



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξης Λογισμικού και Τεχνητής
Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Android εφαρμογής Happy Hour Offers Development of android application Happy Hour Offers
Όνοματεπώνυμο Φοιτητή	Μαρία-Αρετή Ζωγράφου
Πατρώνυμο	Αθανάσιος
Αριθμός Μητρώου	ΜΠΣΠ 18015
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης ΙΟΥΛΙΟΣ 2021

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

(υπογραφή)

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον άνθρωπο που αποτέλεσε το έναυσμα για το ξεκίνημα των μεταπτυχιακών μου σπουδών, καθώς και για την έμπρακτη στήριξη του καθ' όλη τη διάρκεια αυτών.

Επίσης θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή κ. Ευθύμιο Αλέπη για τις γνώσεις που μας παρείχε στη διάρκεια των μεταπτυχιακών σπουδών.

Περιεχόμενα

Περίληψη.....	8
Abstract.....	8
Εισαγωγή.....	8
Διαδεδομένα λειτουργικά συστήματα κινητών συσκευών (Android – IOS).....	9
Android	9
Αρχιτεκτονική του Android	11
IOS	15
Αρχιτεκτονική του IOS	16
Τεχνολογίες, Εργαλεία, Πρωτόκολλα ανάπτυξης εφαρμογής.....	17
Εργαλεία Ανάπτυξης της Εφαρμογής	17
Java.....	17
Software Development Kit(SDK).....	17
Java Development Kit(JDK).....	18
Android Studio.....	18
XML	19
Firebase.....	19
Παρουσίαση και χρήση της εφαρμογής	20
Απαιτήσεις	21
Σύνδεση και Δημιουργία νέου λογαριασμού.	21
Εγγραφή(register) για user type = business (επιχειρηματίας)	22
Προσθήκη προσφοράς.....	24
Ενημέρωση εγγεγραμμένων χρηστών κατά την προσθήκη νέας προσφοράς.....	26
Λίστα προσφορών.....	27
Εγγραφή(register) για user type = customer (πελάτης)	29
Android Permissions	29
Εύρεση προσφορών κοντά στο χρήστη	30
Αρχική σελίδα για χρήστη = πελάτης	31
Κύριο menu.....	33
Λίστα επιχειρήσεων ανά κατηγορία.....	34
Προφίλ επιχείρησης.....	34
Εκδήλωση ενδιαφέροντος για τις προσφορές μιας επιχείρησης.....	36
Λίστα αγαπημένων επιχειρήσεων	37
Διεπαφή ενεργών προσφορών των επιχειρήσεων που έχει εκδηλώσει ενδιαφέρον.....	39
Διεπαφή ειδοποιήσεων (notifications) για εύρεση σε κοντινή απόσταση καταστημάτων με προσφορές	42
Προβολή προφίλ χρήστη.....	43
Συμπεράσματα και μελλοντικές επεκτάσεις.....	44
Βιβλιογραφία – Διαδικτυακές πηγές.....	45

Περίληψη

Τα έξυπνα κινητά έχουν κατακλύσει την καθημερινότητά μας, αποτελώντας αναπόσπαστο κομμάτι της ζωής μας καθώς εξυπηρετούν πλέον το μεγαλύτερο ποσοστό των αναγκών μας όσον αφορά την επικοινωνία, την πληροφόρηση και τη διασκέδαση.

Η παρούσα μεταπτυχιακή διατριβή, αξιοποιώντας τη χρησιμότητα των smartphones, έχει ως κύριο στόχο την ανάπτυξη μιας ψηφιακής εφαρμογής σε περιβάλλον Android.

Η εφαρμογή αφορά προσφορές επιχειρήσεων (Happy Hour Offers)

Μέσω της εφαρμογής, στους επιχειρηματίες δίνεται η δυνατότητα να εισαγάγουν νέες προσφορές καθώς και να ενημερώνουν τους ενδιαφερομένους πελάτες.

Οι πελάτες μπορούν να ενημερώνονται για προσφορές που βρίσκονται σε κοντινή τους απόσταση ή για τις προσφορές των επιχειρήσεων που έχουν εκδηλώσει ενδιαφέρον.

Έπειτα από μια ανασκόπηση των βασικών τεχνολογικών εργαλείων που χρησιμοποιήθηκαν, παρουσιάζεται το κύριο αποτέλεσμα της εργασίας, που είναι μια mobile εφαρμογή, βασισμένη στο λειτουργικό σύστημα Android, ικανή να παρουσιάζει στον τελικό χρήστη τη δυνατότητα ενημέρωσης για προσφορές- Happy Hour Offers, των επιχειρήσεων (cafe – restaurants).

Παρουσιάζονται οι βασικές οθόνες και λειτουργικότητες, καθώς και ο τρόπος υλοποίησης της εφαρμογής.

Abstract

Smartphone has impacted almost all walk of human life. The prominent areas, where impacts of Smartphone are obvious include business, education, health and social life.

This postgraduate dissertation aims to implement an android application for business offers (Happy Hour Offers)

Through the applications business users are given the opportunity to introduce new offers as well as the ability to inform the interested customers.

Customers can be informed about offers that are nearby or about the offers of companies that have expressed interest.

After a review of the basic technological tools used, the main result of the work is presented, which is a mobile application, based on the Android operating system. The application gives the opportunity to be informed about offers - Happy Hour Offers, of businesses (cafe - restaurants).

The basic screens and functionalities are presented, as well as how to implement the application

Εισαγωγή

Η τεχνολογία όσον αφορά το χώρο της κινητής τηλεφωνίας εξελίσσεται με ραγδαίους ρυθμούς . Τα τελευταία χρόνια τα έξυπνα τηλέφωνα (smartphones) ενσωματώνουν

δυνατότητες που μέχρι σήμερα δεν υπήρχαν, με αποτέλεσμα οι κινητές συσκευές να εισχωρούν όλο και περισσότερο στη ζωή μας. Τα έξυπνα τηλέφωνα (smartphones) είναι κινητά τηλέφωνα βασισμένα σε ένα λειτουργικό σύστημα κινητής τηλεφωνίας με περισσότερη προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα απλό κινητό τηλέφωνο.

Πλέον τα σύγχρονα έξυπνα τηλέφωνα διαθέτουν όλα οθόνες αφής υψηλής ανάλυσης και διαδικτυακούς περιηγητές (web browsers) που εμφανίζουν τυποποιημένες ιστοσελίδες καθώς και βελτιστοποιημένες ιστοσελίδες για κινητά.

Τη σημερινή εποχή υπάρχει πληθώρα ηλεκτρονικών εφαρμογών και υπηρεσιών που έχουν ως στόχο καλύψουν τις αυξημένες ανάγκες των χρηστών. Ο χρήστης επιθυμεί να λαμβάνει την πληροφορία που τον ενδιαφέρει ταχύτατα.

Συνεπώς η ευκολία στη χρήση και η πληθώρα δυνατοτήτων που παρέχεται, καθιστά τα smartphones ένα απαραίτητο εργαλείο για το μεγαλύτερο μέρος του πληθυσμού και πλέον αναπόσπαστο κομμάτι της καθημερινότητας του.

Η παρούσα μεταπτυχιακή διατριβή περιγράφει όλα τα βήματα για την ανάπτυξη και τον σχεδιασμό εφαρμογής για έξυπνα τηλέφωνα (smartphones) σε λειτουργικό σύστημα Android.

Το όνομα της εφαρμογής είναι Happy Hour Offres, και απευθύνεται απευθύνεται σε επιχειρηματίες (ιδιοκτήτες cafe restaurant) και σε πελάτες – καταναλωτές που ενδιαφέρονται να ενημερώνονται έγκαιρα.

Οι επιχειρηματίες έχουν τη δυνατότητα να καταχωρούν τις προσφορές των καταστημάτων τους με σκοπό να προσελκύσουν πελάτες και οι καταναλωτές να ενημερώνονται για αυτές τις προσφορές, ούτως ώστε να μπορούν να επιλέξουν τις πιο συμφέρουσες.

Διαδεδομένα λειτουργικά συστήματα κινητών συσκευών (Android – IOS)

Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux .

Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Τα Android αρχικά αναπτύχθηκαν από μια μικρή εταιρία λογισμικού η οποία εξαγοράστηκε από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε τον Νοέμβριο του 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 79 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας.

Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Μια μεγάλη κοινότητα προγραμματιστών ασχολείται με τον προγραμματισμό στο Android και με αυτό τον τρόπο

αυξάνει τις δυνατότητες των συσκευών που το χρησιμοποιούν. Αυτή την στιγμή υπάρχουν πάνω από 200.000 εφαρμογές στο Android Market, το ηλεκτρονικό

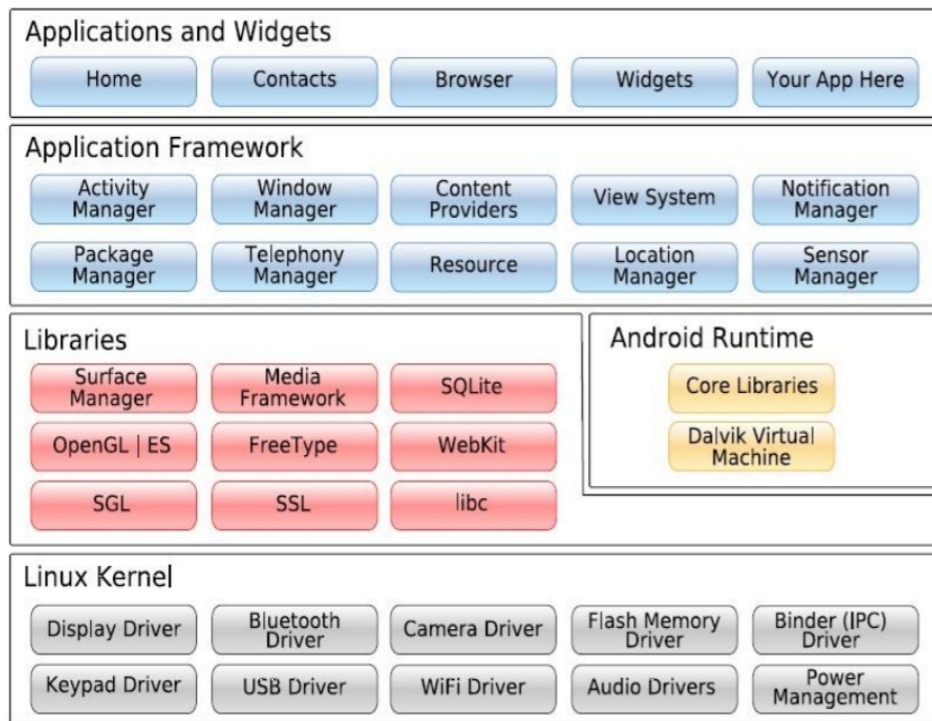
κατάστημα που έχει φτιάξει η Google, αν και υπάρχει και η δυνατότητα αγοράς εφαρμογών από τρίτες εταιρίες.

Από την στιγμή της εισόδου του στην αγορά το Android έχουν παρουσιάσει μια τεράστια αύξηση και στον αριθμό των συσκευών που το χρησιμοποιούν αλλά και του μεριδίου του στην αγορά, και αυτή την στιγμή θεωρείται το πιο διαδεδομένο λειτουργικό σύστημα για smartphones.

Για την ανάπτυξη εφαρμογών στο περιβάλλον του λειτουργικού χρησιμοποιείται το Android Software Development Kit το οποίο περιλαμβάνει ένα μεγάλο σετ από εργαλεία ανάπτυξης. Σε αυτό περιλαμβάνεται ένας debugger, βιβλιοθήκες, ένας εξομοιωτής, βιβλιογραφία, δείγματα κώδικα καθώς και σεμινάρια.

Αρχιτεκτονική του Android

Η αρχιτεκτονική του Android περιλαμβάνει τέσσερα κύρια επίπεδα. Η αρχιτεκτονική αυτή του παρέχει ευελιξία στις αλλαγές και δυνατότητα βελτίωσης. Κάθε επίπεδο είναι ανεξάρτητο από το άλλο και υποστηρίζει διαφορετικού τύπου λειτουργίες. Υπάρχουν φυσικά και μηχανισμοί επικοινωνίας μεταξύ των επιπέδων. Στην παρακάτω εικόνα φαίνονται σχηματικά τα επίπεδα που περιλαμβάνει η αρχιτεκτονική του Android:



Εικόνα 1 Αρχιτεκτονική Android

Πυρήνας Linux(Linux Kernel)

Στο κάτω μέρος των στρωμάτων είναι ο πυρήνας Linux, που παρέχει ένα επίπεδο διαχωρισμού μεταξύ του υλικού της συσκευής και περιέχει όλες τις κύριες λειτουργίες του λειτουργικού συστήματος. Οι λειτουργίες αυτές αφορούν διαχείριση μνήμης, διαχείριση

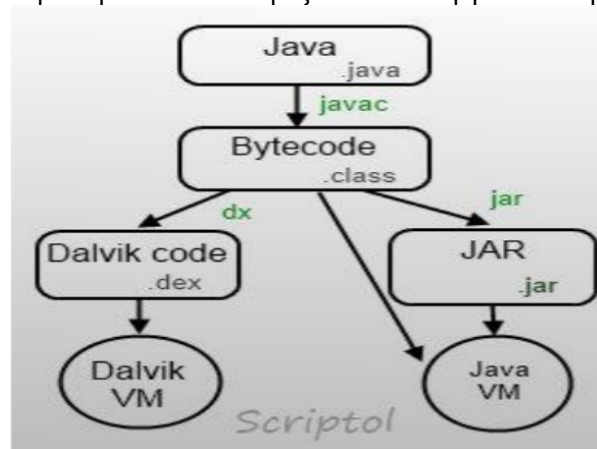
διεργασιών, λειτουργίες δικτύου, ασφάλεια του λειτουργικού, και ένα σύνολο οδηγών υλικού (hardware drivers), όπως κάμερα, πληκτρολόγιο, οθόνη κ.α.

