



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού και Τεχνητής Νοημοσύνης»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη android εφαρμογής Street Art οδηγός πόλης Development of android application Street Art city guide
Όνοματεπώνυμο Φοιτητή	Κωνσταντίνος Χαλαστάνης
Πατρώνυμο	Δημήτριος
Αριθμός Μητρώου	ΜΠΣΠ 18027
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Νοέμβριος 2022**

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Βίβρου Μαρία
Καθηγήτρια

Κωνσταντίνος Πατσάκης
Αναπληρωτής Καθηγητής

Περίληψη

Έχει παρέλθει σχεδόν πάνω από μία δεκαετία που τα έξυπνα κινητά τηλέφωνα έχουν έρθει στη ζωή μας και η πραγματικότητα είναι ότι έχουν αλλάξει κατά πολύ την συμπεριφορά μας σε πολλούς τομείς θετικά αλλά και σε πολλούς αρνητικά. Είναι γεγονός δε πως ακούμε την φράση «η μετάβαση στη ψηφιακή εποχή» όλο και πιο συχνά, συνεχώς από περισσότερα μέσα και η πραγματικότητα είναι πως την τελευταία δεκαετία οι συνήθειές μας σταδιακά έχουν περάσει στον ψηφιακό κόσμο όπως το e-banking, ο τρόπος που κάνουμε τη φορολογική μας δήλωση, η επικοινωνία μας, το άλμπουμ των φωτογραφιών μας και πολλά άλλα.

Η εργασία που αναπτύσσεται στη συνέχεια ανήκει και αυτή σε αυτό το κόσμο της μετάβασης. Αφορά μια εφαρμογή που σκοπός της είναι να αναδείξει την τέχνη του δρόμου. Η δυνατότητα που θέλει να δώσει στο χρήστη είναι να μπορέσει να βρει σε μια λίστα συγκεντρωμένα τα «Street Art» έργα που είναι τοποθετημένα σε διάφορα σημεία της πόλης, να τον ειδοποιήσει για τυχόν κοντινά σε αυτόν με βάση τις ρυθμίσεις του αλλά και να τον βάλει στο παιχνίδι της αναζήτησης καθώς θα μπορέσει και ο ίδιος να προσθέσει ένα «Street Art» έργο εάν το βρει ή αν φυσικά είναι ο δημιουργός του.

Το προγραμματιστικό περιβάλλον που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής είναι το Android Studio και γράφτηκε σε γλώσσα προγραμματισμού Java με xml για την εμφάνιση. Για την υλοποίηση διάφορων λειτουργιών εκτός από τις πολλές δυνατότητες που προσφέρει το Android Studio, χρησιμοποιήθηκε η Firebase πλατφόρμα της Google για να καλύψει την ανάγκη χρήσης βάσης δεδομένων, χώρου αποθήκευσης και αυθεντικοποίησης χρήστη καθώς και άλλες βιβλιοθήκες που θα αναπτυχθούν στη συνέχεια του εγγράφου.

Εν τέλει η android εφαρμογή που θα αναπτυχθεί στη συνέχεια αυτής της εργασίας, θέλει να πετύχει αυτή την «μετάβαση» της εμπειρίας ενός ανθρώπου που θα αναζητούσε τα «έργα του δρόμου» γνωρίζοντας την οδό, ρωτώντας και ψάχνοντας μέσα στην πόλη, σε μια εμπειρία ενός ανοιχτού μουσείου που μπορεί να βρει τα έργα στους «διαδρόμους» της πόλης συμβουλευόμενος το κινητό του τηλέφωνο.

Abstract

It has been almost more than a decade since smart mobile phones came into our lives and the reality is that they have greatly changed our behavior in many areas positively and in many negatively. It is a fact that we hear the phrase "the transition to the digital age" more and more often, constantly from more media and the reality is that in the last decade our habits have gradually moved to the digital world such as e-banking, the way we do our tax return, our communication, our photo album and more.

The subject of the thesis that develops next also belongs to this world of transition. It's about an app that aims to highlight street art. The possibility that the app wants to give to the user is to be able to find in a list gathered the "Street Art" works that are placed in various parts of the city, to notify him of any close to him based on his settings and also to put him in the search game as he will also be able to add a "Street Art" work if he finds it or of course if he is the creator of it.

The development environment used to implement the application is Android Studio and it was written in Java programming language with xml for display. To implement various functions in addition to the many features offered by Android Studio, Google's Firebase platform was used to cover the need to use database, storage and user authentication as well as other libraries that will be developed later in the document.

Finally, the android application that will be developed in the continuation of this thesis, wants to achieve this "transition" of the experience of a person who would look for the "Street Art" by knowing the street, asking and searching in the city, to an experience of a of an open museum that can find the works in the "corridors" of the city by consulting his mobile phone.

Ευχαριστίες

Θέλω να ευχαριστήσω τον καθηγητή μου κύριο Ευθύμιο Αλέπη διότι δέχτηκε με χαρά την πρότασή μου να εκπονήσω την διπλωματική μου εργασία κάτω από τις οδηγίες του, αλλά κυρίως διότι κατά τη συνέντευξη μου πριν παρακολουθήσω το μεταπτυχιακό πρόγραμμα και παρόλο που οι προπτυχιακές μου σπουδές απέχουν σημαντικά, με εμπιστεύτηκε και δέχτηκε τη συμμετοχή μου.

Ακόμη θέλω να ευχαριστήσω την οικογένειά μου και τους φίλους που με στηρίζουν και πιστεύουν σε εμένα και τις επιλογές μου.

Νοέμβριος 2022

Χαλασάνης Κωνσταντίνος

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη.....	3
Abstract.....	3
Ευχαριστίες.....	4
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	5
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	6
1 Εισαγωγή.....	7
1.1 Περιγραφή του υπό μελέτη προβλήματος.....	7
1.2 Σκοπός και στόχοι της εργασίας.....	7
1.3 Βασικοί ορισμοί	7
1.3.1 Android.....	8
1.3.2 Android Studio	8
1.3.3 Java	8
1.3.4 Firebase Platform	9
1.3.5 Firebase Firestore	9
1.3.6 Firebase Cloud Storage	9
1.3.7 Firebase Auth.....	10
1.3.8 Firebase ML Kit.....	10
2 Σχεδίαση της εφαρμογής	11
2.1 Χρήση διαγραμμάτων UML.....	11
2.1.1 Use Case Diagrams.....	11
2.1.2 Sequence Diagrams	14
3 Υλοποίηση της εφαρμογής.....	20
3.1 Χρήση του Android Studio και αρχικές ρυθμίσεις	20
3.1.1 Android Manifest.....	22
3.1.2 Build Gradle.....	22
3.2 Βασική ιδέα και στάδια ανάπτυξης της εφαρμογής	23
4 Προσομοίωση της εφαρμογής	31
4.1 Εκκίνηση της εφαρμογής για πρώτη φορά	31
4.2 Είσοδος στην εφαρμογή ως admin	36
5 Μέθοδοι και μέρη του κώδικα.....	38
5.1 Εκκίνηση της εφαρμογής και permissions	38
5.2 Δημιουργία νέου χρήστη και σύνδεση με τη βάση δεδομένων.....	39
5.3 Υπολογισμός της απόστασης μεταξύ ενός Street Art έργου.....	43
5.4 Χρήση κλάσης Geocoder.....	43
5.5 Χρήση ML kit labeler	44
6 Συμπεράσματα και μελλοντικές επεκτάσεις.....	45
7 Βιβλιογραφικές Πηγές.....	46

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1. Use Case Enter as Guest: Ο χρήστης επιλέγει να εισέλθει ως επισκέπτης.	12
Εικόνα 2. Use Case Create New Account: Ο χρήστης επιλέγει να δημιουργήσει λογαριασμό για να εισέλθει στο σύστημα.	12
Εικόνα 3. Use Case Login: Ο χρήστης κάνει είσοδο στο σύστημα.	13
Εικόνα 4. Use Case Menu: Ο χρήστης έχει πραγματοποιήσει είσοδο και έχει πρόσβαση στο μενού.	14
Εικόνα 5. Sequence Diagram: Ο χρήστης εισέρχεται στο σύστημα ως επισκέπτης.	15
Εικόνα 6. Sequence Diagram: Ο χρήστης δημιουργεί λογαριασμό στο σύστημα.	16
Εικόνα 7. Sequence Diagram: Ο χρήστης κάνει Login στο σύστημα.	17
Εικόνα 8. Sequence Diagram: Ο χρήστης επιλέγει τα settings από το μενού.	18
Εικόνα 9. Sequence Diagram: Ο χρήστης επιλέγει τους χάρτες από το μενού.	18
Εικόνα 10. Sequence Diagram: Ο χρήστης επιλέγει να προσθέσει ένα Street Art.	19
Εικόνα 11. Android Studio: δημιουργία νέου project.	20
Εικόνα 12. Android Studio: Επιλογή ονόματος της εφαρμογής, γλώσσας ανάπτυξης, minimum SDK.	21
Εικόνα 13. Android Studio: Μέρος του AndroidManifest.xml με σχόλια και το link που θα ακολουθήσουμε για να ενεργοποιήσουμε το API KEY.	21
Εικόνα 14. Android Studio: build.gradle και firebase assistant.	22
Εικόνα 15. Android Studio: Το activity_main.xml ως αρχική σελίδα.	23
Εικόνα 16. Android Studio: Το activity_login.xml με τις επιλογές του χρήστη.	24
Εικόνα 17. Android Studio: Το activity_create_account.xml με τα πεδία που πρέπει να συμπληρωθούν από το χρήστη.	24
Εικόνα 18. Δήλωση των permission στο manifest.	25
Εικόνα 19. Android Studio: Το μοντέλο του button label που θα χρησιμοποιηθεί για την προβολή της λίστας μέσω του SettingsRecyclerViewAdapter.	26
Εικόνα 20. Android Studio: Το activity_settings.xml με χρήση Seekbar για την επιλογή της απόστασης και τη recycler view όπου θα προβληθούν τα labels buttons.	26
Εικόνα 21. Android Studio: Το menu.xml.	27
Εικόνα 22. Android Studio: Το activity_add_new.xml.	28
Εικόνα 23. Android Studio: Το activity_admin με δυο SeekBar.	28
Εικόνα 24. Android Studio: Το activity_about.xml.	29
Εικόνα 25. Android Studio: Το activity_street_art_list.xml. Περιέχει ένα recycler view.	29
Εικόνα 26. Android Studio: Το street_art_doc_view.xml. Το μοντέλο για την λίστα του StreetArtListActivity.java.	30
Εικόνα 27. Η εφαρμογή κατά την εκκίνηση θα ζητήσει δικαιώματα από το χρήστη.	31
Εικόνα 28. Ο χρήστης επιλέγει να εισέλθει ως guest.	32
Εικόνα 29. Εγγραφή νέου χρήστη και ρύθμιση των αρχικών του settings.	33
Εικόνα 30. Επιλογή ενός πλακιδίου από τη λίστα.	34
Εικόνα 31. Εισερχόμενες ειδοποιήσεις και προβολή αυτών.	34
Εικόνα 32. Ανάπτυξη notification και λήψη οδηγιών μόλις ο χρήστης κάνει κλικ πάνω του.	35
Εικόνα 33. Επιλογές μενού Map και Add New Street Art.	35
Εικόνα 34. Επιλογές Settings και About από το μενού.	36
Εικόνα 35. Μενού admin user και activity_admin.	37
Εικόνα 36. Μεθοδολογία για την απόδοση δικαιωμάτων στην εφαρμογή.	38
Εικόνα 37. Android Studio: Κώδικας της κλάσης StreetArtUser.	39
Εικόνα 38. Android Studio: Δηλώσεις στην CreateAccountActivity για την σύνδεση με τον authentication server, την Firestore και η αναφορά της collection.	40
Εικόνα 39. Android Studio: Κώδικας της CreateAccountActivity με τη διαδικασία δημιουργίας του user.	42
Εικόνα 40. Firebase Authentication.	42
Εικόνα 41. Firebase Cloud Firestore.	42
Εικόνα 42. Συναρτήσεις υπολογισμού μεταξύ δύο συντεταγμένων.	43
Εικόνα 43. Χρήση της κλάσης Geocoder.	44
Εικόνα 44. Χρήση του ImageLabeler από το interface ML kit.	44

