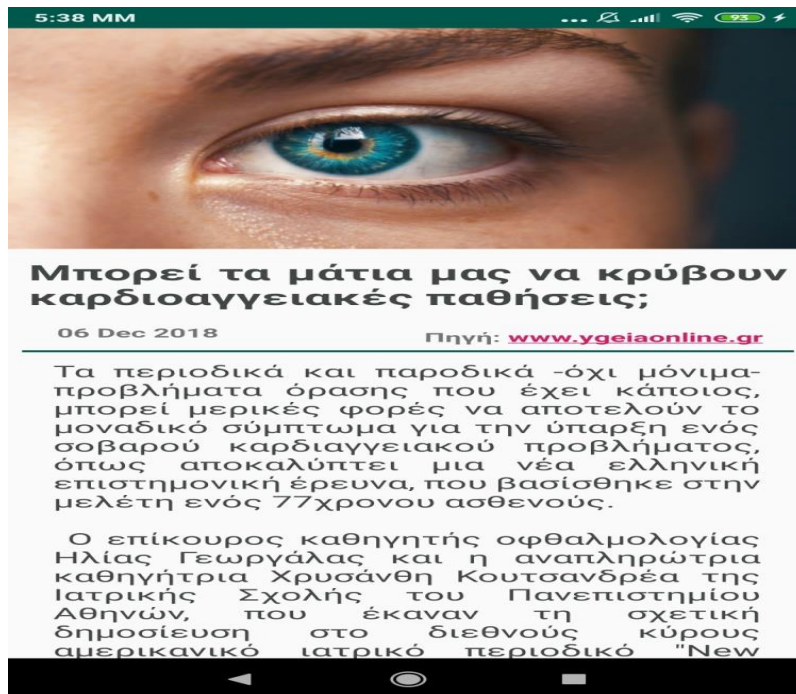
Η κλάση ActivityArticles εμφανίζει τα άρθρα που έχουν αυτόματα αποθηκευτεί στην βάση όταν ανεβαίνουν στην ιστοσελίδα. Έχουν συλλεχθεί από τρεις ιστοσελίδες την www.ygeiaonline.gr , www.onmed.gr και απο την www.iatropedia.gr. Τα θέματα αυτών των άρθρων έχουν σχέση για την υγεία, την διατροφή και την άθληση. Οι παρακάτω εικόνες απεικονίζουν την βάση δεδομένων, το στιγμιότυπα από το UI που βλέπει ο χρήστης και τον κώδικα των κλάσεων για να προβληθούν τα άρθρα.



Εικόνα 72-Εικόνα από της βάσης δεδομένων.



Εικόνα 73-Εικόνα απο το UI των άρθρων.



Εικόνα 74-Εικόνα από το UI των άρθρων.

```

1 package com.example.farmakeioo
2
3 import android.content.Intent
4 import android.os.Bundle
5 import androidx.appcompat.app.AppCompatActivity
6 import com.google.firebase.database.*
7 import kotlin.android.synthetic.main.activity_articles.*
8
9
10 class ActivityArticles : AppCompatActivity() {
11     val title : MutableList<String> = mutableListOf<String>()
12     val description : MutableList<String> = mutableListOf<String>()
13     val imageId : MutableList<String> = mutableListOf<String>()
14     val products : ArrayList<News> = arrayListOf<News>()
15     val date : ArrayList<String> = arrayListOf<String>()
16     val link : ArrayList<String> = arrayListOf<String>()
17     private lateinit var database: DatabaseReference
18
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContentView(R.layout.activity_articles)
22         database = FirebaseDatabase.getInstance().reference
23
24         loadDatabase()
25     }
26
27     fun loadDatabase() {
28
29         val ref : DatabaseReference = FirebaseDatabase.getInstance().getReference( path: "news")
30         ref.addValueEventListener(object : ValueEventListener {
31             override fun onDataChange(dataSnapshot: DataSnapshot) {
32                 products.clear()
33                 description.clear()
34                 title.clear()
35                 imageId.clear()
36                 for (productSnapshot : DataSnapshot! in dataSnapshot.children) {
37                     val product : News? = productSnapshot.getValue(News::class.java)
38                     products.add(product!!)
39                 }
40                 var i=0
41
42                 while (i<products.size)
43                 {
44
45                     title.add( products[i].title)
46                     description.add(products[i].description)
47                     date.add(products[i].date.substring(4,17))
48
49
50

```

Εικόνα 75-Εικόνα από την κλάση ActivityArticles .