Βιβλιοθήκες(Libraries)

Στο επόμενο επίπεδο βρίσκονται οι βιβλιοθήκες. Το σύνολο σχεδόν των βιβλιοθηκών είναι γραμμένο σε C και C++, οι οποίες έχουν μεταγλωττιστεί για τη χρήση τους από το λειτουργικό. Μερικές από τις κύριες βιβλιοθήκες του Android είναι η βιβλιοθήκη WebKit για την υποστήριξη των εφαρμογών του ιστότοπου, η Surface Manager που διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συνθέτει απρόσκοπτα δισδιάστατα και τρισδιάστατα επίπεδα γραφικών τα οποία προέρχονται από πολλαπλές εφαρμογές, η LibWebCore μια μοντέρνα μηχανή υποστήριξης πλοήγησης στο διαδίκτυο (browser engine) η οποία χρησιμοποιείται και από τον ενσωματωμένο browser του Android αλλά και από τις WebViews που ενσωματώνονται στις εφαρμογές, η SGL η γνωστή μηχανή δισδιάστατων γραφικών και η SQLite – μια πανίσχυρη και συνάμα πολύ ελαφριά σχεσιακή βάση δεδομένων.

Χρόνος Εκτέλεσης Εφαρμογής(Android Runtime)

Αυτό είναι το τρίτο τμήμα της αρχιτεκτονικής στο δεύτερο επίπεδο, το οποίο αποτελείται από μια εικονική μηχανή (Dalvik Virtual Machine) που είναι υπεύθυνη για την δημιουργία των εκτελέσιμων αρχείων των εφαρμογών προκειμένου να τα «τρέξει» το λειτουργικό σύστημα και τις βιβλιοθήκες.

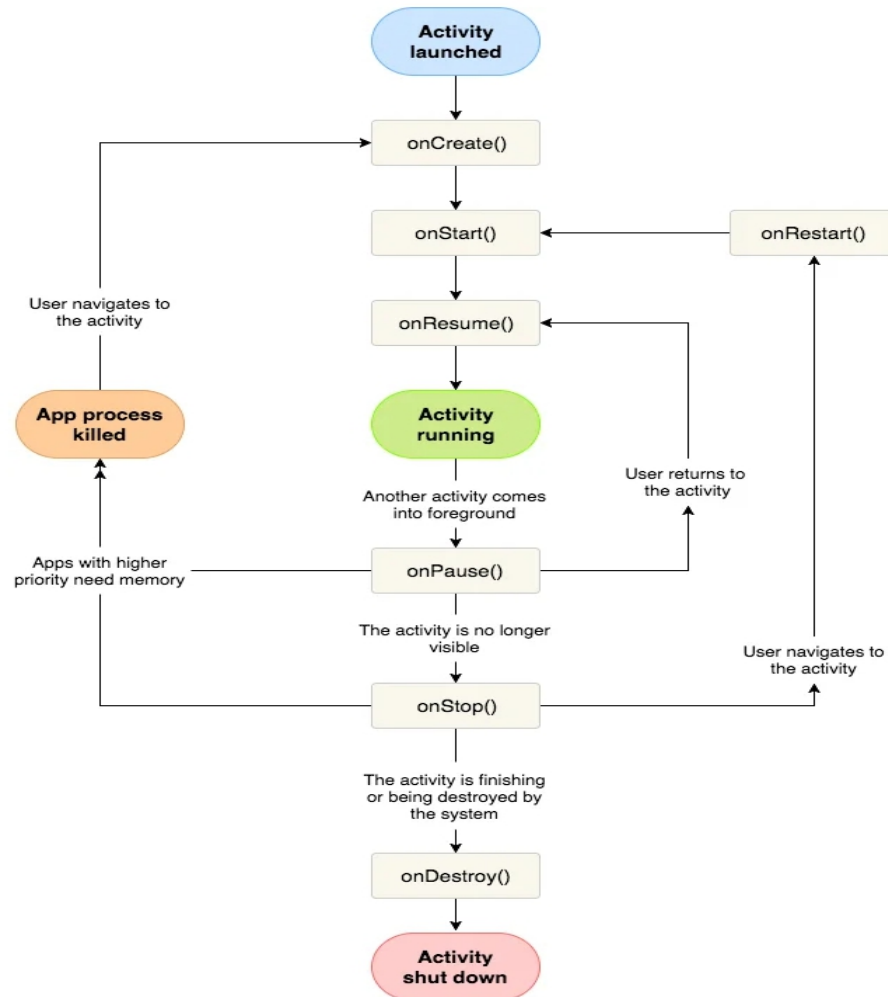


Πλαίσιο Εφαρμογής(Application Framework)

Το πλαίσιο εφαρμογής βρίσκεται στο τρίτο στρώμα της αρχιτεκτονικής του Android. Πρόκειται ουσιαστικά για μια ενσωματωμένη εργαλειοθήκη, που παρέχει ένα σύνολο υπηρεσιών για τις Android εφαρμογές. Όλες αυτές οι υπηρεσίες δίνουν τη δυνατότητα στους προγραμματιστές των εφαρμογών να αναπτύξουν καινοτόμες και πλούσιες σε υλικό, εφαρμογές. Οι προγραμματιστές έχουν στην διάθεση τους τη δυνατότητα ελέγχου του υλικού της συσκευής και μέσω αυτής μπορούν να αποκτήσουν πρόσβαση σε υπηρεσίες εντοπισμού, εκτέλεση διεργασιών παρασκηνίου, και πάρα πολλές ακόμη δυνατότητες.

Ακόμα έχουν πρόσβαση σε όλα τα APIs μεταξύ αυτών και στα κύρια APIs που χρησιμοποιούν οι ενσωματωμένες εφαρμογές. Η δομή των εφαρμογών είναι τέτοια που ευνοείται η επαναχρησιμοποίηση δομικών συστατικών, και επίσης επιτρέπεται η χρήση των δυνατοτήτων τις μίας εφαρμογής από άλλες εφαρμογές, βέβαια κάτω από τις προδιαγραφές ασφάλειας του Android. Τα σημαντικότερα δομικά στοιχεία του πλαισίου εφαρμογών είναι

- **Σύστημα προβολών (View System)**: αποτελεί ένα εκτενές σύνολο από αντικείμενα GUI τα οποία μπορούν να χρησιμοποιηθούν κατά το σχεδιασμό μιας εφαρμογής. Παραδείγματα προβολών είναι οι λίστες (listView), το πλέγμα (GridView), πεδία εισαγωγής κειμένου, κουμπιά, κλπ
- **Πάροχος Περιεχομένου (Content Provider)**: δίνει τη δυνατότητα στις εφαρμογές να μοιράζονται ή να ανταλλάσσουν δεδομένα μιας συγκεκριμένης μορφής η οποία ορίζεται από τον πάροχο. Παραδείγματα δεδομένων, είναι οι επαφές χρήστη και οι βάσεις δεδομένων των εφαρμογών.
- **Διαχειριστής Πόρων (Resource Manager)**: παρέχει πρόσβαση σε υλικό το οποίο δεν είναι σε μορφή κώδικα όπως πχ, εικόνες, αρχεία xml, πίνακες χαρακτήρων, κλπ
- **Διαχειριστής Ειδοποιήσεων (Notification Manager)**: δίνει στις εφαρμογές πρόσβαση στις υπηρεσίες ειδοποιήσεων χρήστη. Τέτοιες είναι οι ειδοποιήσεις στη notification bar, τα toast μηνύματα στο κάτω μέρος της οθόνης, η δόνηση του κινητού και η ενεργοποίηση της οθόνης, κλπ
- **Διαχειριστής Τοποθεσίας (Location Manager)**: δίνει στις εφαρμογές τη δυνατότητα εντοπισμού της τοποθεσίας μη τη χρήση του GPS η των κεραιών των κυψελωτών δικτύων.
- **Διαχειριστής Δραστηριοτήτων (Activity Manager)**: διαχειρίζεται τον κύκλο ζωής των δραστηριοτήτων και παρέχει δυνατότητα πλοήγησης από δραστηριότητα σε δραστηριότητα κρατώντας αποθηκευμένη στη μνήμη τη σειρά εκτέλεσης αυτών. Στο παρακάτω σχεδιάγραμμα φαίνεται λεπτομερώς ο κύκλος ζωής κάθε δραστηριότητας:



Εικόνα 2 Κύκλος ζωής activity

Εφαρμογές (Applications)

Αυτό είναι το τέταρτο και τελευταίο επίπεδο και συναντάμε τις εφαρμογές που θα χρησιμοποιούν τελικά οι χρήστες με διαφάνεια ως προς το τι συμβαίνει πίσω από αυτές ή το τι απαιτείται για την εκτέλεση τους από το λειτουργικό σύστημα. Μερικές από τις πιο γνωστές από τις εφαρμογές αυτές είναι ο φυλλομετρητής, διαχειριστής email, διαχείριση μηνυμάτων, προβολή χαρτών σε συνδυασμό με το στίγμα της συσκευής εάν διαθέτει δέκτη GPS, ημερολόγιο, διαχείριση επαφών, παιχνίδια και πολλές άλλες.

Η Google εξελίσσει διαρκώς το λειτουργικό της, βγάζοντας καινούριες εκδόσεις:



Εικόνα 3 εκδόσεις android

IOS

Το iOS (γνωστό και ως iPhone OS) είναι το λειτουργικό σύστημα για κινητές πλατφόρμες της Apple. Αν και αρχικά αναπτύχθηκε μόνο για το iPhone έχει από τότε επεκταθεί ώστε να υποστηρίζει και άλλες συσκευές της Apple όπως τα iPod Touch και iPad. Το συγκεκριμένο λειτουργικό σύστημα δεν υποστηρίζει άλλες συσκευές εκτός από αυτές της Apple. Ένα από τα μεγάλα πλεονεκτήματα του είναι το App Store το οποίο περιέχει πληθώρα εφαρμογών.

Το περιβάλλον χρήσης του είναι βασισμένο στην άμεση αλληλεπίδραση του χρήστη με την οθόνη αφής της συσκευής. Με αυτόν τον τρόπο ο χειρισμός γίνεται πολύ ευχάριστος, γρήγορος αλλά και απλός για τον χρήστη αφού μπορεί να αλληλεπιδρά με φυσικότητα με τα αντικείμενα που προβάλλονται στην οθόνη. Για παράδειγμα ο χρήστης μέσω της οθόνης αφής πολλαπλών σημείων μπορεί να χρησιμοποιεί διάφορες κινήσεις των δακτύλων του και να παίρνει άμεσα τα αποτελέσματα στην οθόνη.