1 Εισαγωγή

1.1 Περιγραφή του υπό μελέτη προβλήματος

Αναζητώντας την τέχνη του δρόμου περιπλανώμενος στα τετράγωνα της πόλης, σε γειτονίες και πλατείες, παγκάκια, μέσα μαζικής μεταφοράς και κυριολεκτικά σε μέρη που ούτε καν περιμένεις ότι μπορεί να υπάρχει μια παρέμβαση από ανθρώπινο χέρι, συχνά συναντάς μουντζούρες, συνθήματα είτε οπαδικά είτε κομματικά, tags κ.α. και εκεί που κοιτάξεις αφηρημένα ξάφνου εμφανίζονται μπροστά σου αξιοθαύμαστα έργα καλλιτεχνών του δρόμου. Όλα αυτά στο σύνολο τους έρχονται να διχάσουν πολλές φορές την κοινωνία καθώς πέρα από αυτούς που αδιαφορούν, πολλοί είναι αυτοί που τα αντιμετωπίζουν στο σύνολο τους αυστηρά ως βανδαλισμό, την ίδια στιγμή που άλλοι μπορούν να διακρίνουν την ανάγκη έκφρασης των ανθρώπων και δη των νέων παιδιών ειδικά σε αυτές τις δύσκολες ιστορικά δεκαετίες που ζούμε. Και μιας και αναφέραμε την ιστορία, αν αναλογιστούμε ότι ο άνθρωπος ζωγράφιζε τους τοίχους, όπως αρχαιολόγοι έχουν ανακαλύψει στην Πομπηία το 100 π.Χ.-79 μ.Χ. ή στη δική μας γειτονιά στους Μινωικούς τόπους το 1700-1600 π.Χ. και στους αιγυπτιακούς τάφους το 3150 π.Χ. θα φτάσουμε ψάχνοντας στις πρώτες ζωγραφιές στο σπήλαιο Chauvet στη Γαλλία σε τοιχογραφίες που χρονολογούνται στο 30.000 π.Χ. Σύντομα θα καταλήξουμε αναλογιζόμενοι τα παραπάνω πως ο άνθρωπος θα συνεχίζει να εκφράζεται και να παράγει τέχνη όπως κάνει από την Ανώτερη Παλαιολιθική εποχή μέχρι σήμερα. Για να επανέλθουμε όμως στο σήμερα και στο θέμα που θέλουμε να αναδείξουμε, εύκολα γίνεται κατανοητό ότι αυτή η τέχνη που μας περιβάλλει στην πόλη μας, αλλά και σε κάθε πόλη του πλανήτη, είναι δύσκολο να εντοπιστεί. Εδώ λοιπόν έρχεται η εξέλιξη της τεχνολογίας και η ευρέως διαδεδομένη χρήση των έξυπνων τηλεφώνων καθώς πλέον με τις σύγχρονες τεχνολογίες μπορεί να συσσωρευτεί όλη η παραπάνω πληροφορία στην τσέπη μας μέσω μιας εφαρμογής τηλεφώνου, ώστε στον ελεύθερο χρόνο, στη βόλτα ή στο ταξίδι μας σε ένα άλλο μέρος, να είμαστε σε θέση εύκολα να εντοπίσουμε και να θαυμάσουμε αυτά τα όμορφα διάσπαρτα έργα και ίσως να αποτελέσει το έναυσμα ώστε να εκτιμηθεί η δουλειά των καλλιτεχνών από περισσότερο κόσμο, αλλά και να δώσει και στα νέα παιδιά μια τροφή για κριτική σκέψη πάνω στο πώς μπορεί να προάγει πολιτισμό η τέχνη του δρόμου, η τέχνη των πόλεων μας.

1.2 Σκοπός και στόχοι της εργασίας

Ο σκοπός της παρούσας εργασίας είναι η κατασκευή μιας Android εφαρμογής για έξυπνα κινητά τηλέφωνα της οποίας ο χρήστης θα μπορεί να έχει πρόσβαση σε διάφορα έργα που βρίσκονται είτε στον τόπο του είτε ως επισκέπτης σε μια πόλη. Η εφαρμογή που υλοποιήθηκε ονομάζεται Street Art & More! και κατασκευάστηκε σε Android Studio με γλώσσα προγραμματισμού Java.

Οι στόχοι της εργασίας περιλαμβάνουν τα ακόλουθα:

1. Την συσσώρευση της πληροφορίας των Street Art σε μια βάση δεδομένων.
2. Την δημιουργία μιας λίστας προβολής των Street Art όπου ο χρήστης θα μπορεί να δει σε εικόνα τα Street Art.
3. Την δυνατότητα λήψης οδηγιών για όποια εικόνα ενδιαφερθεί ο χρήστης.
4. Την τοποθέτηση των έργων πάνω σε χάρτη για μια ευρεία προβολή.
5. Την δυνατότητα λήψης σε εικόνα ενός Street Art από τον χρήστη και η αποθήκευσή του στη βάση.
6. Την ειδοποίηση του χρήστη σύμφωνα με της δικές του προτιμήσεις για τυχόν έργα που βρίσκονται σε κοντινή του απόσταση.

1.3 Βασικοί ορισμοί

Για την υλοποίηση της εφαρμογής Street Art & More! χρησιμοποιήθηκε το προγραμματιστικό περιβάλλον του Android Studio με τη χρήση γλώσσας προγραμματισμού Java. Απαραίτητη ήταν η χρήση βάσης δεδομένων, μιας πλατφόρμας για την αυθεντικοποίηση του χρήστη και ένας αποθηκευτικός χώρος «νέφους» γι' αυτό και χρησιμοποιήθηκε η firebase πλατφόρμα της Google

που επιπλέον είναι ενσωματωμένη στο προγραμματιστικό περιβάλλον του Android Studio όπως θα δούμε παρακάτω.

1.3.1 Android

Το Android είναι ένα λειτουργικό σύστημα για κινητά που βασίζεται σε μια τροποποιημένη έκδοση του πυρήνα Linux και άλλων λογισμικών ανοιχτού κώδικα, σχεδιασμένο κυρίως για κινητές συσκευές με οθόνη αφής, όπως smartphone, tablet. Εμπορικά χορηγείται από την Google αφότου αποκαλύφθηκε τον Νοέμβριο του 2007, με την πρώτη εμπορική συσκευή Android να λανσάρεται τον Σεπτέμβριο του 2008. Τα βασικά στοιχεία προέρχονται από το Android Open Source Project (AOSP), το οποίο είναι δωρεάν λογισμικό ανοιχτού κώδικα με κύρια άδεια χρήσης βάσει της Apache License. Ο πηγαίος κώδικας έχει χρησιμοποιηθεί για την ανάπτυξη παραλλαγών του Android σε μια σειρά άλλων ηλεκτρονικών συσκευών, όπως κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, φορητές συσκευές αναπαραγωγής πολυμέσων, υπολογιστές, το καθένα με εξειδικευμένο περιβάλλον εργασίας χρήστη. Ορισμένα γνωστά παράγωγα περιλαμβάνουν το Android TV για τηλεοράσεις και το Wear OS για φορητές συσκευές, που αναπτύχθηκαν και τα δύο από την Google. Το Android είναι το λειτουργικό σύστημα με τις περισσότερες πωλήσεις παγκοσμίως σε smartphone από το 2011 και σε tablet από το 2013. Από τον Μάιο του 2021, έχει πάνω από τρία δισεκατομμύρια μηνιαίους ενεργούς χρήστες, τη μεγαλύτερη εγκατεστημένη βάση οποιουδήποτε λειτουργικού συστήματος και από τον Ιανουάριο του 2021, το Google Play Store διαθέτει πάνω από 3 εκατομμύρια εφαρμογές.

1.3.2 Android Studio

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την δημιουργία εφαρμογών Android, με βάση το IntelliJ IDEA, το οποίο ανακοινώθηκε στις 16 Μαΐου 2013 κατά τη διάρκεια της εκδήλωσης I/O 2013 της Google, κυκλοφόρησε την πρώτη σταθερή έκδοση τον Δεκέμβριο του 2014 ξεκινώντας από την έκδοση 1.0 με σκοπό να εξυπηρετήσει τον προγραμματισμό Android εφαρμογών. Το Android Studio περιέχει όλα τα εργαλεία Android για το σχεδιασμό, τη δοκιμή, τον εντοπισμό σφαλμάτων και το προφίλ της εφαρμογής που ο προγραμματιστής θέλει να αναπτύξει. Το Android Studio χρησιμοποιεί το Gradle για τη διαχείριση του project, ένα εργαλείο Build Automation. Για την ανάπτυξη της εφαρμογής αρκεί να κάνουμε λήψη του Android Studio για την πλατφόρμα προτίμησής μας (Windows®, Mac OS X ή Linux) από τον ιστότοπο προγραμματιστών του Android. Με το Android Studio μπορούμε να αναπτύξουμε και να δοκιμάσουμε την εφαρμογή μας είτε σε πραγματική συσκευή είτε σε εξομοιωτή. Μας παρέχει ένα ενοποιημένο περιβάλλον όπου μπορούμε να γράψουμε τον κώδικα μας για όλες τις συσκευές Android. Ακόμη μπορούμε να εφαρμόσουμε αλλαγές στο κώδικα μας και να τις τρέξουμε στην εφαρμογή ενώ εκτελείται στον εξομοιωτή, χωρίς να επανεκκινήσουμε την εφαρμογή. Επίσης μας προσφέρει πρότυπα κώδικα για να χρησιμοποιήσουμε και παρέχει ενσωμάτωση GitHub καθώς και εκτεταμένα εργαλεία και πλαίσια δοκιμών, όπως και εργαλεία Lint για τον εντοπισμό της απόδοσης, τη χρηστικότητα, τη συμβατότητα έκδοσης και άλλα προβλήματα που θα κληθούμε να λύσουμε. Υποστηρίζει την C++ με το Android NDK το οποίο είναι ένα σύνολο εργαλείων που μας επιτρέπει να εισάγουμε τμήματα της εφαρμογής μας σε εγγενή κώδικα. Επίσης παρέχει ενσωματωμένη υποστήριξη για την πλατφόρμα Google Cloud, καθιστώντας εύκολη την ενσωμάτωση του Google Cloud Messaging και του App Engine.

1.3.3 Java

Η Java είναι μια αντικειμενοστρεφής γλώσσα προγραμματισμού που κυκλοφόρησε για πρώτη φορά από τη Sun Microsystems το 1995. Από τότε έχει εξελιχθεί για να τροφοδοτήσει ένα μεγάλο μερίδιο του σημερινού ψηφιακού κόσμου, παρέχοντας την αξιόπιστη πλατφόρμα πάνω στην οποία χτίζονται πολλές υπηρεσίες και εφαρμογές. Νέα καινοτόμα προϊόντα και ψηφιακές υπηρεσίες που έχουν σχεδιαστεί για το μέλλον, συνεχίζουν να βασίζονται επίσης στην Java. Ενώ οι περισσότερες σύγχρονες εφαρμογές Java συνδυάζουν τον χρόνο εκτέλεσης της Java και την εφαρμογή μαζί, εξακολουθούν να υπάρχουν πολλές εφαρμογές, ακόμη και ορισμένες ιστοσελίδες

που δεν λειτουργούν χωρίς την εγκατάσταση Java για επιτραπέζιους υπολογιστές. Από την γέννηση της Java μέχρι σήμερα έχουν κυκλοφορήσει πολλές εκδόσεις με τις υποστηριζόμενες εκδόσεις σήμερα να είναι οι Java 8 και 9. Για να γράψει κάποιος κώδικα Java δε χρειάζεται τίποτα άλλο παρά έναν επεξεργαστή κειμένου, όπως το Σημειωματάριο (Notepad) των Windows ή ο vim (γνωστός στο χώρο του Unix). Παρόλα αυτά, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) βοηθάει πολύ, ιδιαίτερα στον εντοπισμό σφαλμάτων (debugging). Υπάρχουν αρκετά διαθέσιμα IDEs τα περισσότερα των οποίων παρέχονται δωρεάν. Τα πιο γνωστά Ολοκληρωμένα Περιβάλλοντα Ανάπτυξης είναι το NetBeans της Apache, το Eclipse όπως και το IntelliJ Idea της JetBrains στο οποίο βασίζεται και το Android Studio όπου αναπτύχθηκε η εφαρμογή μας.