Η συνάρτηση loadDatabase είναι για να φορτώσει τα δεδομένα από την βάση δεδομένων καλώντας την κλάση News για να αρχικοποιήσει τις μεταβλητές.

```

51         if (products[i].raw_description.contains( other: "img src").not())
52         {
53             imageUrl.add("https://www.gsmarket.ro/7844-large_default/difuzor-htc-desire-510-original.jpg")
54         }
55         else
56         {
57             imageUrl.add(products[i].raw_description.substringAfter( delimiter: "img src=").substringBefore( delimiter: "\ /"))
58         }
59         link.add(products[i].link.substringAfter( delimiter: "://").substringBefore( delimiter: "/"))
60
61         i++
62     }
63
64     }
65
66     val myListAdapter = MyListAdapter( context: this@ActivityArticles, title, description, imageUrl, date)
67     listView.adapter = myListAdapter
68
69     listView.setOnItemClickListener({adapterView, view, position, id ->
70
71         val intent = Intent( packageContext: this@ActivityArticles, NewsPost::class.java)
72         intent.putExtra( name: "description", description[position])
73         intent.putExtra( name: "title", title[position])
74         intent.putExtra( name: "image", imageUrl[position])
75         intent.putExtra( name: "link", link[position])
76         intent.putExtra( name: "date", date[position])
77
78         startActivity(intent)
79     })
80
81     }
82
83     override fun onCancelled(databaseError: DatabaseError) {
84         throw databaseError.toException()
85     }
86
87     })
88
89 }

```

Εικόνα 76-Εικόνα από την κλάση ActivityArticles .

5.3.11 Κλάση News

```

1 package com.example.farmakeioo
2
3 data class News(
4     val author: String = " ",
5     val category: String = " ",
6     var date: String = " ",
7     var description: String = " ",
8     var link: String = " ",
9     var raw_description: String = " ",
10    var title: String = " ")
11    {constructor() : this( author: " ", category: " ", date: " ", description: " ", link: " ", raw_description: " ", title: " ")}

```

Εικόνα 77-Εικόνα από την κλάση News.

5.3.12 Κλάση MyListAdapter

Η κλάση αυτή είναι για να προβάλει τα δεδομένα των άρθρων στον χρήστη για να μπορεί να διαλέξει ποιο θα διαβάσει.

```

1 package com.example.farmakeiioo
2
3 import android.view.View
4 import android.view.ViewGroup
5 import android.widget.AdapterView
6 import android.widget.ImageView
7 import android.widget.TextView
8 import com.squareup.picasso.Picasso
9
10
11 class MyListAdapter(private val context: ActivityArticles, private val title: MutableList<String>, private val description: MutableList<String>,
12 private val imgid: MutableList<String>, private val datee: MutableList<String>)
13 : AdapterView<String>(context, R.layout.news_view, title) {
14
15 override fun getView(position: Int, view: View?, parent: ViewGroup): View {
16 val inflater :LayoutInflater = context.layoutInflater
17 val rowView :View! = inflater.inflate(R.layout.news_view, root: null, attachToRoot: true)
18
19 val titleText :TextView = rowView.findViewById(R.id.title) as TextView
20 val imageView :ImageView = rowView.findViewById(R.id.icon) as ImageView
21 val date :TextView = rowView.findViewById(R.id.date) as TextView
22 val eikona :String =imgid[position]
23 titleText.text = title[position]
24 Picasso.with(context).load(eikona).resize( targetWidth: 500, targetHeight: 455).into(imageView)
25
26 date.text=datee[position]
27
28 return rowView
29 }
30

```

Εικόνα 78-Εικόνα από την κλάση MyListAdapter.

5.3.12 Κλάση NewsPost

Όταν ο χρήστης πατήσει πάνω σε ένα άρθρο η κλάση που καλείται για να προβάλει το άρθρο είναι η NewsPost. Η οποία ο κώδικας της φαίνεται στην επόμενη εικόνα.

```

1 package com.example.farmakeiioo
2
3 import android.os.Bundle
4 import androidx.appcompat.app.AppCompatActivity
5 import com.squareup.picasso.Picasso
6 import kotlinx.android.synthetic.main.news_post.*
7
8 class NewsPost : AppCompatActivity() {
9
10 override fun onCreate(savedInstanceState: Bundle?) {
11 super.onCreate(savedInstanceState)
12 setContentView(R.layout.news_post)
13
14 val title :String? = intent.getStringExtra( name: "title")
15 val description :String? = intent.getStringExtra( name: "description")
16 val image :String? = intent.getStringExtra( name: "image")
17 val datee :String? =intent.getStringExtra( name: "date")
18 val link :String? =intent.getStringExtra( name: "link")
19
20 titlee.text=title
21 Picasso.with( context: this).load(image).resize( targetWidth: 500, targetHeight: 1000).centerInside().into(imagee)
22 date.text=datee
23 source.text="Πηγή: "+link
24 descriptionn.text=description
25
26 }
27

```

Εικόνα 79-Εικόνα από την κλάση NewsPost.