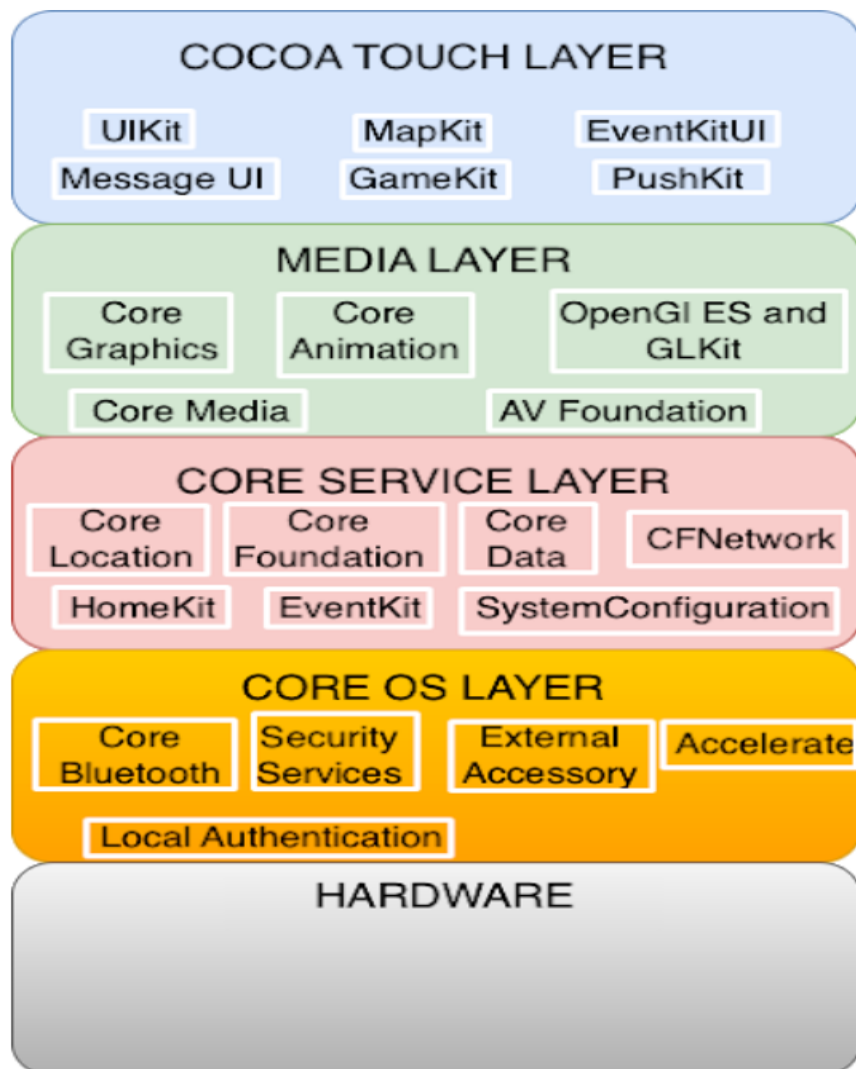
Το iOS, για την ανάπτυξη εφαρμογών στο περιβάλλον του, χρησιμοποιεί το λογισμικό ανάπτυξης εφαρμογών iOS SDK το οποίο αναπτύχθηκε από την Apple και δόθηκε στους προγραμματιστές τον Φεβρουάριο του 2008. Τους δίνει την δυνατότητα να δημιουργήσουν εφαρμογές και να τις δοκιμάσουν σε ένα εξομοιωτή που ονομάζεται iPhone Simulator. Όμως για την εγκατάσταση μια εφαρμογής στη συσκευή, καθώς και για την πώληση της μέσω του App Store πρέπει ο χρήστης να είναι εγγεγραμμένος στο πρόγραμμα των προγραμματιστών iPhone. Ο δημιουργός μιας εφαρμογής μπορεί να την πουλήσει σε οποιαδήποτε τιμή πάνω από την μικρότερη επιτρεπτή τιμή (0.99 ευρώ) και να έχει ένα ποσοστό κέρδους και το υπόλοιπο κέρδος ανήκει στην Apple. Εναλλακτικά, μπορεί να δίνει την εφαρμογή δωρεάν και να μην ζημιώνεται καθόλου από τα έξοδα κυκλοφορίας και διανομής, εκτός βεβαίως από τα έξοδα εγγραφής. Το iOS SDK χρησιμοποιεί τον ίδιο πρόγραμμα γραφής κώδικα που χρησιμοποιεί και το Mac OS X, το Xcode, και περιλαμβάνει και τον iPhone Simulator, ένα πρόγραμμα που μπορεί να χρησιμοποιηθεί για να εξομοιώσει το πως θα φαινόνταν οι εφαρμογές και το

πως θα δούλευαν αν έτρεχαν στο iPhone, και όλα αυτά από υπολογιστή του προγραμματιστή. Το SDK της Apple έχει ως απαιτήσεις συστήματος για να χρησιμοποιηθεί, έναν Intel Mac με λειτουργικό σύστημα Mac OS X Leopard ή και

νεότερο. Άλλα λειτουργικά όπως τα Windows αλλά και παλιότερες εκδόσεις Mac OS X δεν υποστηρίζονται .

Αρχιτεκτονική του iOS

Η αρχιτεκτονική του iOS αποτελείται από τέσσερα επίπεδα - στρώματα: Cocoa Touch, Media, Core Services, Core OS. Τα χαμηλά επίπεδα του iOS προσφέρουν τις βασικές υπηρεσίες στις οποίες στηρίζονται όλες οι εφαρμογές, ενώ τα υψηλότερα στρώματα περιέχουν υπηρεσίες και τεχνολογίες σχετικές με εξελιγμένα γραφικά και διεπαφές χρήστη.



Εικόνα 4 αρχιτεκτονική ios

Στρώμα Cocoa Touch

Το στρώμα κακάο Touch περιέχει βασικά πλαίσια για την οικοδόμηση iOS εφαρμογών. Τα πλαίσια αυτά καθορίζουν την εμφάνιση της εφαρμογής. Μπορούν επίσης να παρέχουν τη βασική υποδομή της εφαρμογής και υποστήριξη για τις βασικές τεχνολογίες, όπως multitasking, είσοδο αφής, push notifications και πολλές υπηρεσίες του συστήματος υψηλού επιπέδου.

Στρώμα Media

Το στρώμα Media περιέχει τα γραφικά, ήχο, βίντεο. Η διαχείριση των πολυμέσων γίνεται από διάφορα API μερικά από τα οποία είναι το OpenAL (open audio library) το οποίο είναι ένα λογισμικό που χρησιμοποιείται σε πολλές διαφορετικές πλατφόρμες και προγραμματιστές παιχνιδιών συχνά χρησιμοποιούν αυτή την τεχνολογία για την παροχή υψηλής ποιότητας ήχου. Είναι σχεδιασμένο έτσι ώστε να αποδίδει ποιοτικό πολυκάναλο τρισδιάστατο ήχο. Το Core Animation το οποίο παράγει κινούμενα περιβάλλοντα χρήσης. Το AVKit Framework που χρησιμεύει στην παρουσίαση βίντεο. Το πλαίσιο αυτό υποστηρίζει τόσο πλήρους οθόνης και αναπαραγωγή βίντεο μερική-οθόνη.

Στρώμα Core Services

Το στρώμα Core Services περιέχει θεμελιώδεις υπηρεσίες του συστήματος για τις εφαρμογές. Κλειδί μεταξύ αυτών των υπηρεσιών είναι το Core Foundation και το Foundation frameworks που ορίζουν τον τύπο τις εφαρμογής. Αυτό το στρώμα περιλαμβάνει επίσης ατομικές τεχνολογίες για να υποστηρίξει λειτουργίες όπως η τοποθεσία, το iCloud, κοινωνικών μέσων μαζικής ενημέρωσης και δικτύωσης.

Στρώμα Core OS

Το Στρώμα Core OS περιέχει τα χαμηλού επιπέδου χαρακτηριστικά στα οποία πολλές τεχνολογίες είναι χτισμένες. Αποτελείται από το TCP/IP για την διασύνδεση των εφαρμογών με το διαδίκτυο, τα Sockets, την διαχείριση ενέργεια, το σύστημα αρχείων και την ασφάλεια των δεδομένων

Τεχνολογίες, Εργαλεία, Πρωτόκολλα ανάπτυξης εφαρμογής

Εργαλεία Ανάπτυξης της Εφαρμογής

Java

Για την εκπόνηση της εργασίας αξιοποιήθηκε η γλώσσα προγραμματισμού Java. Πρόκειται για μια αντικειμενοστραφή γλώσσα προγραμματισμού γενικού σκοπού που αναπτύχθηκε και σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων γλωσσών είναι ότι τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα, δηλαδή προσφέρει την ανεξαρτησία λειτουργικού συστήματος και πλατφόρμας.

Σύμφωνα με την περιγραφή που έχει δοθεί από την εταιρεία Sun για τη γλώσσα προγραμματισμού Java, είναι μια απλή, αντικειμενοστραφής, καταμεμημένη, υψηλής απόδοσης, συμπαγής, ασφαλής, ανεξάρτητης αρχιτεκτονικής, μεταφέρσιμη, υψηλής απόδοσης και ιδιαίτερα δυναμική.

Software Development Kit(SDK)

Ένα kit ανάπτυξης λογισμικού (SDK) είναι συνήθως ένα σύνολο εργαλείων ανάπτυξης λογισμικού που επιτρέπει τη δημιουργία εφαρμογών για ένα συγκεκριμένο πακέτο λογισμικού, framework λογισμικού, πλατφόρμα υλικού, σύστημα υπολογιστή, κονσόλα παιχνιδιών βίντεο, λειτουργικό σύστημα ή παρόμοια πλατφόρμα ανάπτυξης. Για να εμπλουτιστούν οι εφαρμογές με προηγμένες λειτουργίες, διαφημίσεις, ειδοποιήσεις push και πολλά άλλα, οι περισσότεροι προγραμματιστές εφαρμογών υλοποιούν συγκεκριμένα kit ανάπτυξης λογισμικού. Πιο συγκεκριμένα, περιλαμβάνει ένα πρόγραμμα εντοπισμού σφαλμάτων, βιβλιοθήκες, έναν εξομοιωτή συσκευής (εικονική συσκευή) που βασίζεται στο QEMU, τεκμηρίωση, δείγματα κώδικα και tutorials. Προκειμένου όμως να αξιοποιηθούν οι δυνατότητες του SDK απαιτείται ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE-Integrated Development Environment), όπως για παράδειγμα το “Android Studio”.

Java Development Kit(JDK)

Το Java Development Kit (JDK) προσφέρει εργαλεία τα οποία χρησιμοποιούνται από ολοκληρωμένα περιβάλλοντα IDEs καθώς και από διάφορα πακέτα ανάπτυξης λογισμικού SDKs, όπως είναι ο μεταγλωττιστής Java. Το JDK παρέχει τη δυνατότητα σε προγράμματα Java να λειτουργήσουν κανονικά στο σύστημα, καθώς περιλαμβάνει ένα περιβάλλον εκτέλεσης Java (Java Runtime Environment – JRE). Το JRE ή διαφορετικά το ιδιωτικό περιβάλλον εκτέλεσης περιλαμβάνει μια εικονική μηχανή (Java Virtual Machine) καθώς και βιβλιοθήκες κλάσεων οι οποίες είναι ιδιαίτερα χρήσιμες στους προγραμματιστές.

Android Studio

Το Android Studio είναι το επίσημο περιβάλλον ανάπτυξης για εφαρμογές Android. Πρωτοεμφανίστηκε 16 Μαΐου 2013 σε συνέδριο της Google και είναι βασισμένο στο λογισμικό JetBrains-Intelli J IDEA ενώ είναι κατασκευασμένο σε Java. Επίσης εμπεριέχει το Android SDK το οποίο περιλαμβάνει τις βιβλιοθήκες που χρειάζονται για την ανάπτυξη των εφαρμογών. Το πρόγραμμα αυτό είναι διαθέσιμο δωρεάν για Windows, Mac OS και Linux, και αντικατέστησε το Eclipse Android Development Tools (ADT), το οποίο μέχρι το 2014 αποτελούσε το κύριο IDE για κατασκευή λογισμικού Android.

Η κυκλοφορία αυτού του νέου προγραμματιστικού περιβάλλοντος ανακοινώθηκε από την Google στις 16 Μαΐου 2013, στο Συνέδριο Google I/O, ενώ η κατασκευή του βρισκόταν ακόμα σε πρώιμο στάδιο. Η πρώτη σταθερή έκδοση του Android Studio ήταν διαθέσιμη για το κοινό το Δεκέμβριο του 2014 (έκδοση 1.0).

Το Android Studio προσφέρει διάφορες υπηρεσίες στους χρήστες, κάποιες από αυτές είναι οι εξής:

- Ένα ευέλικτο Cradle-based σύστημα κατασκευής.
- Ένα γρήγορο και με πολλές λειτουργίες εξομοιωτή (emulator).
- Δυνατότητα ανάπτυξης εφαρμογών για όλες τις συσκευές Android (Smartphones, Smart watches , Smart TV's).
- Περιέχει εργαλεία ελέγχου της απόδοσης , της χρηστικότητας και και έλεγχο συμβατότητας των εφαρμογών. •
- Υποστήριξη C++
- Υποστήριξη Google Cloud Platform.

XML

Το XML προκύπτει από τις λέξεις eXtensible Markup Language και σχεδιάστηκε με σκοπό την αποθήκευση και μεταφορά δεδομένων, με τρόπο ανεξάρτητο του υλικού και του λογισμικού. Έχει κοινά στοιχεία με την HTML, αλλά και διαφορές απ' αυτήν.

Χαρακτηριστικό της XML αποτελεί το γεγονός ότι είναι επεκτάσιμη κι επικεντρώνεται στην περιγραφή του είδους των δεδομένων που μεταφέρει. Στην εφαρμογή τα xml αρχεία χρησιμοποιούνται για σχεδίαση των διεπαφών (οθόνες με κουμπιά και λίστες που διευκολύνουν το χρήστη στις επιλογές του) και υπάρχουν στον υποφάκελο layout του φακέλου res.