1.3.4 Firebase Platform

Η Firebase της Google είναι μια πλατφόρμα ανάπτυξης εφαρμογών που επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές iOS, Android και Ιστού. Η Firebase της Google προσφέρει πολλές δυνατότητες που τη καθιστούν ένα από τα σημαντικότερα εργαλεία ανάπτυξης και υποστήριξης εφαρμογών κινητών συσκευών και ιστού. Μειώνει τον φόρτο εργασίας και τον χρόνο ανάπτυξης. Η Firebase είναι μια cloud υπηρεσία Backend-as-a-Service (Baas) η οποία παρέχει στους προγραμματιστές μια ποικιλία εργαλείων και υπηρεσιών για να τους βοηθήσει να αναπτύξουν ποιοτικές εφαρμογές, να αναπτύξουν τη βάση χρηστών τους, να αποθηκεύσουν αρχεία και παρέχει λύσεις που εύκολα μπορούν να βελτιώσουν την απόδοση των εφαρμογών τους. Είναι χτισμένο στην υποδομή της Google και συνεργάζεται άψογα με την πλατφόρμα του Android Studio. Το Firebase κατηγοριοποιείται ως πρόγραμμα βάσης δεδομένων NoSQL, το οποίο αποθηκεύει δεδομένα σε έγγραφα που μοιάζουν με JSON. Στη παρακάτω εφαρμογή χρησιμοποιείται η πλατφόρμα της Firebase για την ανάγκη χρήσης βάσης δεδομένων, όπου προτιμήθηκε η Firestore, το Storage για χώρο αποθήκευσης στο cloud, η αυθεντικοποίηση των χρηστών με την Firebase Auth και το ML kit της firebase για τη χρήση machine learning.

1.3.5 Firebase Firestore

Το Cloud Firestore είναι μια ευέλικτη, επεκτάσιμη βάση δεδομένων για ανάπτυξη εφαρμογών κινητών, ιστού και διακομιστών από τη Firebase και το Google Cloud. Διατηρεί τα δεδομένα συγχρονισμένα μεταξύ των εφαρμογών-πελατών σε πραγματικό χρόνο και προσφέρει υποστήριξη εκτός σύνδεσης για κινητά και τον ιστό. Στα θετικά του Cloud Firestore επίσης προστίθεται η ενσωμάτωση και συνεργασία με άλλα προϊόντα της Firebase και Google Cloud. Ανήκει στην κατηγορία των NoSQL βάσεων δεδομένων. Το Cloud Firestore είναι επίσης διαθέσιμο σε native Node.js, Java, Python, Unity, C++ και Go SDK, εκτός από τα API REST και RPC. Τα δεδομένα μας αποθηκεύονται σε έγγραφα, οργανωμένα σε συλλογές που θυμίζουν την JSON δομή και υποστηρίζουν πολλούς διαφορετικούς τύπους δεδομένων, από απλές συμβολοσειρές και αριθμούς έως πολύπλοκα, ένθετα αντικείμενα. Μπορούμε επίσης να δημιουργήσουμε υποσυλλογές μέσα σε έγγραφα αλλά και πολύπλοκες ιεραρχικές δομές δεδομένων που κλιμακώνονται καθώς μεγαλώνει η βάση μας.

1.3.6 Firebase Cloud Storage

Το Cloud Storage της Firebase είναι μια ισχυρή, απλή και οικονομικά αποδοτική υπηρεσία αποθήκευσης δεδομένων. Το Firebase SDK για Cloud Storage προσθέτει ασφάλεια στις μεταφορτώσεις και λήψεις αρχείων για τις εφαρμογές μας, ανεξάρτητα από την ποιότητα του δικτύου καθώς υποστηρίζει την συνέχιση των μεταφορτώσεων ή των λήψεων από εκεί που σταμάτησαν, εξοικονομώντας χρόνο και bandwidth των χρηστών. Η χρήση των SDK μπορεί να γίνει για την αποθήκευση εικόνων, ήχου, βίντεο ή άλλου περιεχομένου που δημιουργείται από τον χρήστη.

1.3.7 Firebase Auth

Η Firebase Auth παρέχει την λύση μας για την αυθεντικοποίηση των χρηστών η οποία είναι προσαρμόσιμη, ανοιχτού κώδικα και χειρίζεται τις ροές διεπαφής χρήστη για τη σύνδεση τους. Εφαρμόζει βέλτιστες πρακτικές για τον έλεγχο ταυτότητας σε κινητές συσκευές και ιστότοπους. Παρέχει ολοκληρωμένη ασφάλεια και γρήγορη υλοποίηση μόνο μέσα σε λίγες γραμμές κώδικα και αυτό είναι που καθιστά ιδανικό εργαλείο στην ανάπτυξη της εφαρμογής μας καθώς θα χρειάζονταν πολλαπλάσιος χρόνος για την ανάπτυξη ενός δικού μας συστήματος αυθεντικοποίησης χρήστη.

1.3.8 Firebase ML Kit

Το ML Kit της Firebase είναι ένα SDK για κινητές συσκευές που φέρνει την τεχνογνωσία του Machine Learning της Google στις Android και iOS εφαρμογές σε ένα ισχυρό αλλά εύχρηστο πακέτο. Δίχως την απαίτηση εμπειρίας στη μηχανική εκμάθηση, μπορεί ο χρήστης να εφαρμόσει τη λειτουργικότητα που χρειάζεται σε λίγες μόνο γραμμές κώδικα. Δεν χρειάζεται να έχει βαθιά γνώση των νευρωνικών δικτύων ή τη βελτιστοποίηση μοντέλων. Παρέχονται πολλές λειτουργίες είτε μέσω cloud είτε στη συσκευή και όπως θα δούμε στη παρακάτω εφαρμογή γίνεται χρήση η λειτουργία για την λήψη πληροφοριών περιεχομένου εικόνας image labeling.

2 Σχεδίαση της εφαρμογής

2.1 Χρήση διαγραμμάτων UML

Η Ενοποιημένη Γλώσσα Σχεδίασης Προτύπων (Unified Modeling Language) UML, είναι η πρότυπη γλώσσα σχεδίασης και αποτύπωσης προτύπων στη μηχανική λογισμικού. Χρησιμοποιείται για τη γραφική απεικόνιση, τον προσδιορισμό, την κατασκευή και την τεκμηρίωση των στοιχείων ενός συστήματος λογισμικού. Για την αναπαράσταση της εφαρμογής μας παρακάτω θα χρησιμοποιήσουμε διάγραμμα περίπτωσης χρήσης και διάγραμμα ακολουθίας.

2.1.1 Use Case Diagrams

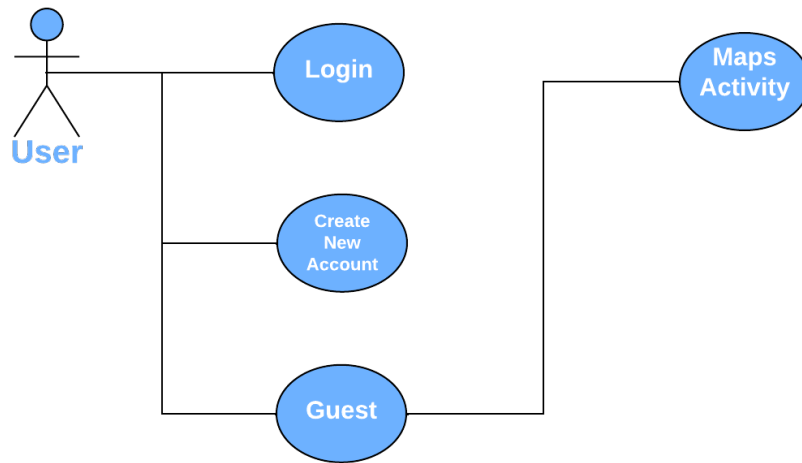
Τα διαγράμματα περιπτώσεων χρήσης (use-case) παρουσιάζουν ένα σύστημα όπως αυτό θα φαινόταν από τη θέση ενός εξωτερικού παρατηρητή. Τα διαγράμματα αυτά απεικονίζουν τη συμπεριφορά ενός συστήματος γραφικά και μπορεί να καλύπτουν όλο το σύστημα ή τα πιο γενικά υποσυστήματα τα οποία απεικονίζονται με τα use-cases. Η ανάγκη της ανάλυσης των συστημάτων σε διαγράμματα use-case εφαρμόζεται ώστε να γίνει κατανοητό πως το σύστημα είναι σχεδιασμένο να λειτουργεί. Στη φάση της σχεδίασης τα χρησιμοποιούμε για να καθορίσουμε τη συμπεριφορά του συστήματος για να βοηθηθούμε στη φάση της υλοποίησης. Ένα διάγραμμα use-case περιέχει :

- **Actors:** Χρήστες είναι αυτοί οι οποίοι αλληλοεπιδρούν με το σύστημα. Μπορεί να είναι ένα φυσικό πρόσωπο ή ένα άλλο σύστημα, στη περίπτωσή μας ο χρήστης της εφαρμογής.
- **Use cases:** Τις περιπτώσεις χρήσης οι οποίες περιγράφουν το σύστημα από τις οποίες εύλογα μπορούμε να αντιληφθούμε τις λειτουργίες του
- **Interactions ή relationships:** Αλληλεπιδράσεις ή σχέσεις ή συσχετίσεις ανάμεσα σε Actors και Use cases.

Στα παρακάτω διαγράμματα περιπτώσεων χρήσης παρουσιάζονται οι ενέργειες που πρέπει να κάνει ο χρήστης της εφαρμογής στη πρώτη του γνωριμία με αυτή. Οι επιλογές που του δίνονται κατά την εκκίνηση της εφαρμογής είναι:

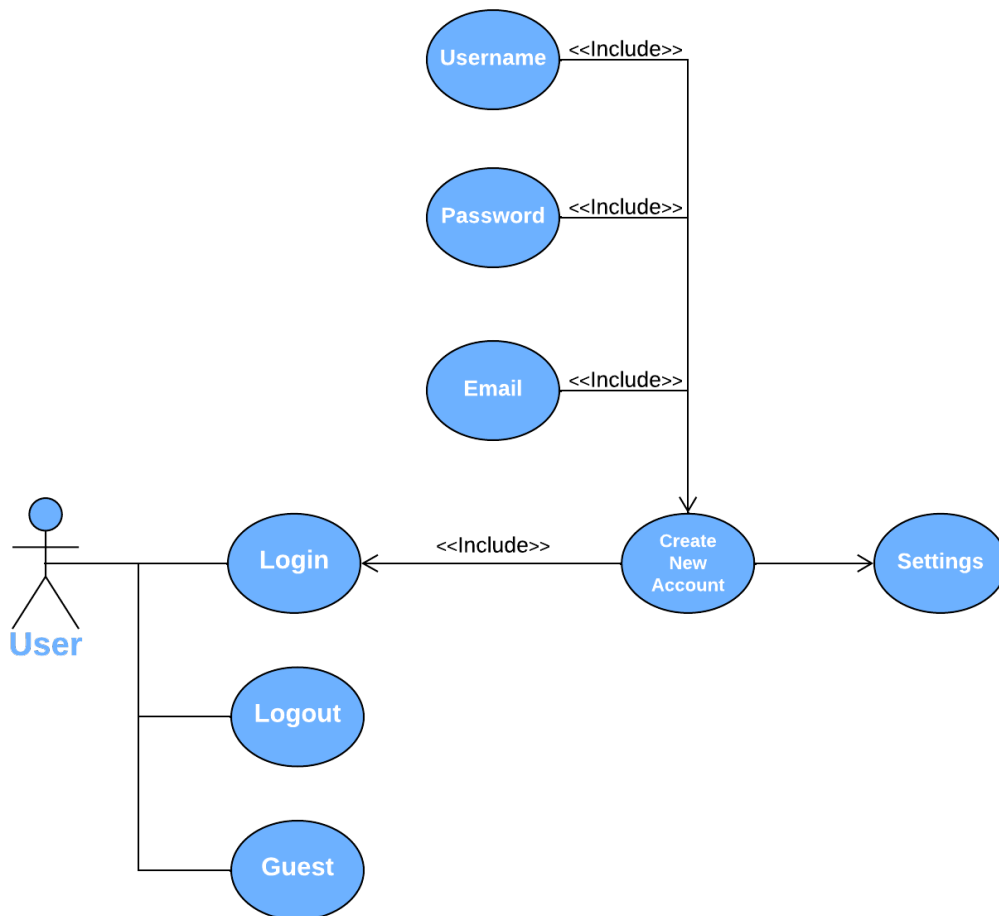
- Να εισέλθει στην εφαρμογή ως επισκέπτης (Guest User) όπου έχει περιορισμένες επιλογές.
- Να δημιουργήσει καινούριο λογαριασμό (Create New Account) ώστε να μπορεί να χρησιμοποιήσει όλες τις δυνατότητες της εφαρμογής.
- Να κάνει είσοδο (Login) στην εφαρμογή εφόσον είναι εγγεγραμμένος χρήστης.