6. Επίλογος

Στο κεφάλαιο αυτό, παρουσιάζουμε τα συμπεράσματα της έρευνας που πραγματοποιήθηκε στα πλαίσια της διπλωματικής εργασίας. Τέλος γίνεται λόγος για τυχόν μελλοντικές επεκτάσεις, ώστε η εφαρμογή να γίνει αποδοτικότερη ως προς τον χρήστη.

6.1 Συμπεράσματα

Στόχος της παρούσας διπλωματικής ήταν, η δημιουργία ενός συστήματος για την εύρεση πλησιέστερου νοσοκομείου και φαρμακείου. Ο στόχος επιτεύχθηκε, καθώς δημιουργήσαμε μία έξυπνη εφαρμογή, η οποία των προτιμήσεων του χρήστη, προτείνει φαρμακεία και νοσοκομεία υπολογίζοντας την απόσταση και την κοντινότερη διαδρομή. Επίσης ο χρήστης μπορεί να βρει πληροφορίες για το φάρμακο που διαθέτει εύκολα και γρήγορα με την χρήση του barcode scanner. Τέλος μπορεί να ενημερωθεί με άρθρα σχετικά για την υγεία, την διατροφή και θέματα που τον απασχολεί.

6.2 Μελλοντικές επεκτάσεις

Η εφαρμογή Pharmacy που δημιουργήθηκε στα πλαίσια της εκπόνησης της διπλωματικής εργασίας, σχεδιάστηκε με σκοπό να βοηθήσει τον χρήστη και να κάνει την ζωή του πιο εύκολη. Παρά το γεγονός ότι η συγκεκριμένη εφαρμογή πληροί όλες τις προδιαγραφές του συστήματος, σίγουρα μπορούν να γίνουν αρκετές επεκτάσεις ώστε να βελτιωθεί ακόμη περισσότερο η εφαρμογή.

Αρχικά θα μπορούσε να προσθεθούν και άλλες πόλεις για να υπάρχει μεγαλύτερο εύρος φαρμακείων. Έτσι ο χρήστης δεν θα περιορίζεται μόνο στις πόλεις που υπάρχουν στο σύστημα.

Επίσης θα μπορούσε να αποθηκεύει τα φάρμακα του στην εφαρμογή για να γνωρίζει πάντα τι φάρμακα έχει καταναλώσει. Που θα βοηθούσε τον γιατρό να αποκτήσει μια εικόνα για τις αγωγές που έχει πάρει ο χρήστης.

7. Βιβλιογραφικές αναφορές

- [1] <https://ppissis.com.cy/el/blog/kinito-telefono-i-exelixa-tis-syskevis>
- [2] <https://el.wikipedia.org/wiki/Android>
- [3] <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [4] https://en.wikipedia.org/wiki/Android_version_history
- [5] https://www.android.com/intl/en_uk/history/
- [6] <https://developer.android.com/distribute/marketing-tools/brand-guidelines>
- [7] https://www.android.com/intl/en_uk/versions/jelly-bean-4-3/
- [8] https://www.android.com/intl/en_uk/versions/kit-kat-4-4/
- [9] https://www.android.com/intl/en_uk/versions/lollipop-5-0/
- [10] https://www.android.com/intl/en_uk/versions/marshmallow-6-0/
- [11] https://www.android.com/intl/en_uk/versions/nougat-7-0/
- [12] <https://www.android.com/versions/oreo-8-0/>
- [13] <https://www.android.com/versions/pie-9-0/>
- [14] <https://kotlinlang.org/docs/reference/android-overview.html>
- [15] https://www.w3schools.com/php/php_intro.asp
- [16] <https://docs.oracle.com/cloud/latest/mysql-cloud/index.html>
- [17] https://developer.android.com/studio/?gclid=Cj0KCQiAmsrxBRDaARIsANyiD1oeS0I5I0wgVUqn16_Zdh2PqT_Z6VI9F_U5nWvvXB4I9ruGGcGenk4aAvNPEALw_wcB
- [18] <https://developer.android.com/studio/intro>
- [19] <https://el.wikipedia.org/wiki/XAMPP>

[20] <https://firebase.google.com/docs/database>

[21] <https://www.photoshopnews.com/feature-stories/photoshop-profile-thomas-john-knoll-10/>

[22] <https://zapier.com/learn/getting-started-guide/what-is-zapier/>

[23] <https://www.androidauthority.com/history-android-os-name-789433/>