Firestore

Τον Οκτώβριο του 2017, η Firestore κυκλοφόρησε το Cloud Firestore, μια βάση δεδομένων εγγραφών σε πραγματικό χρόνο ως προϊόν που διαδέχθηκε την αρχική βάση δεδομένων Firestore Realtime.

Υπηρεσίες

1. Firestore Cloud Messaging

Γνωστό και με την ονομασία Google Cloud Messaging (GCM), το Firestore Cloud Messaging (FCM) είναι μια λύση πολλαπλών πλατφορμών για μηνύματα και ειδοποιήσεις σε εφαρμογές Android, iOS και ιστού. Από το 2016 μπορούν να χρησιμοποιηθούν είναι δωρεάν

2. Firestore Authentication

Πρόκειται για μια υπηρεσία πιστοποίησης των χρηστών. Υποστηρίζει κοινωνικούς παρόχους σύνδεσης όπως το Facebook, και Google κ.α., καθώς και άλλους παρόχους υπηρεσιών όπως τα παιχνίδια Google Play, Apple, Yahoo κ.α. Επιπλέον, περιλαμβάνει ένα σύστημα διαχείρισης χρήστη με το οποίο οι προγραμματιστές μπορούν να επιτρέπουν τον έλεγχο ταυτότητας χρήστη με email και κωδικό πρόσβασης που είναι αποθηκευμένα στο Firestore.

3. Firestore Realtime Database

Είναι μία βάση δεδομένων σε πραγματικό χρόνο. Η υπηρεσία παρέχει στους προγραμματιστές εφαρμογών ένα API που επιτρέπει στα δεδομένα εφαρμογών να συγχρονίζονται μεταξύ των πελατών και να αποθηκεύονται στο cloud του Firestore. Η εταιρεία παρέχει βιβλιοθήκες πελάτη που επιτρέπουν την ενσωμάτωση με Android, iOS, το JavaScript, Java, Objective-C, Swift και Node.js εφαρμογές. Η βάση δεδομένων είναι επίσης προσβάσιμη μέσω ενός REST API και συνδέσμων για διάφορα πλαίσια JavaScript όπως Angular.js, React και άλλα. Οι προγραμματιστές που θέλουν να χρησιμοποιούν τη βάση δεδομένων σε πραγματικό χρόνο μπορούν να προστατεύσουν τα δεδομένα τους χρησιμοποιώντας τους κανόνες ασφαλείας που εφαρμόζονται από τον διακομιστή.

4. Cloud Firestore

Τον Ιανουάριο του 2019, το Cloud Firestore κυκλοφόρησε επίσημα από το beta. Πρόκειται για το διάδοχο του αρχικού συστήματος βάσεων δεδομένων του Firestore, της βάσης δεδομένων σε πραγματικό χρόνο. Παρέχει τη δυνατότητα για ένθετα έγγραφα και πεδία και όχι την ιεραρχημένη προβολή που παρέχεται στη βάση δεδομένων σε πραγματικό χρόνο.

5. Firebase Storage

Παρέχει ασφαλείς μεταφορτώσεις αρχείων και λήψεις για εφαρμογές Firebase προς χρήση για την αποθήκευση εικόνων, ήχου, βίντεο ή άλλου περιεχομένου που δημιουργείται από χρήστες. Υποστηρίζεται από το Google Cloud Storage

6. Firebase Hosting

Μία στατική αλλά και δυναμική υπηρεσία φιλοξενίας ιστοσελίδων που ξεκίνησε το Μάιο του 2014. Υποστηρίζει τη φιλοξενία στατικών αρχείων όπως CSS, HTML, JavaScript και άλλα αρχεία, καθώς και υποστήριξη μέσω του Cloud Functions. Η υπηρεσία παρέχει αρχεία μέσω δικτύου παράδοσης περιεχομένου (CDN) μέσω HTTP Secure (HTTPS) και Secure Sockets Layer encryption (SSL). Η εταιρεία δηλώνει ότι το Firebase Hosting αναπτύχθηκε από αιτήματα πελατών. Αυτό γιατί οι προγραμματιστές που χρησιμοποιούν το Firebase για τη βάση δεδομένων σε πραγματικό χρόνο συνήθως χρειάζονται και ένα μέρος για να φιλοξενήσουν το περιεχόμενό τους.

7. ML Kit

Ένα σύστημα κινητής μηχανικής εκμάθησης για προγραμματιστές που κυκλοφόρησε το Μάιο του 2018, σε έκδοση beta κατά τη διάρκεια του Google I/O 2018. Τα ML Kit API διαθέτουν μια ποικιλία χαρακτηριστικών, όπως αναγνώριση οπτικών χαρακτήρων, ανίχνευση προσώπων, σάρωση γραμμωτών κωδικών, επισήμανση εικόνων και αναγνώριση ορόσημων. Προς το παρόν είναι διαθέσιμο για προγραμματιστές iOS ή Android.

8. Crashlytics

Το Crash Reporting δημιουργεί λεπτομερείς αναφορές σφαλμάτων στην εφαρμογή. Τα σφάλματα ομαδοποιούνται σε ομάδες παρόμοιων ιχνών στοιβάς και ενεργοποιούνται από τη σοβαρότητα του αντίκτυπου στους χρήστες της εφαρμογής. Εκτός από τις αυτόματες αναφορές, ο προγραμματιστής μπορεί να καταγράψει προσαρμοσμένα συμβάντα για να καταγράψει τα βήματα που οδηγούν σε σφάλμα. Πριν αποκτήσει το Crashlytics, το Firebase χρησιμοποιούσε την υπηρεσία Firebase Crash Reporting.

9. Άλλες Υπηρεσίες

Το Firebase διαθέτει και άλλες υπηρεσίες όπως το Firebase Performance, που παρέχει πληροφορίες σχετικά με την απόδοση μιας εφαρμογής και τους χρόνους καθυστέρησης που βιώνουν οι χρήστες της εφαρμογής και το Firebase Test Lab, που παρέχει υποδομή που βασίζεται σε σύννεφο για τη δοκιμή εφαρμογών Android και iOS σε μια μεγάλη ποικιλία συσκευών και διαμορφώσεων συσκευών.

Παρουσίαση και χρήση της εφαρμογής

Στο κεφάλαιο αρχικά θα αναφερθούμε στην ανάλυση απαιτήσεων, το σχεδιασμό της εφαρμογής και της βάσης δεδομένων, ολοκληρώνοντας το κεφάλαιο με τον τρόπο υλοποίησης της.

Απαιτήσεις

Στην παρούσα ενότητα γίνεται περιγραφή της ανάλυσης απαιτήσεων της εφαρμογής που αναπτύχθηκε.

- Εγγραφή χρήστη στο σύστημα Ο χρήστης μπορεί να είναι είτε ο επιχειρηματίας (business) , είτε ο καταναλωτής (customer)
- Ταυτοποίηση χρήστη.
- Δυνατότητα δημιουργίας προσφορών , καθορισμός τύπου και διάρκειας αυτών.(Για τους επιχειρηματίες)
- Ο καταναλωτής/πελάτης έχει τη δυνατότητα να βλέπει τη λίστα με τα καταστήματα ανά κατηγορία,

και να δηλώνει ενδιαφέρον σε αυτά , ούτως ώστε να λαμβάνει ειδοποίηση με την προσθήκη κάποιας νέας προσφοράς.

- Ενημέρωση χρήστη εάν βρίσκεται πλησίον καταστήματος με προσφορά.
- Λίστα με τις προσφορές για τις οποίες έχει λάβει ενημέρωση.
- Δυνατότητα πλοήγησης στις ενεργές προσφορές των καταστημάτων όπου έχει δηλώσει ενδιαφέρον.
- Χρήση Google Map για τον εύκολο εντοπισμό της επιχείρησης
- Κατάλληλη ενημέρωση για τις νέες προσφορές καταστημάτων όπου είναι εγγεγραμμένος.

Σύνδεση και Δημιουργία νέου λογαριασμού.

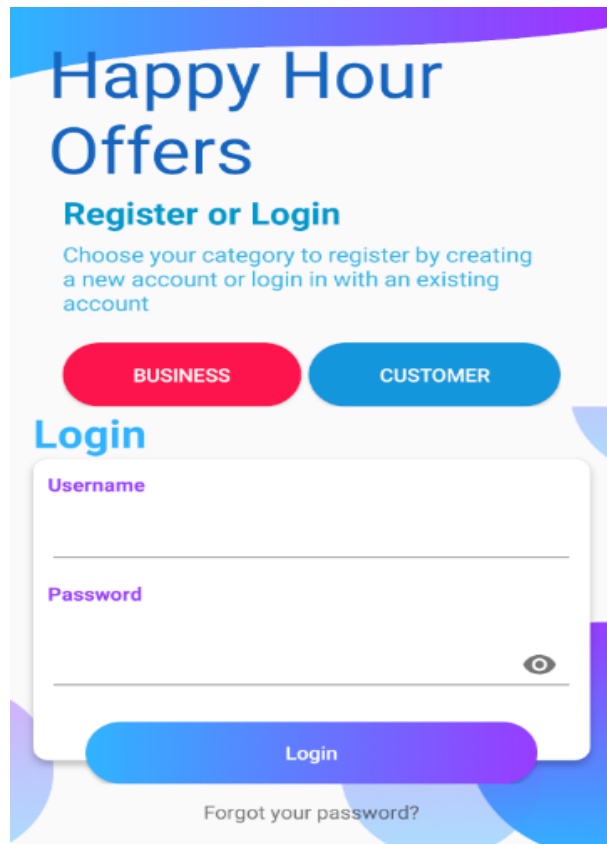
Προϋπόθεση για την χρήση της πλατφόρμας είναι η εγγραφή του χρήστη στο σύστημα. Την πρώτη φορά που θα ανοίξει ο χρήστης την εφαρμογή θα αντικρίσει μια οθόνη για την ταυτοποίηση του . Οι χρήστες που δεν έχουν εγγραφεί ακόμα μπορούν να επιλέξουν την κατηγορία χρήστη που προτρέπει στην οθόνη εγγραφής .Δίνεται η δυνατότητα εγγραφής για επιχειρηματία ή πελάτη.

Σε περίπτωση που ο χρήστης έχει ήδη λογαριασμό, για να συνδεθεί στην εφαρμογή, πρέπει να εισάγει το email και το password και στη συνέχεια να πατήσει το κουμπί Login.

Ο αλγόριθμος κατά την υποβολή της εγγραφής προχωράει ως εξής:

Γίνεται ο έλεγχος των πεδίων για τυχόν μη αποδεκτές τιμές (validation). Αν τα πεδία είναι αποδεκτά τότε ο αλγόριθμος αποστέλλει το αίτημα στον εξυπηρετητή. Στην περίπτωση σφάλματος εμφανίζεται ανάλογο μήνυμα στον χρήστη.

Αν το αίτημα γίνει αποδεκτό τότε εμφανίζεται ανάλογο μήνυμα και ο χρήστης μεταφέρεται στην κύρια οθόνη ανάλογα με τον τύπο του χρήστη (business ή customer).



Εγγραφή(register) για user type = business (επιχειρηματίας)

Με την επιλογή κατηγορίας `business` δημιουργείται ένα `object Intent`, το οποίο είναι υπεύθυνο για τη μετάβαση της εφαρμογής στο επόμενο `activity` (`BusinessAuthActivity`).

Ο χρήστης έχει τη δυνατότητα να επιλέξει το ακριβές σημείο όπου βρίσκεται η επιχείρησή του στο χάρτη.

Μπορεί να εισάγει φωτογραφία της επιχείρησής του.