Στη περίπτωση που ο χρήστης επιλέξει να μην εγγραφεί στο σύστημα και να συνεχίσει ως επισκέπτης, το σύστημα θα παρακάμψει τις βασικές του λειτουργίες και θα κατευθύνει τον χρήστη στον χάρτη. Εκεί μπορεί να δει συγκεντρωμένους δείκτες (markers) των εγγραφών Street Art που υπάρχουν τοποθετημένα στη περιοχή.



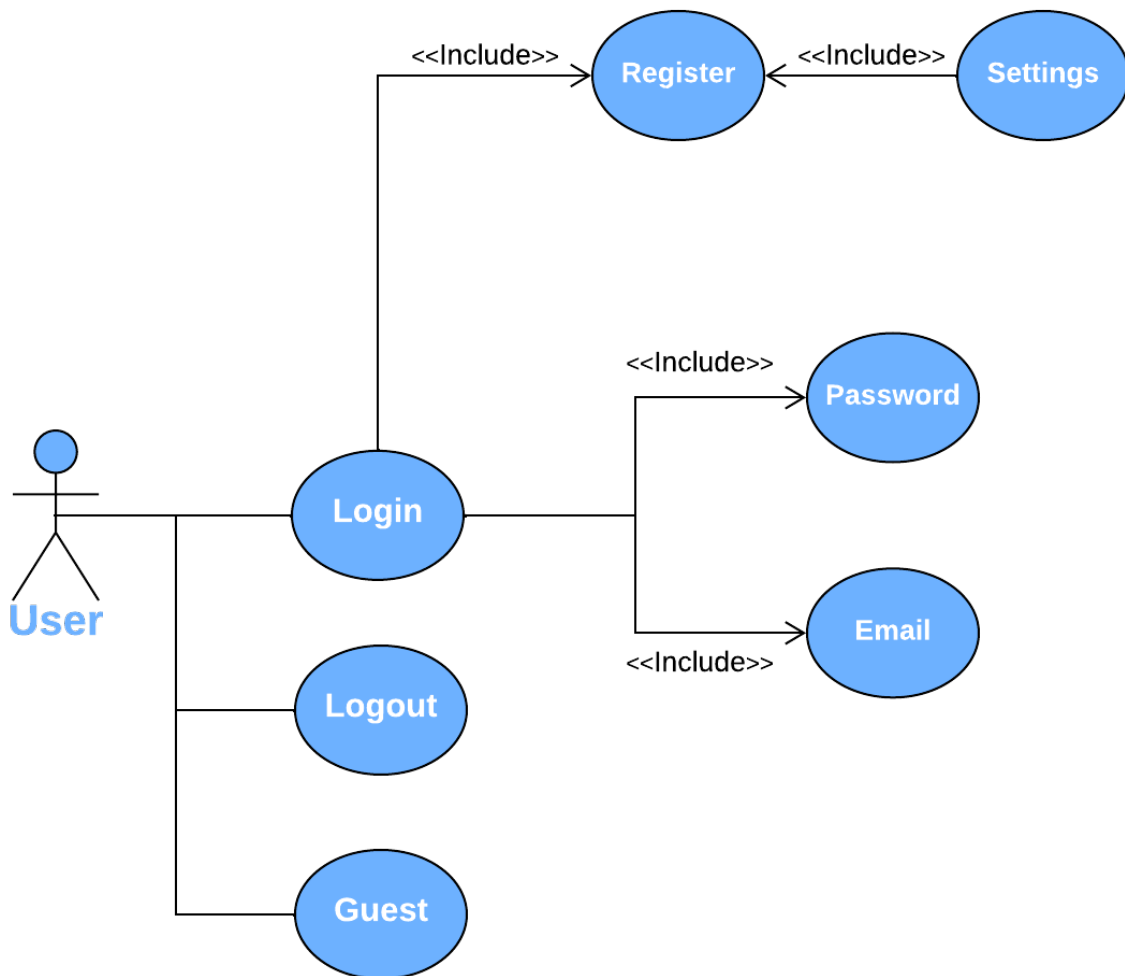
Εικόνα 1. Use Case Enter as Guest: Ο χρήστης επιλέγει να εισέλθει ως επισκέπτης.

Στην περίπτωση που ο χρήστης επιλέξει να δημιουργήσει νέο λογαριασμό, τότε θα του ζητηθεί να συμπληρώσει μια φόρμα με τα απαραίτητα credentials (username, email και password). Εάν ακολουθήσει σωστά τη διαδικασία, ακολούθως θα κατευθυνθεί από την εφαρμογή στις ρυθμίσεις προτού εισέλθει στο σύστημα όπως φαίνεται και στο επόμενο διάγραμμα περίπτωσης που ακολουθεί.



Εικόνα 2. Use Case Create New Account: Ο χρήστης επιλέγει να δημιουργήσει λογαριασμό για να εισέλθει στο σύστημα.

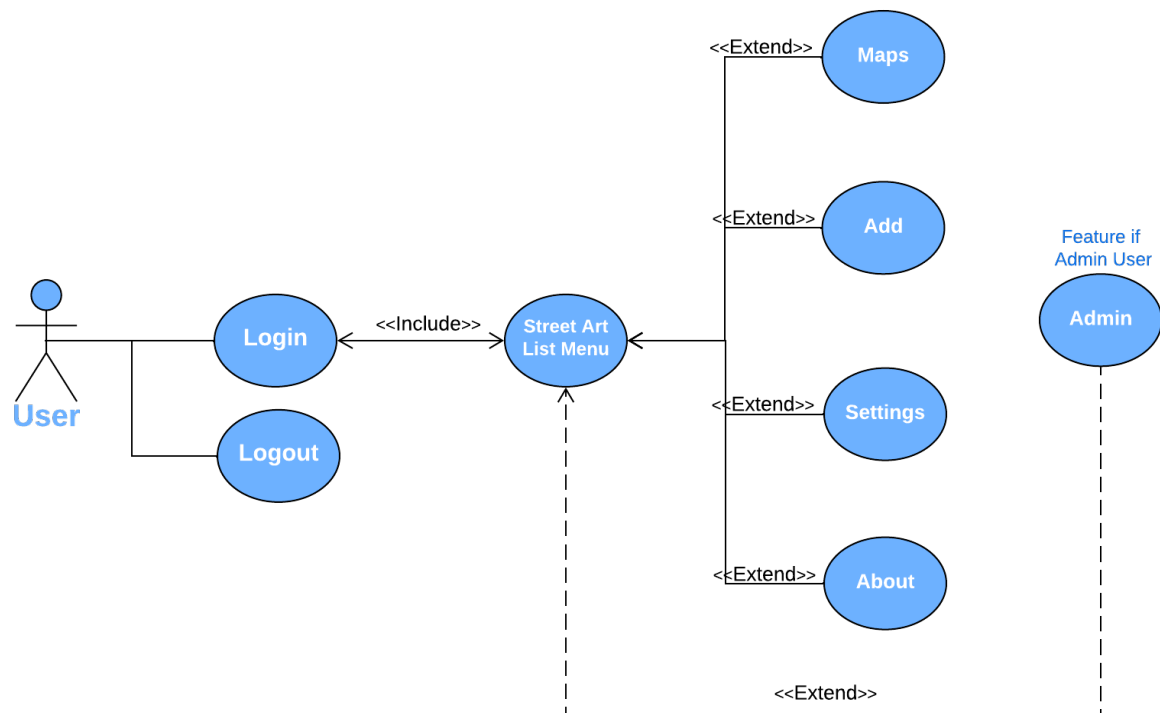
Στην τρίτη περίπτωση θα δούμε την περίπτωση χρήσης όπου ο χρήστης έχει στο παρελθόν εγγραφεί στο σύστημα και θέλει να πραγματοποιήσει είσοδο στο σύστημα. Σε αυτή τη περίπτωση θα πρέπει να εισάγει το email με το οποίο έχει εγγραφεί και τον κωδικό του πρόσβασης. Εφόσον τα στοιχεία που θα καταχωρήσει επαληθευθούν από τον authentication server, θα κατευθυνθεί απευθείας στο κύριο activity της εφαρμογής μιας και τα settings του έχουν αποθηκευτεί από την πρώτη φορά που δημιούργησε λογαριασμό και παραμένουν τα ίδια από την πρώτη φορά. Σε αυτή τη περίπτωση εάν θελήσει να αλλάξει ρυθμίσεις θα πρέπει να επιλέξει τα settings από το μενού όπως θα δούμε αργότερα.



Εικόνα 3. Use Case Login: Ο χρήστης κάνει είσοδο στο σύστημα.

Στο τελευταίο διάγραμμα περίπτωσης χρήσης απεικονίζεται το σύστημα στην περίπτωση που ο χρήστης είναι ήδη εγγεγραμμένος και πλέον έχει πρόσβαση σε όλες τις επιλογές και δυνατότητες της εφαρμογής. Επίσης απεικονίζεται η περίπτωση στην οποία ο χρήστης έχει συνδεθεί ως Admin user όπου επεκτείνονται οι λειτουργίες του συστήματος με ένα νέο activity όπου ο admin user μπορεί να δει κάποιες πληροφορίες όσον αφορά το περιεχόμενο της εφαρμογής που διαμορφώνεται από τους χρήστες. Οι επιλογές του μενού που αναπτύσσονται στο επόμενο διάγραμμα είναι:

- Maps
- Add New
- Settings
- About
- Admin



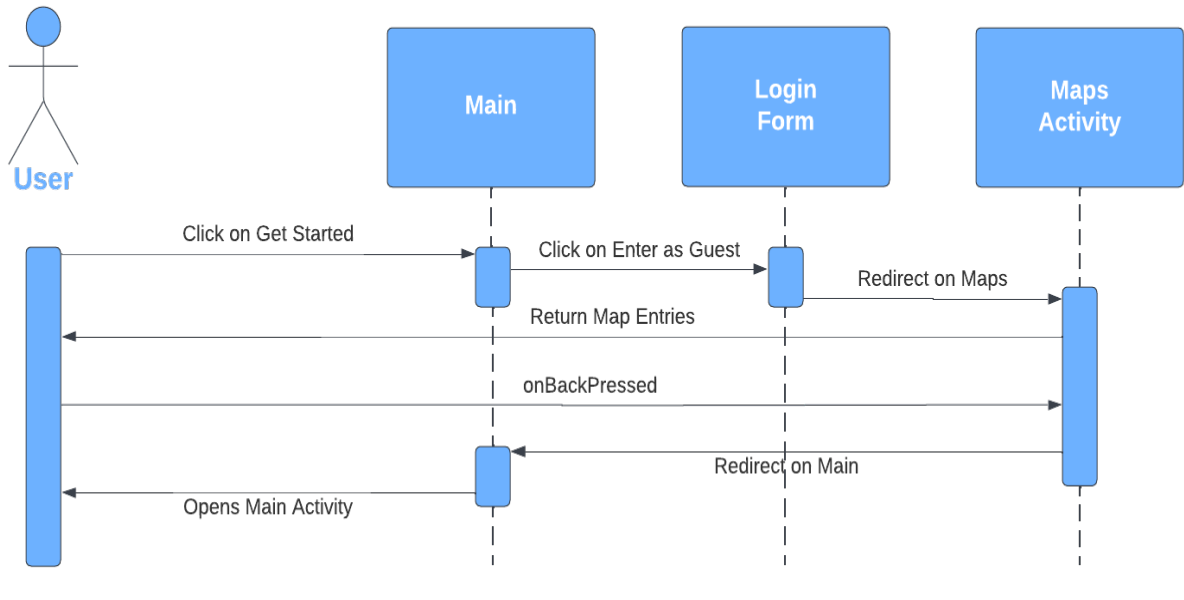
Εικόνα 4. Use Case Menu: Ο χρήστης έχει πραγματοποιήσει είσοδο και έχει πρόσβαση στο μενού.