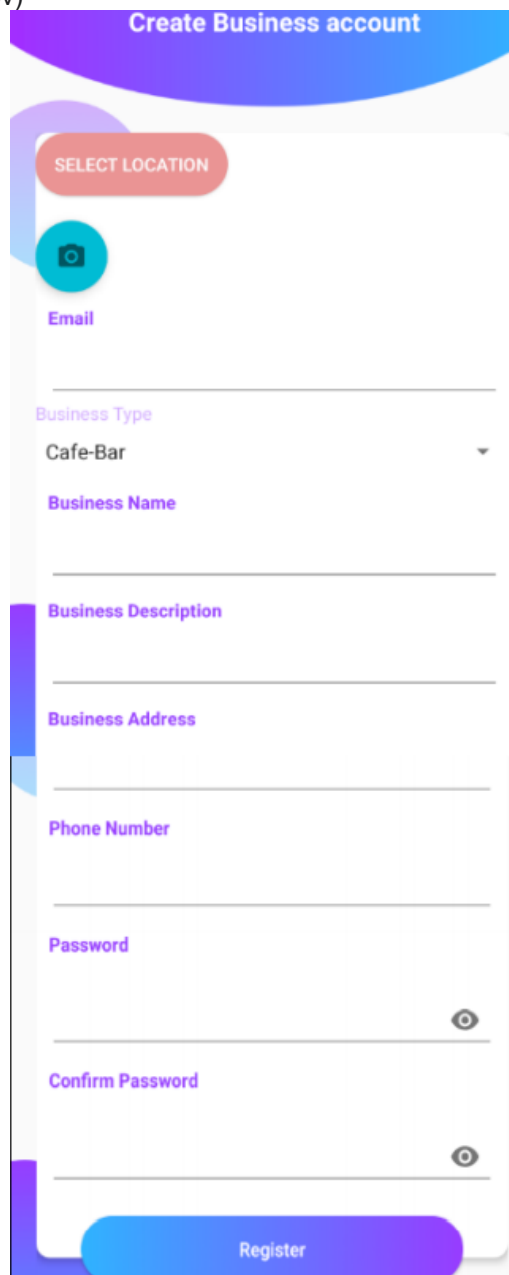
Απαραίτητα στοιχεία για την εγγραφή αποτελούν τα:

- διεύθυνση ηλεκτρονικού ταχυδρομείου
- τύπος επιχείρησης (`cafe-bar` ή `restaurant`)
- όνομα επιχείρησης
- σύντομη περιγραφή επιχείρησης
- διεύθυνση επιχείρησης
- τηλέφωνο
- κωδικός και επιβεβαίωση αυτού

Τα πεδία που απαιτούνται για την εγγραφή εμφανίζονται σε ένα `scrollView`.

Μετά την επιτυχή εγγραφή του χρήστη μεταφέρεται στο κύριο activity όπου υπάρχει η λίστα με τις προσφορές του. Αρχικά δεν υπάρχει καμία οπότε ένα μήνυμα εμφανίζεται που ενημερώνει το χρήστη πως δεν υπάρχει κάποια προσφορά.

Τα δεδομένα αποθηκεύονται στη Cloud Firestore και στη Firebase Realtime Database (γίνεται χρήση και των δύο επιλογών)



The image shows a mobile application screen titled "Create Business account". The form includes the following fields and elements:

- A red button labeled "SELECT LOCATION" at the top.
- A camera icon in a teal circle for profile picture selection.
- An "Email" input field.
- A "Business Type" dropdown menu with "Cafe-Bar" selected.
- A "Business Name" input field.
- A "Business Description" input field.
- A "Business Address" input field.
- A "Phone Number" input field.
- A "Password" input field with a visibility toggle icon.
- A "Confirm Password" input field with a visibility toggle icon.
- A large blue button labeled "Register" at the bottom.

Εικόνα register business ui

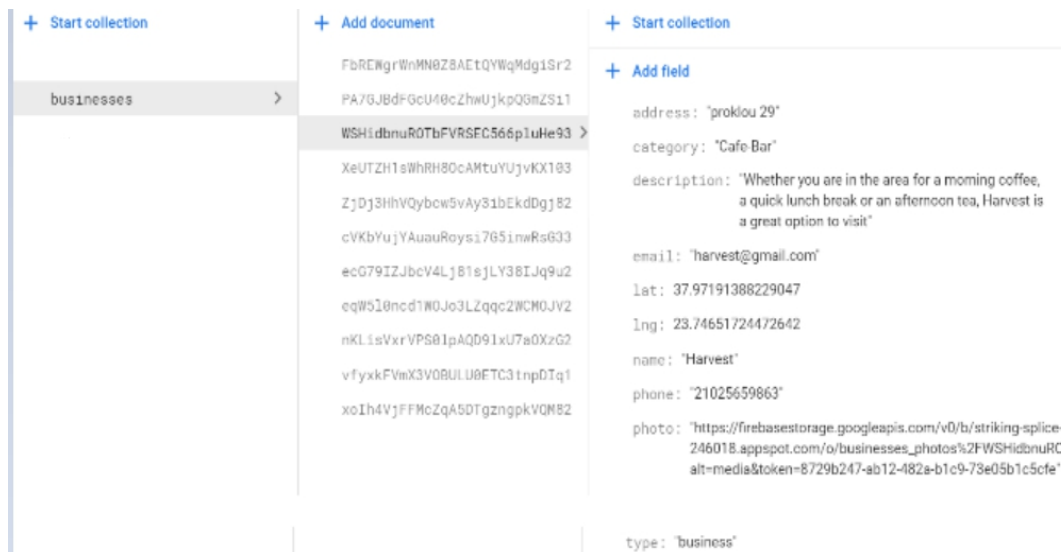
Παρακάτω φαίνεται η απεικόνιση των δεδομένων στη firestore database
 Η φωτογραφία της επιχείρησης αποθηκεύεται στο cloud storage της firebase.

```

FirebaseUser currentUser = mAuth.getCurrentUser();
String uid = FirebaseAuth.getInstance().getCurrentUser().getUid();
StorageReference storage = FirebaseStorage.getInstance().getReference().child("businesses_photos")
    .child(FirebaseAuth.getInstance().getCurrentUser().getUid());

storage.putFile(uri).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {

```



Προσθήκη προσφοράς

Μετά την επιτυχή εγγραφή, μεταβαίνει στο AdminActivity
 Εκεί ο business χρήστης έχει τη δυνατότητα να εισαγάγει νέα προσφορά επιλέγοντας το εικονίδιο το οποίο οδηγεί στο activity AddOffer

```

//on clicking to adding button
OffersFloatingActionButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(getActivity(), AddOffer.class));
    }
});

```

Ο χρήστης μπορεί να καταχωρήσει μία προσφορά συμπληρώνοντας τα στοιχεία

- όνομα προσφοράς
- περιγραφή προσφοράς
- κατηγορία προσφοράς
- from date time
- to date time
- όπως απεικονίζεται στο παρακάτω παράδειγμα

The screenshot shows the 'Add Offer' form with the following details:

- Offer Name:** Brunch
- Offer Description:** All options of Brunch for 5 euro
- Offer Category:** Brunch (dropdown menu is open showing: Drinks, Coffee, Beers, Brunch, Lunch, Coctails)
- Start Date/Time:** Tue, Jul 06 2021, 11:10 AM
- End Date/Time:** Sat, Jul 10 2021, 9:10 PM
- Image:** Happy HOUR logo
- Action:** ADD button

Εγγραφές στη βάση δεδομένων:

businesses	2zb6Iuvtc3fphIhbHA1y >	category: "Drinks"
offers	P3gALM0zVqn4dbj98KkH	description: "coctails 1+1"
	QXQ4M4ggYgI6I1PzbYIm	email: "harvest@gmail.com"
users	RBkhnDYF16X0nok7nRJ0	endTime: "Sun, Jul 11 2021 (12:24 PM)"
	T1q8QzDP03nKr4CFYjmM	idd: "2zb6Iuvtc3fphIhbHA1y"
	a8YdA3bDTgMXXKuRHIZN	lat: 37.97191388229047
	cJu9p1CqUv5UWBCqT1h9	lng: 23.74651724472642
	gv5BwuqD9ps6TpxnGY5N	owneridd: "WSHidbnuROtbFVRSEC566pluHe93"
	mZkS636bxuY0qgbvsEmd	ownername: "Harvest"
	qJihey0vTdFo09nRVSeQ	posted: 1625595862444
	vrtYU7NZRYIuHXhKUvq3	sortDate: 20210711
	VFTYU/NZKYIUMAKNUVq...	startTime: "Tue, Jul 06 2021 (12:24 PM)"
		title: "coctails"

Κώδικας που αποθηκεύει τις προσφορές

```
CollectionReference firebaseFirestore = FirebaseFirestore.getInstance().collection("offers");
String idd = firebaseFirestore.document().getId();
//enrich object
happyHour.setIdd(idd);
happyHour.setEmail(email);
happyHour.setOwnername(FirebaseAuth.getInstance().getCurrentUser().getDisplayName());
happyHour.setLat(lat);
happyHour.setSortDate(datetoday);
happyHour.setLng(lng);
happyHour.setPosted(System.currentTimeMillis());
happyHour.setOwneridd(FirebaseAuth.getInstance().getCurrentUser().getUid());
Snackbar.make(relativeLayout, text: "Adding Offer, please wait", Snackbar.LENGTH_INDEFINITE).show();
add.setVisibility(View.GONE);

firebaseFirestore.document(idd).set(happyHour).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if (task.isSuccessful()) {
            FirebaseDatabase.getInstance().getReference().child("offers").child(idd).setValue(happyHour, (listener, error) -> {
                Snackbar.make(relativeLayout, text: "Added successfully", Snackbar.LENGTH_LONG).show();
                sendSmsToSubscribers(happyHour);
            });
        }
    }
});
```

Ενημέρωση εγγεγραμμένων χρηστών κατά την προσθήκη νέας προσφοράς

Κάθε φορά που ο χρήστης-επιχειρηματίας προσθέτει μία προσφορά, εάν υπάρχουν εγγεγραμμένοι / ενδιαφερόμενοι πελάτες ενημερώνονται μέσω sms.

Το πως εγγράφονται οι πελάτες αναλύεται παρακάτω στο κεφάλαιο "Εκδήλωση ενδιαφέροντος για τις προσφορές μιας επιχείρησης".

```

public void sendSmsToSubscribers(HappyHour happyHour) {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.SEND_SMS) == PackageManager.PERMISSION_GRANTED) {
        FirebaseFirestore.getInstance().collection("businesses").document(FirebaseAuth.getInstance()
            .getCurrentUser().getUid()).collection("subscribers").get()
            .addOnCompleteListener()

            task -> {
                if (task.isSuccessful()) {
                    List<String> users = new ArrayList<>();
                    for (QueryDocumentSnapshot document : task.getResult()) {
                        Map<String, Object> data = document.getData();
                        String user = (String) data.get("user");
                        users.add(user);
                    }

                    FirebaseFirestore.getInstance().collection("users").get()
                        .addOnCompleteListener()
                            task2 -> {
                                if (task2.isSuccessful()) {
                                    for (QueryDocumentSnapshot document2 : task2.getResult()) {
                                        if (users.contains(document2.getId())) {
                                            Map<String, Object> data2 = document2.getData();
                                            String phone = (String) data2.get("phone");
                                            if (!phone.startsWith("+30")) {
                                                phone = "+30".concat(phone);
                                            }
                                            sendSMS(phone, "business " + happyHour.getOwnername()
                                                + " added new offer for " + happyHour.getTitle() +
                                                " description is: " + happyHour.getDescription()
                                                + "Please check your list for details"
                                            );
                                        }
                                    }
                                }
                            }
                }
            }
};
}

```

Λίστα προσφορών

Το admin activity περιέχει το OffersFragment `extends` Fragment , και μπορούμε να δούμε τις προσφορές της επιχείρησης.

```

adapter = new AdminOfferAdapter(getActivity(), OfferList);
OffersRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
OffersRecyclerView.setAdapter(adapter);
FirebaseRecyclerOptions<HappyHour> options =
    new FirebaseRecyclerOptions.Builder<HappyHour>()
        .setQuery(query, HappyHour.class)
        .build();
query.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if (!snapshot.exists()) {
            Toast.makeText(getActivity(), "You have no offer history", Toast.LENGTH_LONG).show();
        }
        OfferList.clear();
        for (DataSnapshot offerSnapshot : snapshot.getChildren()) {
            HappyHour happyHourOffer = offerSnapshot.getValue(HappyHour.class);

            AdminOffer adminOffer = new AdminOffer(happyHourOffer.getTitle(),
                happyHourOffer.getDescription());
            adminOffer.setStartTime(happyHourOffer.getStartTime());
            adminOffer.setEndTime(happyHourOffer.getEndTime());
            adminOffer.setCategory(happyHourOffer.getCategory());
            adminOffer.setIdd(happyHourOffer.getIdd());

            OfferList.add(adminOffer);
        }
        adapter.notifyDataSetChanged();
        bar.setVisibility(View.INVISIBLE);
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Toast.makeText(getActivity(), error.getMessage(), Toast.LENGTH_LONG).show();
        bar.setVisibility(View.INVISIBLE);
    }
});

```

Εγγραφή(register) για user type = customer (πελάτης)

Με την επιλογή κατηγορίας customer δημιουργείται ένα object Intent, το οποίο είναι υπεύθυνο για τη μετάβαση της εφαρμογής στο επόμενο activity (RegisterActivity).