2.1.2 Sequence Diagrams

Τα διαγράμματα ακολουθίας (Sequence Diagrams) χρησιμοποιούνται για να δείξουν την επικοινωνία ανάμεσα σε αντικείμενα με χρονική σειρά αλλά όχι όμως και με χρονική διάρκεια. Παρουσιάζουν την αλληλεπίδραση των αντικειμένων μέσω της ανταλλαγής μηνυμάτων. Ένα διάγραμμα ακολουθίας περιέχει χειριστές αντικείμενα και μηνύματα που ανταλλάσσουν τα αντικείμενα. Τα αντικείμενα οργανώνονται στον οριζόντιο άξονα και τα γεγονότα στον κάθετο άξονα του χρόνου. Τα διαγράμματα ακολουθίας χρησιμοποιούνται για να περιγράψουν τη ροή του ελέγχου μέσα στο σύστημα. Είναι βασική κατηγορία διαγραμμάτων για την ανάλυση και κατανόηση των συστημάτων.

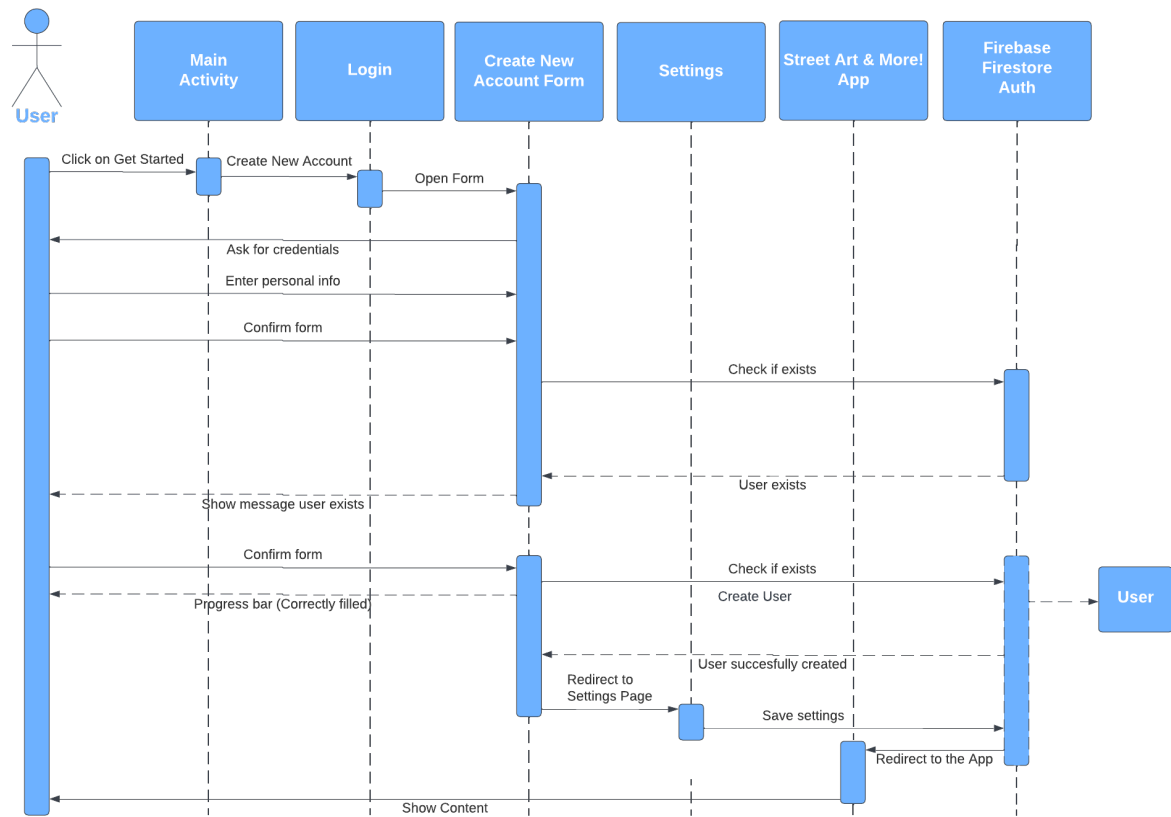
Στα διαγράμματα ακολουθίας παρακάτω φαίνονται όπως και στα διαγράμματα περίπτωσης χρήσης η διαδικασία του χρήστη για να εισέλθει ως επισκέπτης, η διαδικασία εγγραφής του στο σύστημα, η διαδικασία εισόδου του εφόσον έχει εκ των προτέρων δημιουργήσει λογαριασμό, όπως και οι βασικές λειτουργίες της εφαρμογής όπου μέσω των διαγραμμάτων ακολουθίας, παρότι δείχνουν περισσότερο πολύπλοκα από τα διαγράμματα περίπτωσης χρήσης, επιτυγχάνεται ευκολότερα η κατανόησή τους.

Το παρακάτω διάγραμμα παρουσιάζει τη διαδικασία εισόδου του χρήστη ως επισκέπτη. Όπως αναφέραμε και νωρίτερα ο χρήστης αφότου ανοίξει την εφαρμογή, θα του ζητηθεί να εισέλθει στο σύστημα. Στην περίπτωση που επιλέξει να παρακάμψει την εγγραφή του, αυτόματως θα παρακαμφθούν και οι βασικές λειτουργίες της εφαρμογής. Επιλέχθηκε να υπάρχει και αυτή η δυνατότητα προς το χρήστη ώστε η εφαρμογή να μην είναι αυστηρή στον τυχόν χρήστη που θέλει να την δοκιμάσει. Η λογική μας λέει ότι θα του δώσουμε τη δυνατότητα να δει ένα μικρό κομμάτι από το περιεχόμενο ώστε να τον προϊδεάσουμε για την λειτουργία της εφαρμογής, και σε περίπτωση που τον «κερδίσουμε» θα επιστρέψει ώστε να δημιουργήσει τον προσωπικό του λογαριασμό. Οπότε όταν επιλέξει να συνδεθεί ως επισκέπτης όπως φαίνεται και στο διάγραμμα, θα κατευθυνθεί στον χάρτη της εφαρμογής όπου μπορεί να δει τις «πινέζες» με τα Street Art στη περιοχή.



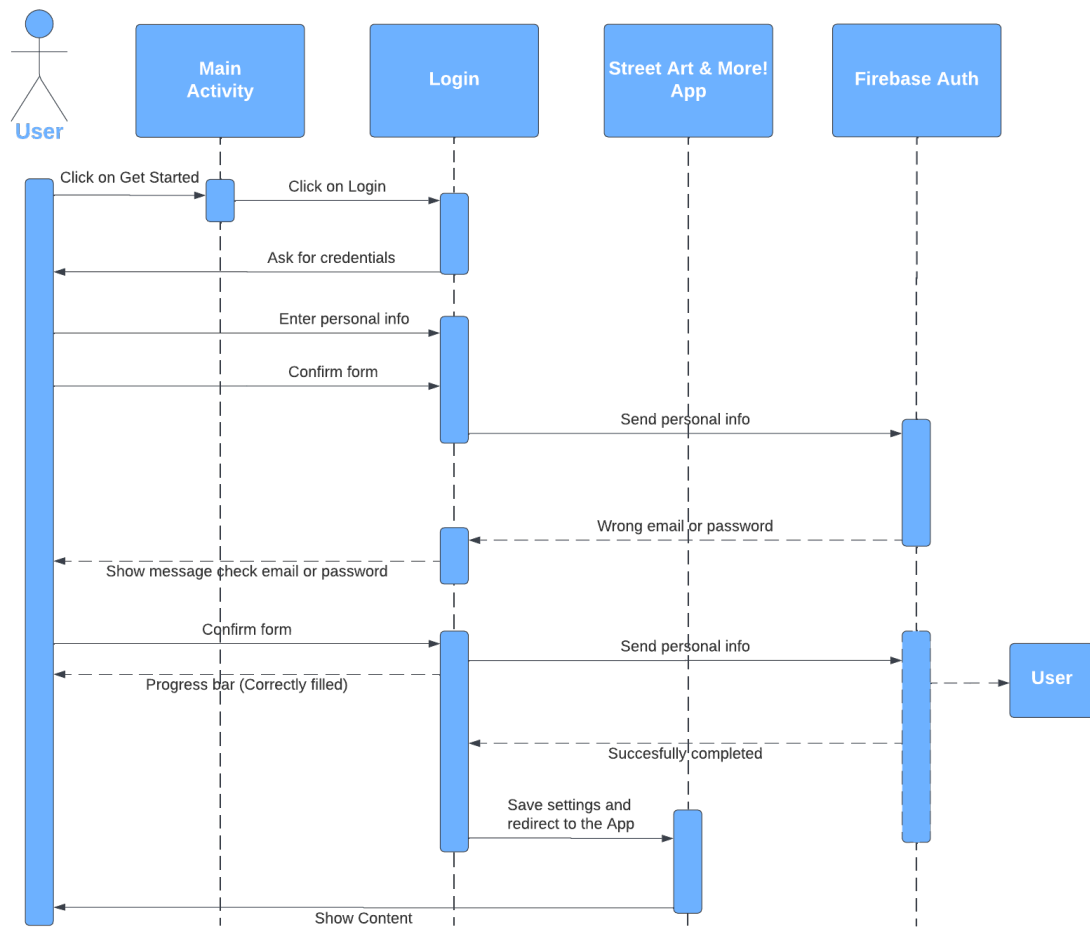
Εικόνα 5. Sequence Diagram: Ο χρήστης εισέρχεται στο σύστημα ως επισκέπτης.

Στο επόμενο διάγραμμα θα δούμε τη διαδικασία δημιουργίας νέου λογαριασμού χρήστη. Στο διάγραμμα απεικονίζονται τα μηνύματα που ανταλλάσσονται στη εφαρμογή και οι προϋποθέσεις της φόρμας εγγραφής, από όπου και θα περάσουν τα μηνύματα για να καταλήξουν στην Firebase και τον authentication server ώστε να καταχωρηθεί ο χρήστης, εφόσον βέβαια πρώτα έχει καταχωρήσει τα σωστά credentials. Ενδιαφέρον έχει η διαδικασία καθώς είναι εμφανές το πώς μετά την επιτυχή καταχώρηση του χρήστη, οδηγείται στις ρυθμίσεις της εφαρμογής όπου μετά την επιλογή των settings του, πατώντας save γίνεται εκ νέου επικοινωνία με τη firebase για να καταχωρηθούν στη Firestore οι προτιμήσεις του.



Εικόνα 6. Sequence Diagram: Ο χρήστης δημιουργεί λογαριασμό στο σύστημα.

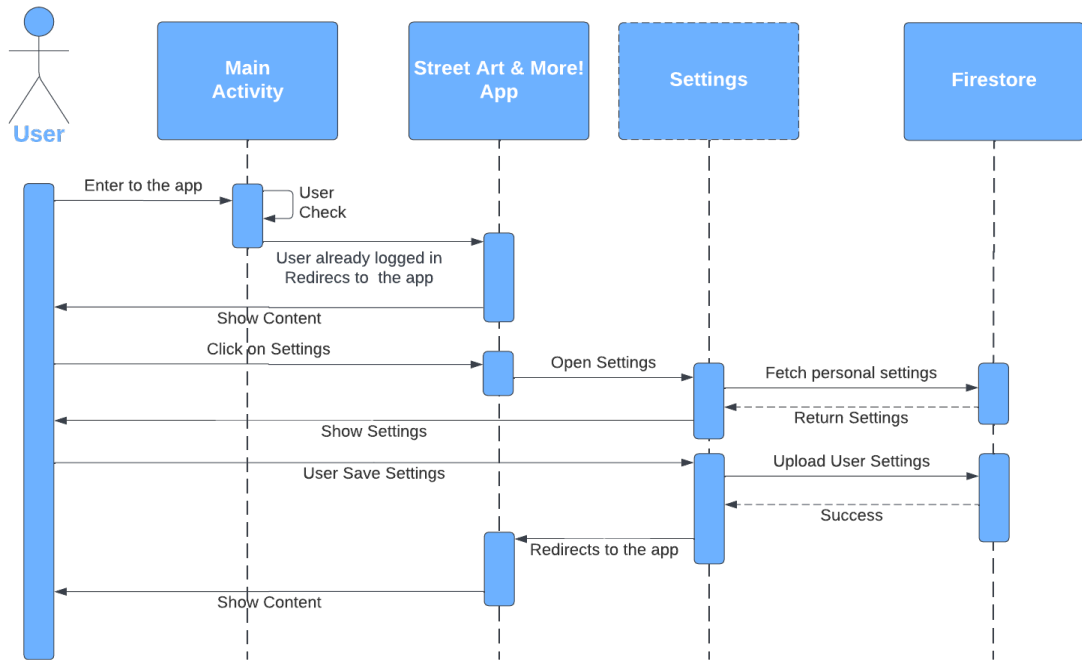
Στο επόμενο διάγραμμα θα δούμε τη διαδικασία του χρήστη να εισέλθει στο σύστημα αφού πλέον είναι πιστοποιημένος χρήστης. Αρκεί να εισάγει σωστά το email του και το password ώστε να εγκριθεί η είσοδος του από τον authentication server. Στη συνέχεια αποκτά πρόσβαση στις υπόλοιπες λειτουργίες του κυρίως μενού της εφαρμογής.



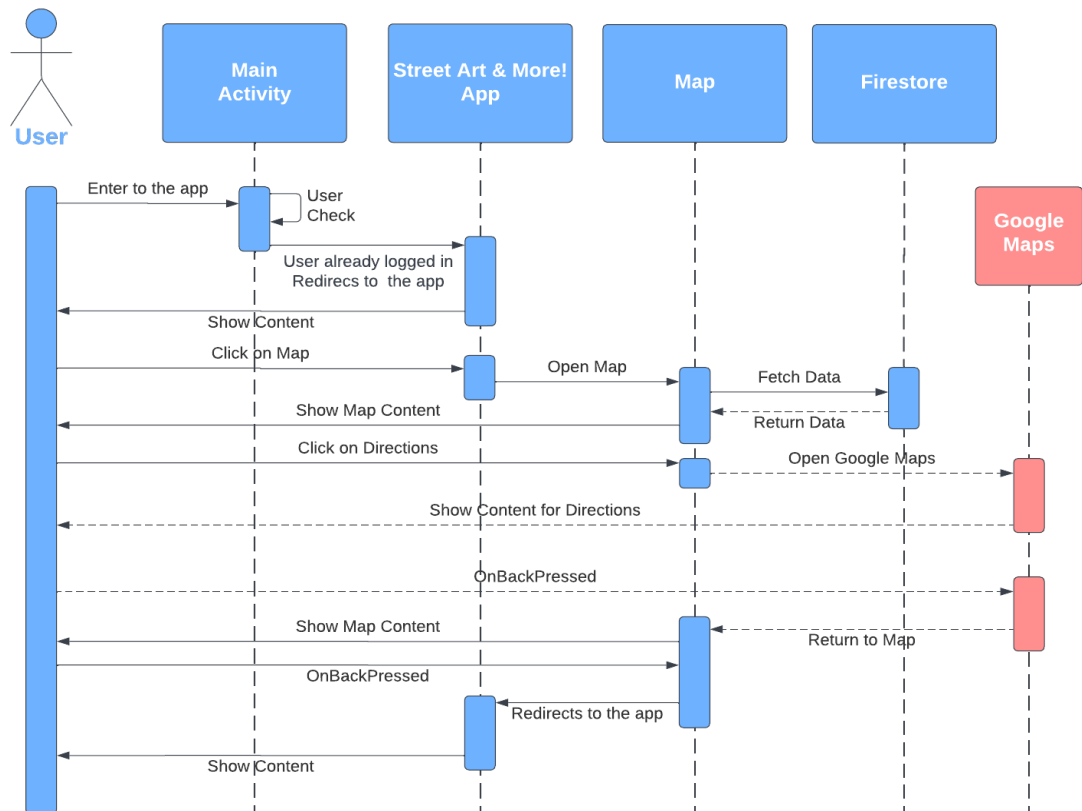
Εικόνα 7. Sequence Diagram: Ο χρήστης κάνει Login στο σύστημα.

Στο πρώτο από τα δύο παρακάτω διαγράμματα που ακολουθούν θα δούμε τη διαδικασία που πρέπει να κάνει ο χρήστης ώστε να αλλάξει τις ρυθμίσεις του. Όπως θα δούμε στο διάγραμμα πραγματοποιείται μια ανάγνωση read στη Firestore βάση δεδομένων και μια εγγραφή μετά την επίσκεψη του χρήστη με τις νέες ρυθμίσεις.

Στο δεύτερο διάγραμμα βλέπουμε μια διαδικασία που στη πράξη φαίνεται πολύ απλή, αλλά έχει ενδιαφέρον η απεικόνισή της όσον αφορά τα μηνύματα που ανταλλάσσονται. Ο χρήστης επιλέγει να ανοίξει το χάρτη, ο χάρτης εν συνεχεία θα πρέπει να καλέσει την βάση δεδομένων ώστε να αντλήσει τα στοιχεία που χρειάζεται για να τοποθετήσει τις πινέζες στο χάρτη. Αν ο χρήστης πατήσει σε μια από τις πινέζες, ακολούθως μπορεί να επιλέξει να οδηγηθεί μέσω του «directions button» στο Street Art. Τότε θα κατευθυνθεί στην εφαρμογή χαρτών της Google όπου μπορεί είτε να παραμείνει για να ακολουθήσει τις οδηγίες, είτε με το κουμπί «back» να επιστρέψει στην προβολή του χάρτη της εφαρμογής.

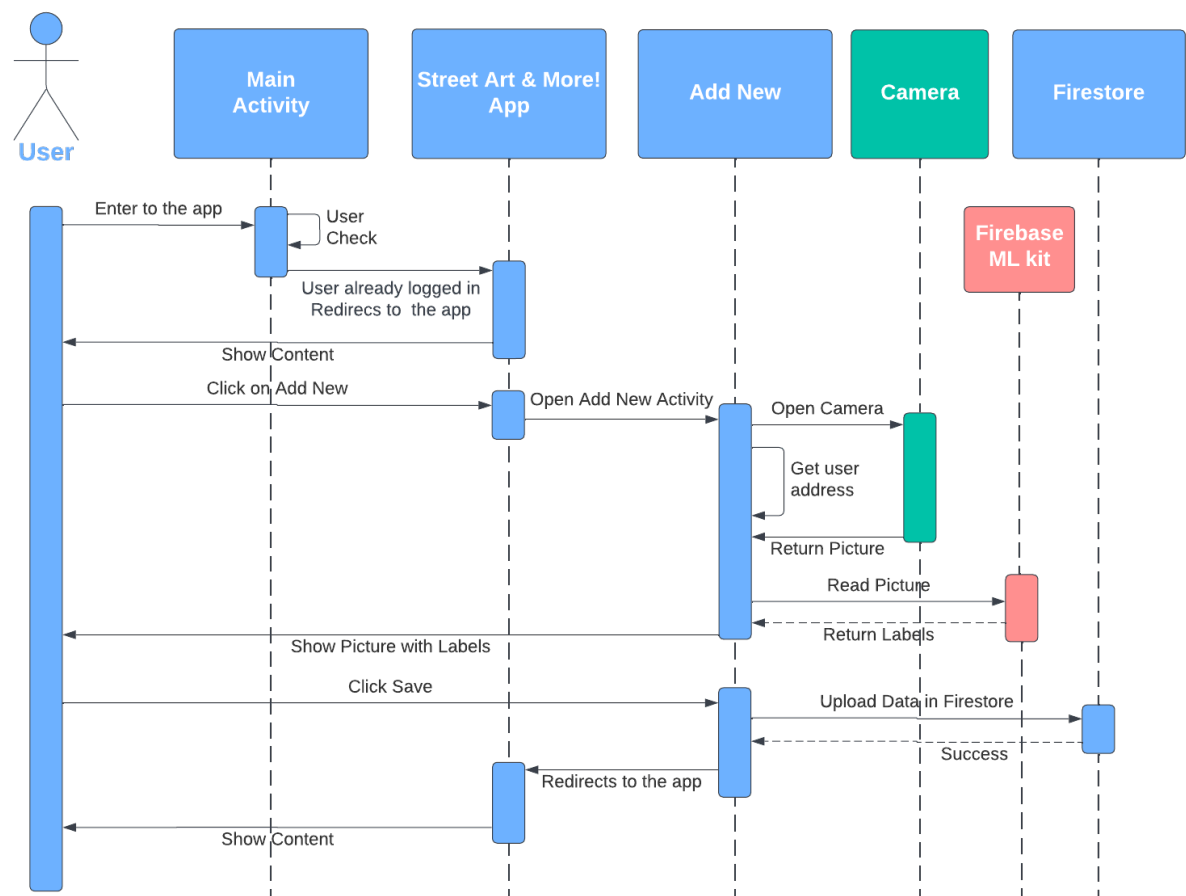


Εικόνα 8. Sequence Diagram: Ο χρήστης επιλέγει τα settings από το μενού.



Εικόνα 9. Sequence Diagram: Ο χρήστης επιλέγει τους χάρτες από το μενού.

Το τελευταίο διάγραμμα που φαίνεται στη παρακάτω εικόνα αφορά την επιλογή Add New Street Art, το οποίο επίσης έχει ενδιαφέρον από πλευράς ανταλλαγής μηνυμάτων. Θα δούμε ότι όταν ο χρήστης επιλέξει να προσθέσει ένα καινούριο Street Art, η εφαρμογή θα χρησιμοποιήσει τα resources του τηλεφώνου μας για να ανοίξει την camera. Στο παρασκήνιο το Location Service βρίσκει τις συντεταγμένες που βρισκόμαστε και με την χρήση του Geocoder θα μετατρέψει τις συντεταγμένες σε διεύθυνση την οποία και θα θέσει στο αντίστοιχο input text της εφαρμογής. Την ίδια στιγμή μόλις τραβήξουμε τη φωτογραφία θα αναλυθεί μέσω του Machine Learning kit της Firebase και θα μας επιστρέψει τα labels που διακρίνει στην εικόνα μας. Ο χρήστης λοιπόν κατά την ολοκλήρωση της διαδικασίας θα δει την εικόνα, τα labels και την διεύθυνση. Πατώντας save θα καταχωρηθεί ως νέο Street Art, όπως επίσης θα καταχωρηθούν και τα labels σε μια δική τους collection. Αυτή η collection ακολούθως χρησιμοποιείται για την εμφάνιση button στα settings, τα οποία μπορεί να ενεργοποιήσει ο χρήστης για να πάρει προτάσεις με βάση τις επιλογές του. Μια λειτουργία που θα την αναλύσουμε και παρακάτω.