Απαραίτητα στοιχεία για την εγγραφή αποτελούν τα:

- διεύθυνση ηλεκτρονικού ταχυδρομείου
- όνομα
- τηλέφωνο
- κωδικός και επιβεβαίωση αυτού

Android Permissions

Ο σκοπός μιας άδειας είναι να προστατεύσει το απόρρητο ενός χρήστη Android. Οι εφαρμογές Android πρέπει να ζητούν άδεια πρόσβασης σε ευαίσθητα δεδομένα χρήστη (όπως τοποθεσία και SMS), ανάλογα με τη λειτουργία, το σύστημα μπορεί να χορηγήσει αυτόματα την άδεια ή ενδέχεται να ζητήσει από το χρήστη να εγκρίνει το αίτημα

Για την εφαρμογή απαιτούνται οι παρακάτω άδειες οι οποίες δηλώνονται στο αρχείο AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
```

Εύρεση προσφορών κοντά στο χρήστη

Υπηρεσίες τοποθεσίας

Το Android λειτουργικό προσφέρει στις εφαρμογές πρόσβαση στις location services που υποστηρίζονται από την συσκευή μέσω των κλάσεων του πακέτου android.location. Κεντρικό τμήμα του location framework είναι η κλάση υπηρεσίας συστήματος LocationManager, που παρέχει APIs για τον προσδιορισμό της θέσης και της κατεύθυνσης της υποκείμενης συσκευής. Όπως και με τις υπόλοιπες υπηρεσίες συστήματος του Android, δεν δίνεται άμεση πρόσβαση στην LocationManager. Αντίθετα ο χρήστης αιτείται ένα στιγμιότυπο από το σύστημα καλώντας την

`getSystemService(Context.LOCATION_SERVICE)` .Η μέθοδος αυτή επιστρέφει μία σύνδεση σε ένα νέο στιγμιότυπο της `LocationManager` απο όπου η εφαρμογή μπορεί να βρει την τρέχουσα τοποθεσία του χρήστη.

Η συγκεκριμένη κλάση απαιτεί είτε την άδεια `ACCESS_COARSE_LOCATION` ή την `ACCESS_FINE_LOCATION` για να είναι δυνατή η λήψη της τοποθεσίας της συσκευής.

Η εφαρμογή ελέγχει εάν ο χρήστης βρίσκεται σε κοντινή απόσταση (< 3000 μέτρων) με κάποια επιχείρηση που να έχει ενεργή προσφορά.

Εάν βρεθεί προσφορά σε κοντινή απόσταση εγγράφεται στη βάση στα `notifications` και ο χρήστης έχει τη δυνατότητα ανά πάσα στιγμή να δει όλα τα `notifications` που τον αφορούν.

Για τον υπολογισμό της απόστασης μεταξύ του χρήστη και των καταστημάτων θα πρέπει να πραγματοποιηθούν τα εξής βήματα. Αρχικά να γίνει λήψη της λίστας των προσφορών και της τοποθεσίας των επιχειρήσεων από την `firebase` και στην συνέχεια να γίνει σύγκριση με την τρέχουσα τοποθεσία του χρήστη.

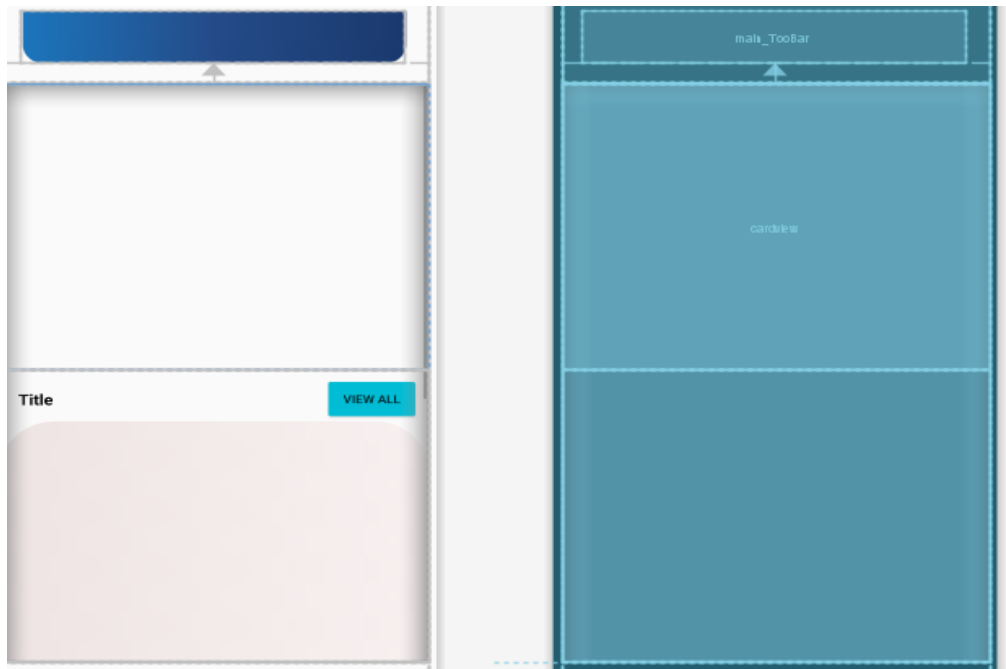
```
public void setLocationListener() {
    locationListener = new LocationListener() {
        @Override
        public void onLocationChanged(Location location) {
            for (HappyHour offer : offerList) {
                Location offerLocation = new Location("");
                offerLocation.setLatitude(offer.getLat());
                offerLocation.setLongitude(offer.getLng());
                float distanceInMeters = location.distanceTo(offerLocation);
                if (distanceInMeters < DISTANCE_NOTIFIED) {
                    DatabaseReference reference = FirebaseDatabase.getInstance().getReference().child("notifications")
                        .child(FirebaseAuth.getInstance().getCurrentUser().getUid());
                    String owner_id = offer.getId();
                    Map map = new HashMap();
                    map.put("businessName", offer.getOwnername());
                    map.put("offerName", offer.getTitle());
                    map.put("offerDescription", offer.getDescription());
                    map.put("lat", offer.getLat());
                    map.put("lng", offer.getLng());
                    map.put("fromTo", offer.getStartTime() + "-" + offer.getEndTime());
                    map.put("notified", new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss").format(new Date()));
                    reference.child(owner_id).updateChildren(map).addOnSuccessListener(new OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {
                            //inform the user if the notification is a new one
                            if (!notificationOffersList.contains(owner_id)) {
                                Toast.makeText(MainActivity.this, "new offer near to you!", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
                }
            }
        }
    };
}
```

Αρχική σελίδα για χρήστη = πελάτης

Στο `MainActivity` υπάρχει ένα `cardView` όπου εμφανίζονται όλες οι ενεργές προσφορές. (`MyOfferAdapter`)

και δύο `GridView`, όπου εμφανίζονται οι επιχειρήσεις των δύο κατηγοριών (`cafe-bar`, `restaurant`)

Παρακάτω φαίνεται η δομή της κύριας σελίδας του πελάτη:



```
xml version="1.0" encoding="utf-8"
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

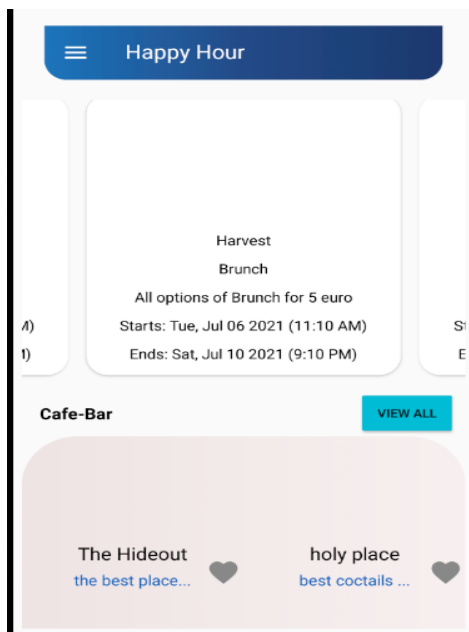
    <TextView
        android:id="@+id/grid_business_layout_textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:text="Title"
        android:textColor="#000"
        android:textSize="16sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/grid_button_layout_viewall_button"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="@+id/grid_button_layout_viewall_button" />

    <Button
        android:id="@+id/grid_button_layout_viewall_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:backgroundTint="@color/colorAccent"
        android:text="View all"
        android:textSize="12sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

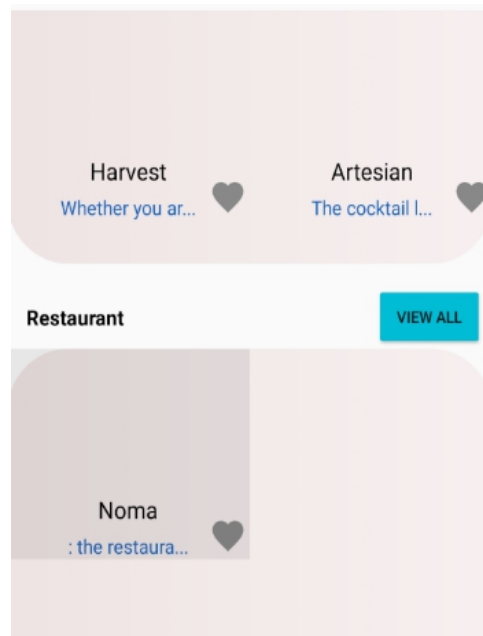
    <GridView
        android:id="@+id/business_layout_gridview"
        android:layout_width="match_parent"
        android:layout_height="350dp"
        android:layout_marginBottom="8dp"
        android:background="@drawable/square_shape"
        android:horizontalSpacing="1dp"
        android:numColumns="2"
        android:verticalSpacing="1dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/grid_button_layout_viewall_button" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Η κύρια σελίδα του πελάτη εμφανίζεται δίνοντας τη δυνατότητα στο χρήστη να δει όλες τις ενεργές προσφορές καθώς επίσης και κάποια καταστήματα ανά κατηγορία:

Μεταπτυχιακή Διατριβή

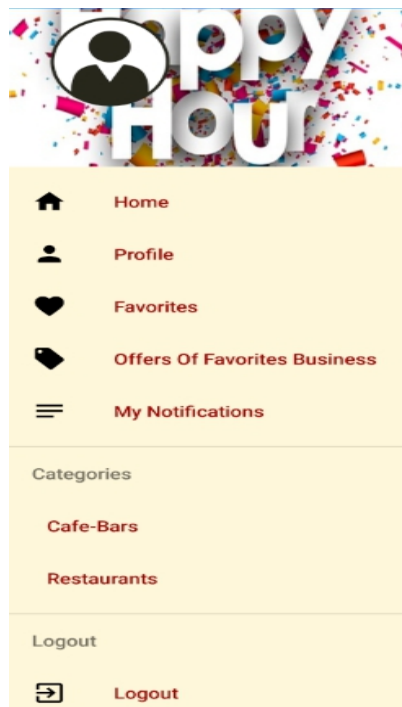
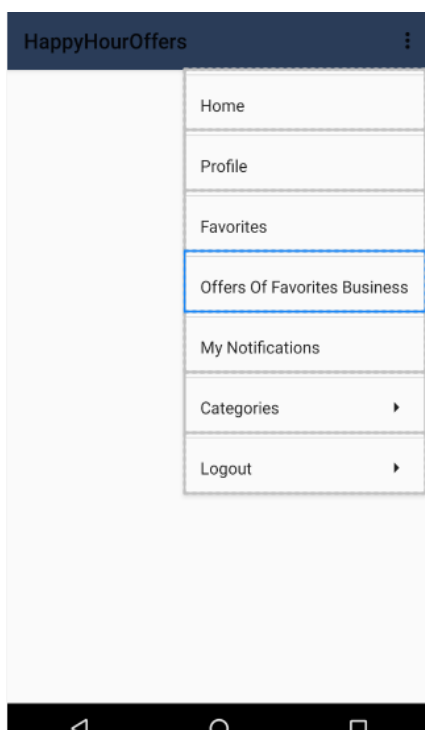


Ζωγράφου Μαρία-Αρετή



Κύριο menu

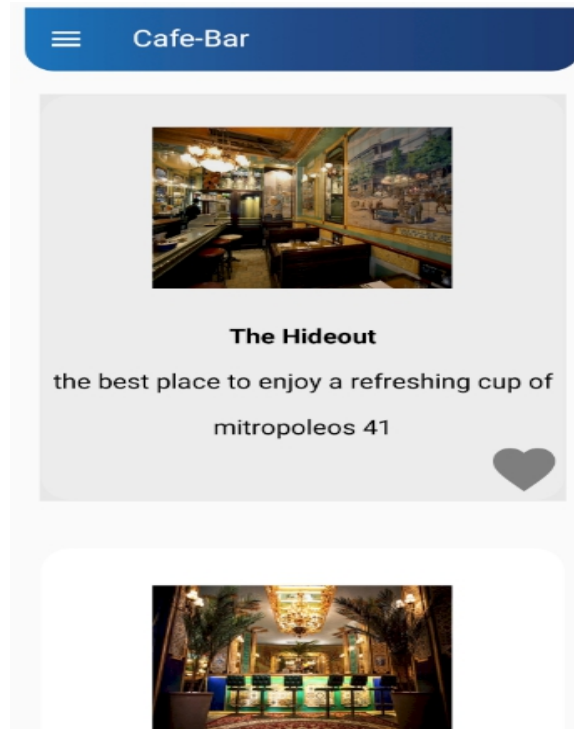
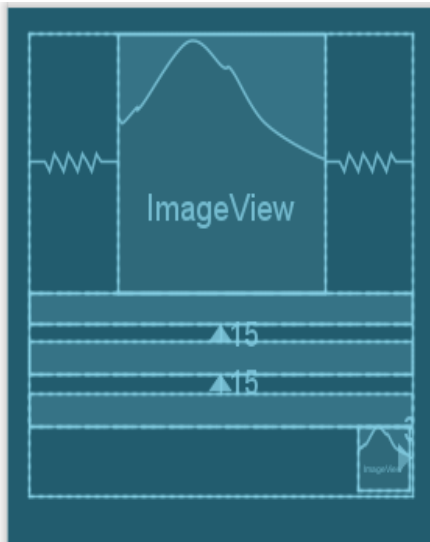
Σε όλες τις οθόνες υπάρχει η main toolbar που εμπεριέχει το main_menu.xml