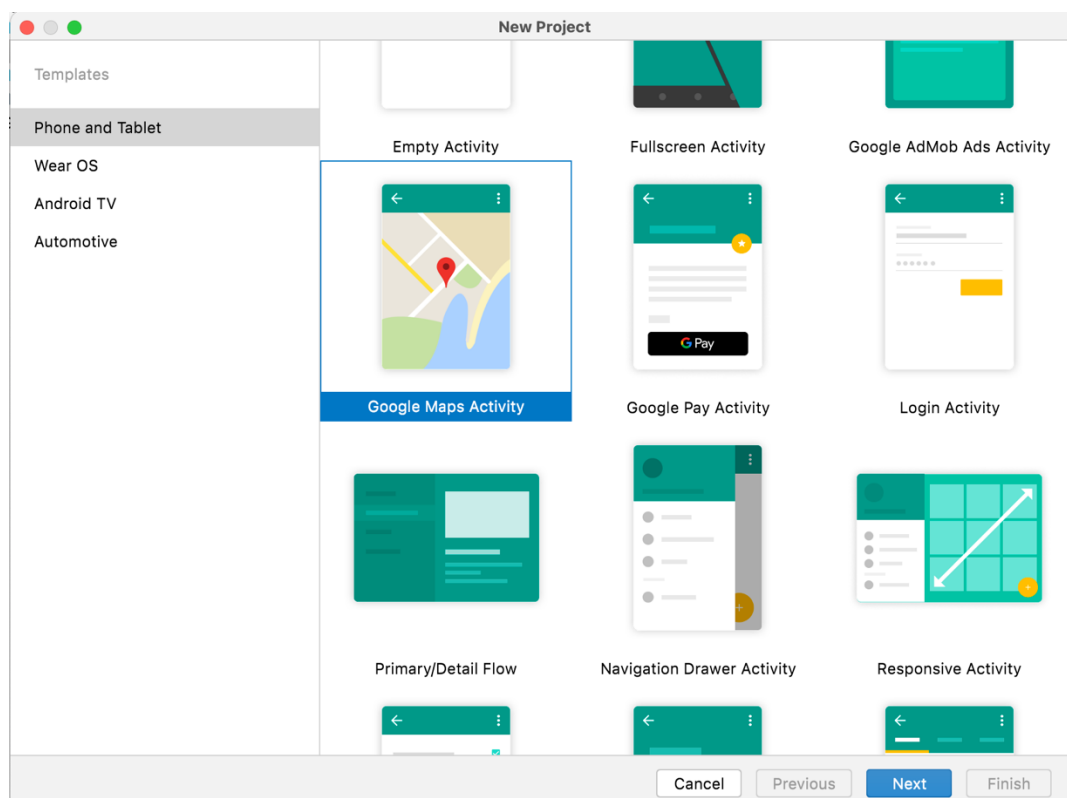
Εικόνα 10. Sequence Diagram: Ο χρήστης επιλέγει να προσθέσει ένα Street Art.

3 Υλοποίηση της εφαρμογής

Σε αυτό το κεφάλαιο θα αναλυθεί πως υλοποιήθηκε η εφαρμογή Street Art. Η υλοποίηση ενός project οτιδήποτε και αν αφορά αποτελεί από τη φύση του ένα πρόβλημα ή ένα σύνολο προβλημάτων τα οποία πρέπει να επιλυθούν για να είναι επιτυχές. Ο σκοπός της παρακάτω ανάλυσης είναι να παρουσιαστούν τα βήματα τα οποία ακολουθήθηκαν για την υλοποίηση, με στόχο την κατανόηση των προβλημάτων που ανέκυψαν ώστε να αναδειχθούν οι λύσεις τις οποίες επιλέξαμε για την δημιουργία της εφαρμογής μας.

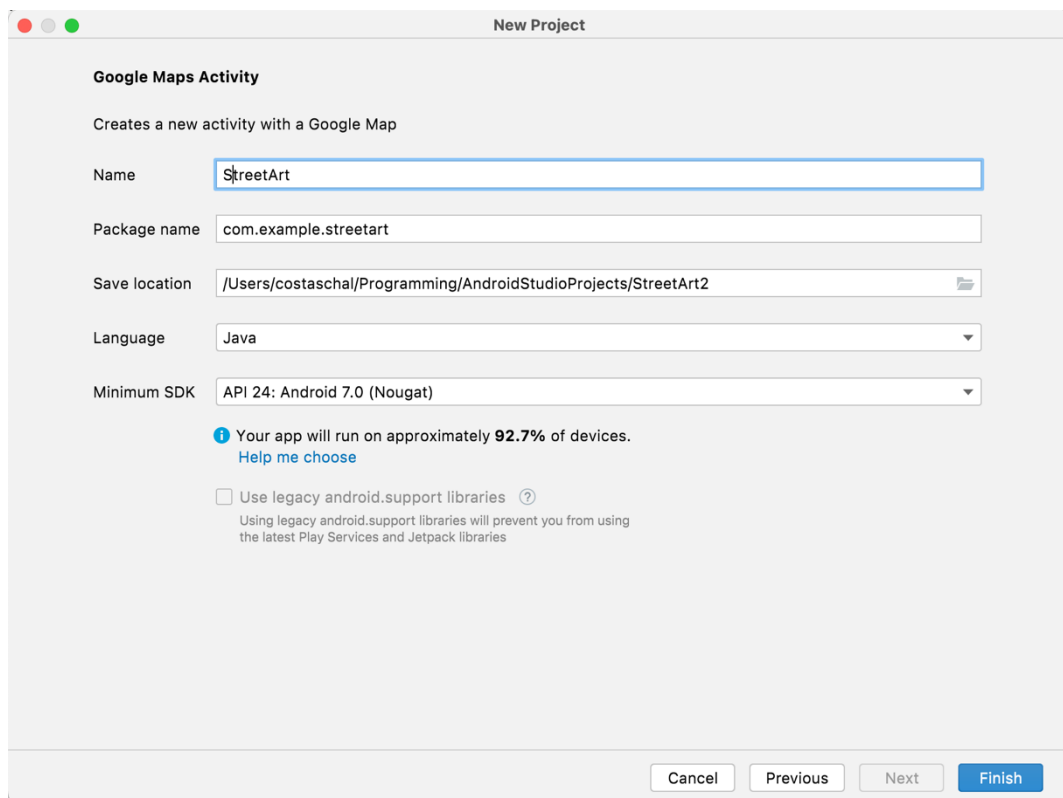
3.1 Χρήση του Android Studio και αρχικές ρυθμίσεις

Υπάρχουν πολλά frameworks που μπορούμε να αναπτύξουμε τις εφαρμογές μας, στα οποία υποστηρίζονται διάφορες προγραμματιστικές γλώσσες. Εδώ χρησιμοποιούμε Android Studio με την χρήση της γλώσσας προγραμματισμού Java. Το Android Studio μπορούμε να το προμηθευτούμε δωρεάν από την ιστοσελίδα προγραμματιστών του Android. Αφού το εγκαταστήσουμε στον υπολογιστή μας επιλέγουμε να δημιουργήσουμε νέο project.



Εικόνα 11. Android Studio: δημιουργία νέου project.

Όπως φαίνεται και στην εικόνα 11 επιλέγοντας να δημιουργήσουμε νέο project το Android studio θα μας προτείνει κάποιες επιλογές στα αριστερά του παραθύρου όσον αφορά τη συσκευή που απευθύνεται η εφαρμογή μας, και δεξιά κάποιες προτάσεις για το activity που θα επιλέξουμε. Στη περίπτωση μας και λόγω του ότι η εφαρμογή μας θα αφορά δεδομένα τα οποία θα τοποθετήσουμε αργότερα σε ένα χάρτη, θα επιλέξουμε το Google Maps Activity. Στη συνέχεια θα πρέπει να επιλέξουμε το όνομα της εφαρμογής καθώς και την γλώσσα προγραμματισμού στην οποία θέλουμε να αναπτύξουμε τον κώδικα της εφαρμογής μας. Επίσης θα πρέπει να επιλέξουμε και το ελάχιστο κιτ ανάπτυξης λογισμικού (SDK). Επιλέξαμε το Nougat διότι καλύπτει μεγάλο ποσοστό των συσκευών που χρησιμοποιούνται από τους χρήστες, πληροφορία που μας δίνει το σύστημα του Android Studio όπως φαίνεται και στην παρακάτω εικόνα.



Εικόνα 12. Android Studio: Επιλογή ονόματος της εφαρμογής, γλώσσας ανάπτυξης, minimum SDK.

Εν συνεχεία θα μεταφερθούμε στο προγραμματιστικό περιβάλλον του Android Studio όπου αφότου η εφαρμογή γίνει build αυτόματα θα μας ανοίξει την `MapsActivity.java` όπως και το `AndroidManifest.xml` στο οποίο ενημερωνόμαστε ότι για να μπορέσουμε να λάβουμε πληροφορίες χαρτών θα πρέπει να συνδεθούμε στην κονσόλα χαρτών της Google ώστε να ενεργοποιήσουμε το κατάλληλο API KEY και να κάνουμε copy το κλειδί του API ώστε να το τοποθετήσουμε στον κώδικα μας. Μετά από αυτές τις ενέργειες είμαστε πλέον σε θέση να ξεκινήσουμε να προγραμματίζουμε την εφαρμογή που οραματιζόμαστε

```

<!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow the directions here:

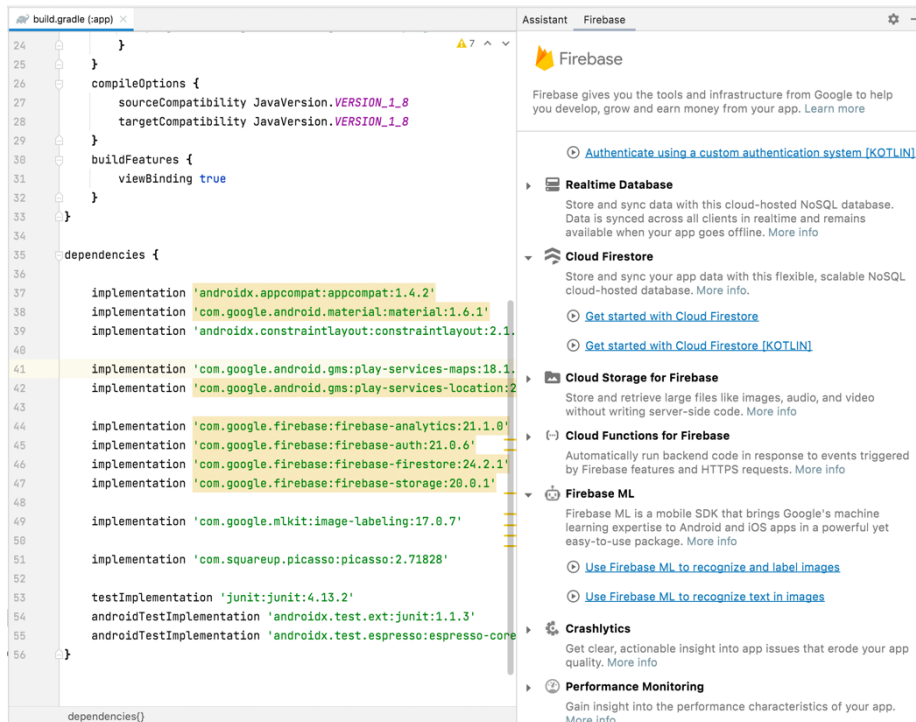
  https://developers.google.com/maps/documentation/android-sdk/get-api-key

  Once you have your API key (it starts with "AIza"), define a new property in your
  project's local.properties file (e.g. MAPS_API_KEY=Aiza...), and replace the
  "YOUR_API_KEY" string in this file with "${MAPS_API_KEY}".
-->
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="YOUR_API_KEY" />

```

Εικόνα 13. Android Studio: Μέρος του `AndroidManifest.xml` με σχόλια και το link που θα ακολουθήσουμε για να ενεργοποιήσουμε το API KEY.