Λίστα επιχειρήσεων ανά κατηγορία

Με την επιλογή του view all στην κατηγορία cafe-bar ο χρήστης μεταβαίνει στο CategoryActivity όπου υπάρχει ένα RecyclerView , και μπορούμε να δούμε τις επιχειρήσεις αυτής της κατηγορίας (category_business_list.xml)

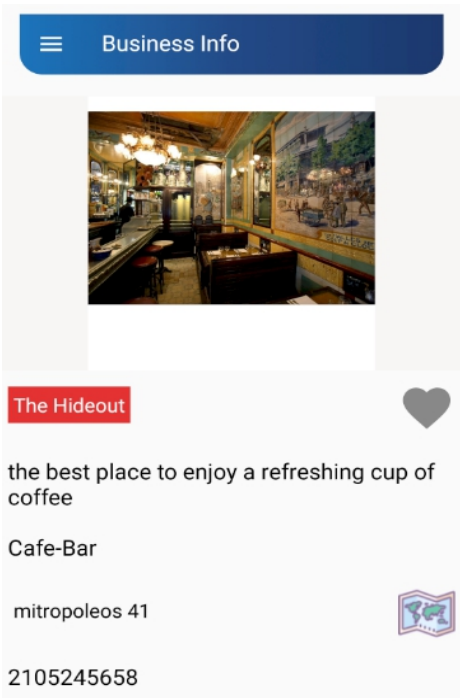
Πατώντας σε κάθε εγγραφή στη λίστα, ο χρήστης μεταβεί στο προφίλ της εκάστοτε επιχείρησης



Προφίλ επιχείρησης

Το BusinessInfoActivity παρέχει τις πληροφορίες για την επιχείρηση.

Ο χρήστης έχει τη δυνατότητα να εγγραφεί, δηλώνοντας ενδιαφέρον για να παρακολουθεί τις προσφορές της συγκεκριμένης επιχείρησής. Πατώντας στο χάρτη μπορεί να δει την ακριβή τοποθεσία της επιχείρησης, καθώς και την τρέχουσα τοποθεσία του ίδιου.



```
private void setBusinessData() {
    DocumentReference docRef = FirebaseFirestore.getInstance().collection("businesses").document("businessId");
    docRef.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DocumentSnapshot> task) {
            if (task.isSuccessful()) {
                DocumentSnapshot document = task.getResult();
                if (document.exists()) {
                    Map<String, Object> data = document.getData();
                    businessName = (String) data.get("name");
                    businessDescription = (String) data.get("description");
                    businessImage = (String) data.get("photo");
                    if (businessImage != null) {
                        Glide.with(getBaseContext()).load(businessImage).into(image);
                    }

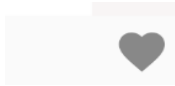
                    businessAddress = (String) data.get("address");
                    BPhone.setText((String) data.get("phone"));
                    BAddress.setText(businessAddress);
                    BCategory.setText((String) data.get("category"));

                    BName.setText(businessName);
                    BDescription.setText(businessDescription);
                    businessMap.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                            Intent intent = new Intent(BusinessInfoActivity.this, MapMarker.class);

                            intent.putExtra("lat", (Double) data.get("lat"));
                            intent.putExtra("lng", (Double) data.get("lng"));
                            intent.putExtra("name", businessName);
                            startActivity(intent);
                        }
                    });
                } else {
                    Toast.makeText(BusinessInfoActivity.this, "Could not retrieve data for business " + businessId, Toast.LENGTH_LONG).show();
                }
            } else {
                Toast.makeText(BusinessInfoActivity.this, task.getException().toString(), Toast.LENGTH_LONG).show();
            }
        }
    });
}
```

Εκδήλωση ενδιαφέροντος για τις προσφορές μιας επιχείρησης

Ο χρήστης έχει τη δυνατότητα να δηλώσει ενδιαφέρον για μια ή περισσότερες επιχειρήσεις. Αυτό του δίνει τη δυνατότητα να ενημερώνεται μέσω sms κάθε φορά που προστίθεται μία νέα προσφορά στην εκάστοτε επιχείρηση και να βλέπει σε λίστα όλες τις ενεργές προσφορές των επιχειρήσεων που έχει εγγραφεί.



Για να δηλώσει ενδιαφέρον μπορεί να πατήσει σε οποιαδήποτε διεπαφή εμφανίζεται το εικονίδιο: Όταν ο χρήστης έχει δηλώσει ενδιαφέρον το εικονίδιο εμφανίζεται ως:



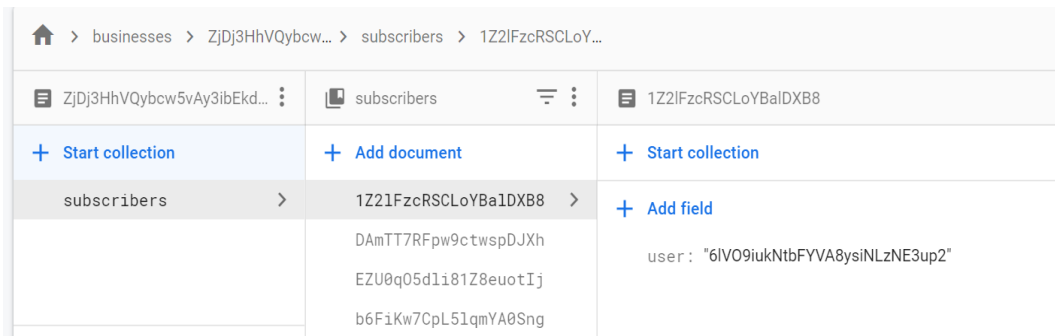
Η εγγραφή του χρήστη στην επιχείρηση αποθηκεύεται στη firebase στα "subscriptions" ανά χρήστη, καθώς επίσης και ως collection στη firebaseFirestore του collection "businesses"

```
public static void addSubscriptions(String businessName , String businessAddress , String owner_id){
    Map map = new HashMap();
    map.put( k: "businessName", businessName);
    map.put( k: "businessAddress", businessAddress);
    FirebaseDatabase.getInstance().getReference().child("subscriptions")
        .child(FirebaseAuth.getInstance().getCurrentUser().getUid()).child(owner_id).updateChildren(map);
    Map map1=new HashMap();
    map1.put( k: "user",FirebaseAuth.getInstance().getCurrentUser().getUid());
    FirebaseFirestore.getInstance().collection( collectionPath: "businesses").document(owner_id).collection( collectionPath: "subscribers")
        .add(map1);
}
```

Στη διαδικασία του unsubscribe εκτελείται η διαγραφή κατά την ίδια λογική:

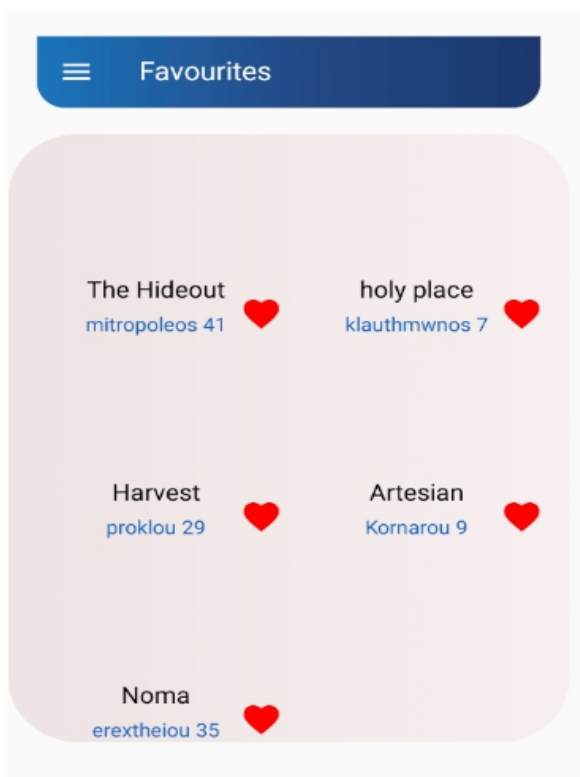
```
public static void removeSubscriptions(String owner_id){
    FirebaseDatabase.getInstance().getReference().child("subscriptions")
        .child(FirebaseAuth.getInstance().getCurrentUser().getUid())
        .child(owner_id).removeValue();

    FirebaseFirestore.getInstance().collection( collectionPath: "businesses").document(owner_id)
        .collection( collectionPath: "subscribers").get() .addOnCompleteListener(
            task -> {
                if (task.isSuccessful()) {
                    for (QueryDocumentSnapshot document : task.getResult()) {
                        Map<String, Object> data = document.getData();
                        String user = (String) data.get("user");
                        if (user.equals(FirebaseAuth.getInstance().getCurrentUser().getUid())) {
                            document.getReference().delete();
                        }
                    }
                }
            });
}
```

Λίστα αγαπημένων επιχειρήσεων

Επιλέγοντας Favorites από το μενού επιλογών , ο χρήστης έχει τη δυνατότητα να δει όλες τις επιχειρήσεις που έχει εγγραφεί.



```
public void Retrieve_fav() {
    LinearLayout myLayout = (LinearLayout) findViewById(R.id.recyclerViewLayout);
    LayoutInflater inflater = getLayoutInflater();
    inflater.inflate(R.layout.favourite_recycler_view, myLayout, attachToRoot: false);
    final RecyclerView rc = myLayout.findViewById(R.id.recyclerView);
    GridLayoutManager mGridLayoutManager;
    mGridLayoutManager = new GridLayoutManager(context: this, spanCount: 2, GridLayoutManager.VERTICAL, reverseLayout: false);
    rc.setLayoutManager(mGridLayoutManager);
    final List<FavouritesClass> favourite_list = new ArrayList<>();
    my_adapter = new MyAdapter_Recycler_View(favourite_list);
    rc.setAdapter(my_adapter);
    FirebaseDatabase.getInstance().getReference().child("subscriptions").child(FirebaseAuth.getInstance().getCurrentUser().getUid())
        .addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                for (DataSnapshot singleSnapshot : dataSnapshot.getChildren()) {
                    String businessName = singleSnapshot.child("businessName").getValue(String.class);
                    String businessAddress = singleSnapshot.child("businessAddress").getValue(String.class);
                    FavouritesClass fav = new FavouritesClass(businessName, businessAddress, checked: true);
                    fav = singleSnapshot.getValue(FavouritesClass.class);
                    favourite_list.add(fav);
                }
            }
            my_adapter.notifyDataSetChanged();
        })
}
```

```

public class MyAdapter_Recycler_View extends RecyclerView.Adapter<MyAdapter_Recycler_View.ViewHolder> {
    private List<FavouritesClass> horizontalBusinessModelList;

    public MyAdapter_Recycler_View(List<FavouritesClass> horizontalBusinessModelList) {
        this.horizontalBusinessModelList = horizontalBusinessModelList;
    }

    @NonNull
    @Override
    public MyAdapter_Recycler_View.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.horizontal_item, null);
        return new ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(@NonNull MyAdapter_Recycler_View.ViewHolder holder, int position) {
        FavouritesClass horizontalBusinessModel = horizontalBusinessModelList.get(position);

        holder.businessName.setText(horizontalBusinessModel.getBusinessName());
        holder.businessAddress.setText(horizontalBusinessModel.getBusinessAddress());
        holder.checkBox.setImageResource(R.drawable.ic_baseline_favorite_24);
    }

    @Override
    public int getItemCount() {
        return horizontalBusinessModelList.size();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {

        ImageView businessImage;
        TextView businessName, businessAddress;
        ImageView checkBox;

        public ViewHolder(@NonNull View view) {
            super(view);
            businessImage = view.findViewById(R.id.item_image);
            businessName = view.findViewById(R.id.item_title);
            businessAddress = view.findViewById(R.id.item_description);

            checkBox = view.findViewById(R.id.check_box);
        }
    }
}

```

Διεπαφή ενεργών προσφορών των επιχειρήσεων που έχει εκδηλώσει ενδιαφέρον

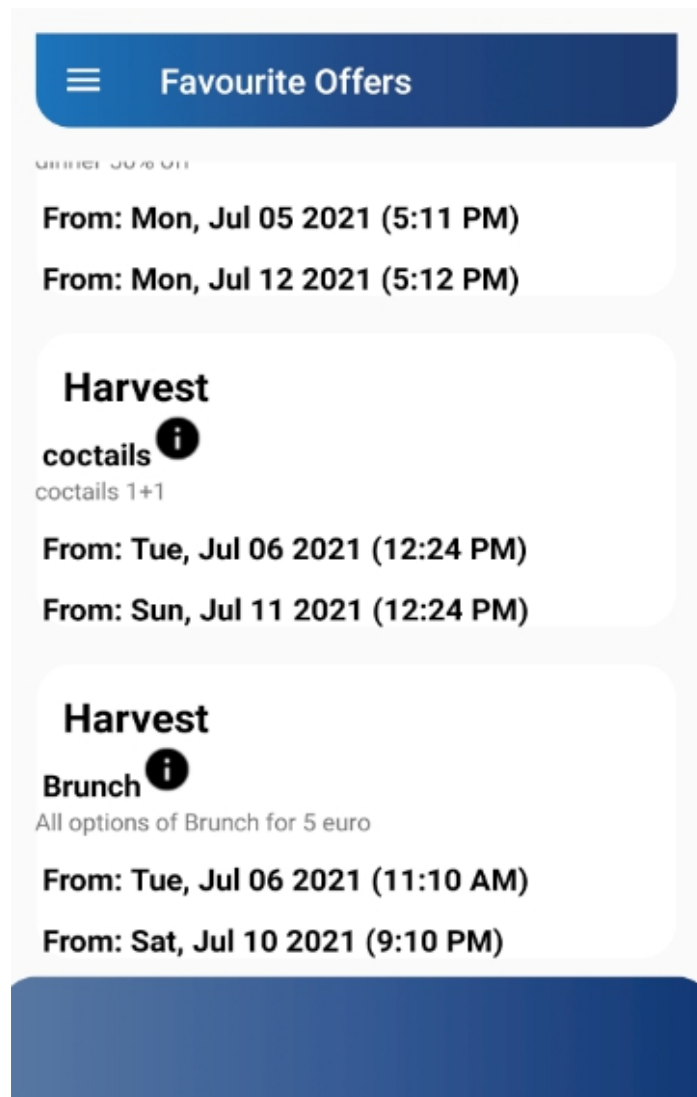
Επιλέγοντας “Offers of Favorites Business” ο χρήστης μπορεί να δει όλες τις ενεργές προσφορές των επιχειρήσεων που έχει εγγραφεί - δηλώσει ενδιαφέρον (FavoritesOffersActivity). Τα αποτελέσματα εμφανίζονται σε ένα RecyclerView ενός Fragment.

```

<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:background="@drawable/business_container">
    <TextView
        android:id="@+id/business_title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="16dp"
        android:text="business name"
        android:textColor="#000"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/offer_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="4dp"
        android:layout_marginTop="12dp"
        android:text=""
        android:textColor="#000000"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/business_title" />
    <ImageView
        android:id="@+id/businessDetails"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_alignParentRight="true"
        app:layout_constraintLeft_toRightOf="@+id/offer_name"
        app:srcCompat="@drawable/business_information"
        app:layout_constraintTop_toBottomOf="@+id/business_title"
        tools:ignore="MissingConstraints" />
    <TextView
        android:id="@+id/offer_details"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginRight="8dp"
        android:text=""
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/offer_name" />
    <TextView
        android:id="@+id/offer_from"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="4dp"
        android:layout_marginTop="12dp"
        android:text=""
        android:textColor="#000000"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/offer_details" />
    <TextView
        android:id="@+id/offer_to"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="4dp"
        android:layout_marginTop="12dp"
        android:text=""
        android:textColor="#000000"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/offer_from" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Η διεπαφή εμφανίζεται ως εξής:



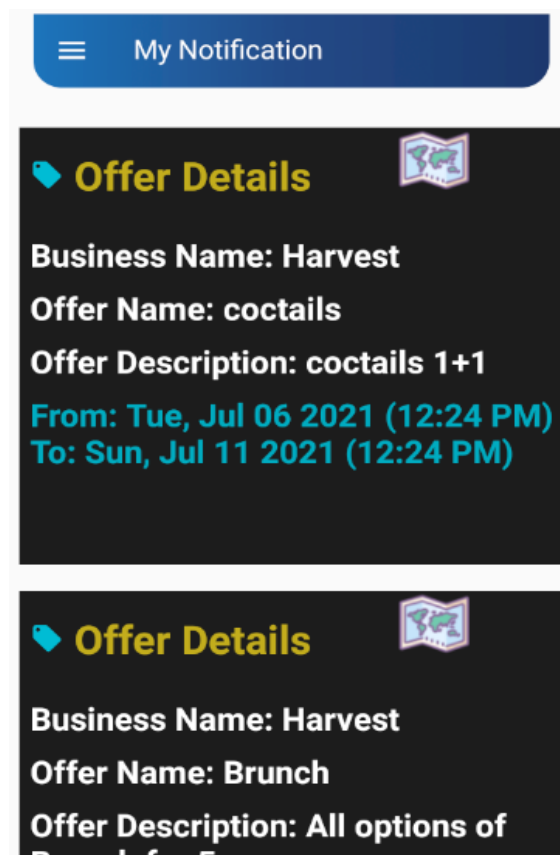
Εάν ο χρήστης επιλέξει το ενημερωτικό εικονίδιο μεταβαίνει στην οθόνη με τις πληροφορίες της επιχείρησης.

```
businessDetails.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(context, BusinessInfoActivity.class);
        intent.putExtra( name: "id", happyHourList.get(position).getOwnerid());
        context.startActivity(intent);
    }
});
```

Διεπαφή ειδοποιήσεων (notifications) για εύρεση σε κοντινή απόσταση καταστημάτων με προσφορές

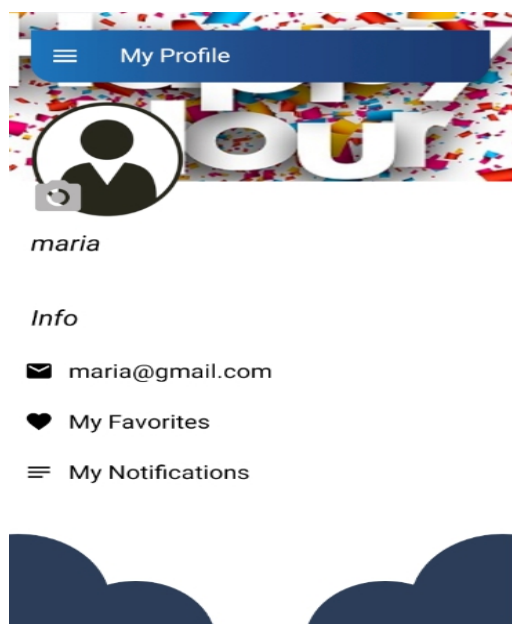
Δίνεται η δυνατότητα στο χρήστη επιλέγοντας “My Notifications” από το μενού επιλογών να δει όλα τα notification για προσφορές που έχει βρεθεί σε κοντινή απόσταση (<3000 μέτρων) σε σχέση με την τοποθεσία της εκάστοτε επιχείρησης.

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View view = inflater.inflate(R.layout.fragment_notification, container, false);
    mAuth = FirebaseAuth.getInstance();
    currentUser = mAuth.getCurrentUser().getUid();
    NotificationItemRecyclerView = view.findViewById(R.id.notificationrecycler);
    notificationItemList = new ArrayList<MyNotificationModel>();
    adapter = new NotificationAdapter(getActivity(), notificationItemList);
    NotificationItemRecyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
    NotificationItemRecyclerView.setAdapter(adapter);
    String userId = FirebaseAuth.getInstance().getCurrentUser().getUid();
    FirebaseDatabase.getInstance().getReference().child("notifications").child(userId)
        .addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                for (DataSnapshot singleSnapshot : dataSnapshot.getChildren()) {
                    String businessName = singleSnapshot.child("businessName").getValue(String.class);
                    String offerDescription = singleSnapshot.child("offerDescription").getValue(String.class);
                    String offerName = singleSnapshot.child("offerName").getValue(String.class);
                    String fromTo = singleSnapshot.child("fromTo").getValue(String.class);
                    String notified = singleSnapshot.child("notified").getValue(String.class);
                    Double lat = singleSnapshot.child("lat").getValue(Double.class);
                    Double lng = singleSnapshot.child("lng").getValue(Double.class);
                    MyNotificationModel not = new MyNotificationModel("From: " + fromTo.replace("-", "\nTo: "), "Business Name: " + businessName,
                        "Offer Name: " + offerName, "Offer Description: " + offerDescription, notified, lat, lng);
                    notificationItemList.add(not);
                }
                adapter.notifyDataSetChanged();
            }
            @Override
            public void onCancelled(DatabaseError databaseError) {
            }
        });
    return view;
}
```



Προβολή προφίλ χρήστη

Τέλος ο χρήστης έχει τη δυνατότητα να δει τα στοιχεία του προφίλ του , με συνδέσμους στις οθόνες My Favorites και My Notifications



Συμπεράσματα και μελλοντικές επεκτάσεις

Το παρόν έγγραφο είχε ως κύριο στόχο την παρουσίαση της android εφαρμογής που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής διατριβής. Επιπλέον, κρίθηκε σκόπιμο να περιγραφούν ,κάποια βασικά στοιχεία για το android ,καθώς επίσης και εργαλεία που κάνουν δυνατή την ανάπτυξη εφαρμογών για android συσκευές.

Στη παρούσα μεταπτυχιακή εργασία σχεδιάστηκε και υλοποιήθηκε μια Android εφαρμογή με τη για έξυπνα τηλέφωνα που αφορά την εισαγωγή και εύρεση καθημερινών προσφορών των επιχειρήσεων εστίασης.

Παράλληλα, είναι εύχρηστη και λειτουργική, καθώς δεν περιέχει πολύπλοκες λειτουργίες .

Ο χρήστης δεν περιορίζεται στην χρήση της εφαρμογής σε μία μόνο συσκευή, αλλά μπορεί να έχει πρόσβαση και σε οποιαδήποτε άλλη, εφόσον θυμάται το email και τον κωδικό πρόσβασης

Από την άλλη μεριά,για να λειτουργήσει η συγκεκριμένη υλοποίηση , είναι απαραίτητη η πρόσβαση στο διαδίκτυο, καθώς τα δεδομένα ανακτώνται τη χρονική στιγμή που ο χρήστης ανοίγει την εφαρμογή. Σε αντίθετη περίπτωση, οι εγγραφές δεν εμφανίζονται και η εφαρμογή δεν μπορεί πρακτικά να χρησιμοποιηθεί .

Στην εποχή μας που ανάπτυξη των έξυπνων συσκευών και των υπηρεσιών που προσφέρουν είναι ραγδαία και συνεχής κάθε εφαρμογή πρέπει να αναβαθμίζεται και να βελτιώνεται.

Έτσι και οι δυνατότητες της εφαρμογής που υλοποιήσαμε μπορούν να επεκταθούν μελλοντικά προς αρκετές κατευθύνσεις.

Μερικές ιδέες για μελλοντική επέκταση της εφαρμογής παρουσιάζονται παρακάτω:

- Βελτιστοποίηση των διεπαφών ως προς το γραφικό περιβάλλον(UI) και την εμπειρία χρήσης(UX)

- Δυνατότητα αναζήτησης προσφορών βάσει τύπου προσφοράς/ημερομηνίας και περιοχής
- Δυνατότητα στον πελάτη να εγγράφεται/δηλώνει ενδιαφέρον και σε κατηγορίες προσφορών,
- Αν κρίνεται απαραίτητο εισαγωγή push notifications.
- Προσθήκη ενός ρόλου admin όπου θα ορίζει τα emails των επιχειρήσεων που θα επιτρέπεται να εγγραφούν.

Βιβλιογραφία – Διαδικτυακές πηγές

Android. <https://www.android.com/>

Android Emulator. <https://developer.android.com/studio/run/emulator>

Android History and Versions. https://en.wikipedia.org/wiki/Android_version_history

Android Studio. https://en.wikipedia.org/wiki/Android_Studio

Android Operating System. [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

IOS Operating System <https://en.wikipedia.org/wiki/IOS>

Add Firebase. <https://firebase.google.com/docs/android/setup>

Cloud Firestore. <https://firebase.google.com/docs/firestore>

Firestore Query Data. <https://firebase.google.com/docs/firestore/query-data/listen>

Developer Android. <https://developer.android.com/docs>

Android layouts. <https://developer.android.com/guide/topics/ui/declaring-layout>

MaterialDateTimePicker. <https://github.com/wdullaer/MaterialDateTimePicker>

Glide. <https://github.com/bumptech/glide>