Μία ακόμη Βασική λειτουργία που θα πρέπει να ρυθμίσουμε είναι το connection με την βάση δεδομένων, στην περίπτωση μας την firebase. Το προγραμματιστικό περιβάλλον του Android Studio μας βοηθά, μιας και μέσα από τα εργαλεία μπορούμε να κάνουμε κλικ στην επιλογή firebase όπου θα μας ανοίξει τον βοηθό (assistant), ο οποίος μας παρέχει ένα σύνολο επιλογών, ώστε αυτόματα να προσθέσουμε τα απαραίτητα dependencies στο build.gradle, να κάνουμε το connection με την βάση και ύστερα να είμαστε σε θέση προγραμματιστικά να δημιουργήσουμε τις collections και τα documents που χρειαζόμαστε.



Εικόνα 14. Android Studio: build.gradle και firebase assistant

3.1.1 Android Manifest

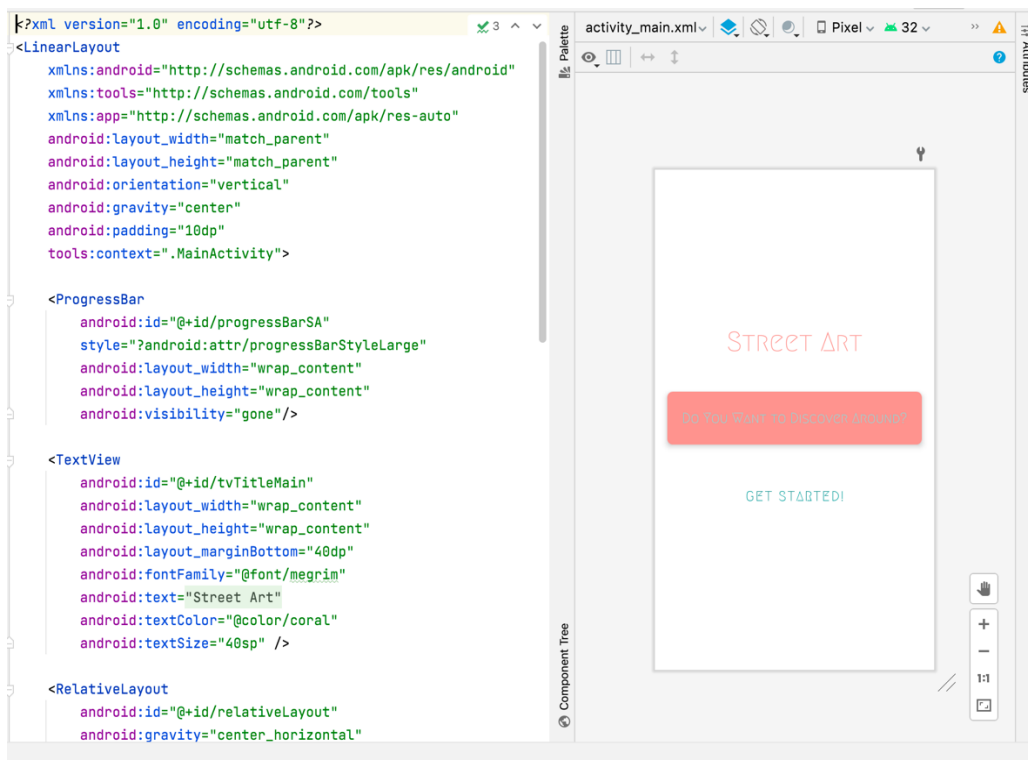
Το AndroidManifest.xml είναι ένα αρχείο δήλωσης και περιγράφει βασικές πληροφορίες σχετικά με την εφαρμογή μας, τα εργαλεία κατασκευής Android, το λειτουργικό σύστημα Android και το Google Play. Είναι απαραίτητο να υπάρχει σε κάθε εφαρμογή και περιέχει τα στοιχεία της εφαρμογής, τα οποία περιλαμβάνουν όλες τις δραστηριότητες, τις υπηρεσίες, τους παρόχους περιεχομένου, τα δικαιώματα που χρειάζεται η εφαρμογή για πρόσβαση σε προστατευμένα μέρη του συστήματος ή άλλων εφαρμογών, τα χαρακτηριστικά υλικού και λογισμικού που απαιτεί η εφαρμογή και πολλές άλλες λειτουργίες.

3.1.2 Build Gradle

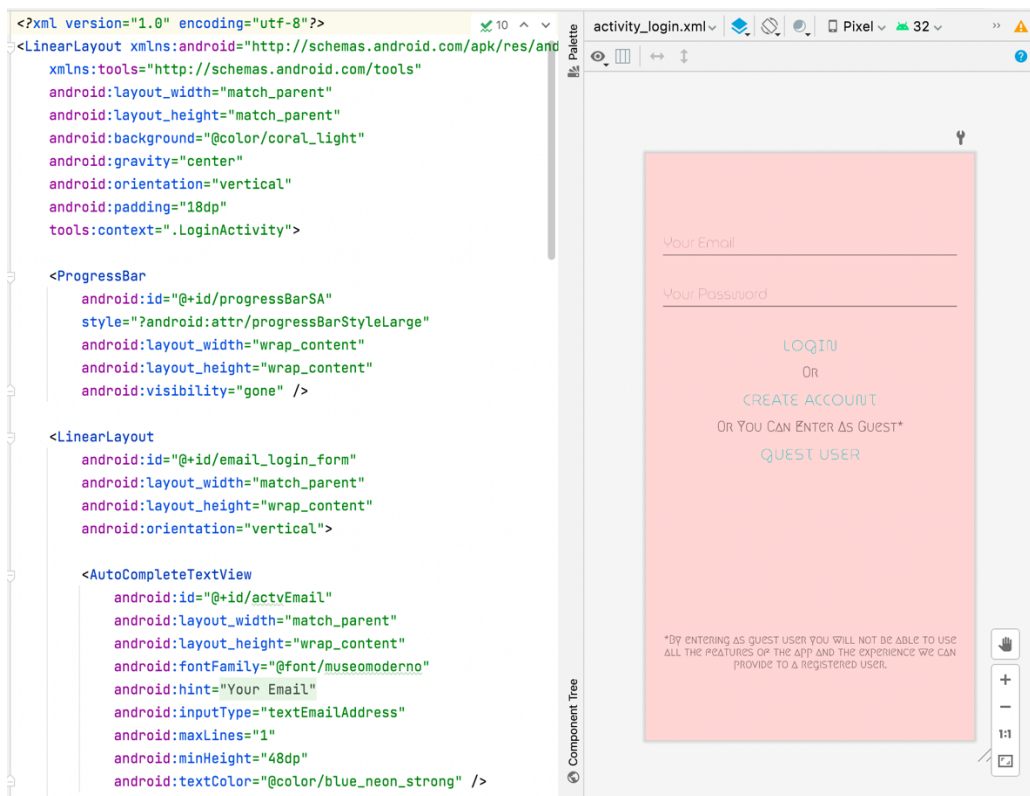
Το Android Studio χρησιμοποιεί το Gradle, ένα προηγμένο kit εργαλείων κατασκευής, για την αυτοματοποίηση και τη διαχείριση της διαδικασίας κατασκευής, ενώ μας επιτρέπει να ορίσουμε custom build configurations όπως παράδειγμα το Picasso για την προβολή εικόνων. Κάθε build configuration μπορεί να ορίσει το δικό του σύνολο κώδικα και πόρων, ενώ θα επαναχρησιμοποιήσει τα μέρη σε όλες τις εκδόσεις της εφαρμογής μας. Το Android plugin για το Gradle λειτουργεί με το kit εργαλείων δημιουργίας για να παρέχει διεργασίες και παραμετροποιήσιμες ρυθμίσεις που είναι συγκεκριμένες για τη δημιουργία και τη δοκιμή Android εφαρμογών.

3.2 Βασική ιδέα και στάδια ανάπτυξης της εφαρμογής

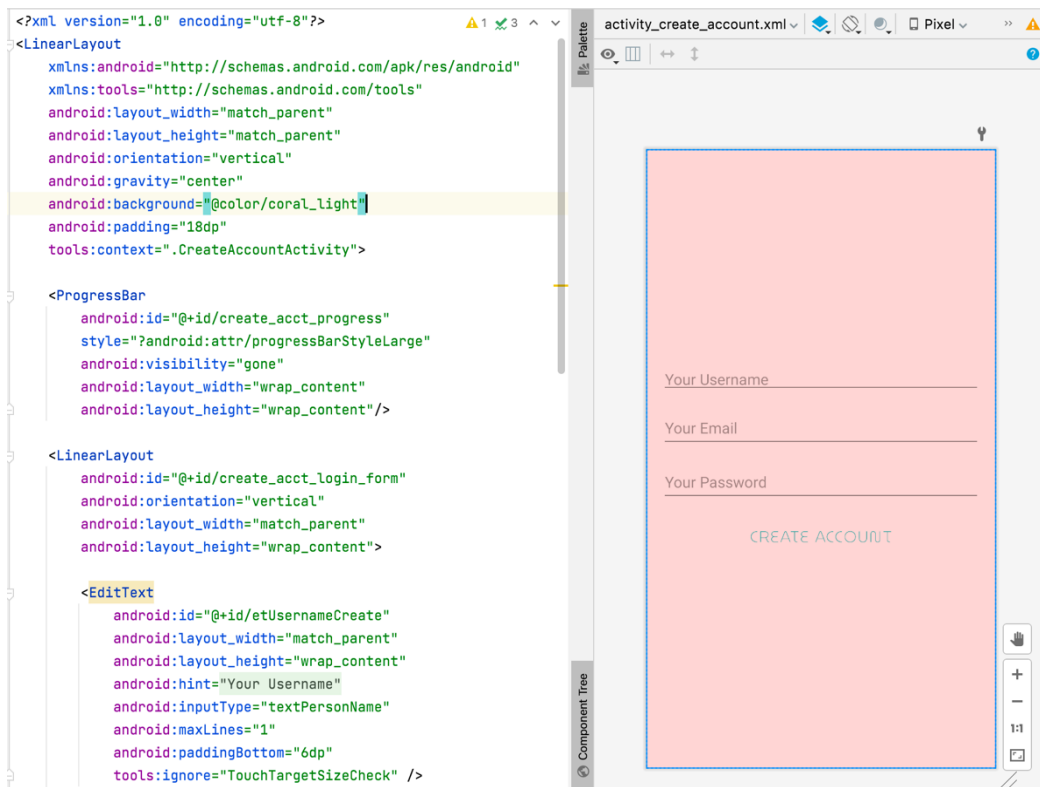
Η βασική ιδέα για τη λειτουργία της εφαρμογής που πρόκειται να αναπτύξουμε είναι μία εφαρμογή στην οποία ο χρήστης θα μπορεί αφού εισέλθει στο σύστημα να έχει πρόσβαση σε μία λίστα όπου θα του προβάλλονται τα Street Art έργα. Στη λίστα αυτή θέλουμε να του δώσουμε τη δυνατότητα να δει κάνοντας scroll down το σύνολο των έργων που είναι αποθηκευμένα στη βάση μας και να μπορεί να αλληλοεπιδράσει με αυτά. Μία από τις επιλογές που θέλουμε να του δώσουμε είναι να προβάλλει ένα από τα έργα σε μεγαλύτερο μέγεθος και μία δεύτερη αρχικά επιλογή θα είναι να μπορέσει να πάρει οδηγίες από αυτό ώστε μέσω του χάρτη να κατευθυνθεί στην φυσική τοποθεσία και να το παρατηρήσει από κοντά. Αρχικά βέβαια και προτού φτάσουμε στην προβολή της λίστας θα πρέπει ο χρήστης να κατασκευαστεί. Οπότε σκεφτήκαμε να δημιουργήσουμε ένα καινούργιο Activity, το MainActivity στο οποίο καλωσορίζουμε τον χρήστη και μέσω ενός κουμπιού θα τον μεταφέρουμε στην σελίδα ταυτοποίησης χρήστη, όπου μπορεί να εισέλθει είτε ως επισκέπτης, είτε να δημιουργήσει νέο λογαριασμό ή να εισέλθει εφόσον είναι ήδη εγγεγραμμένος χρήστης στην εφαρμογή μας.



Εικόνα 15. Android Studio: Το activity_main.xml ως αρχική σελίδα



Εικόνα 16. Android Studio: Το activity_login.xml με τις επιλογές του χρήστη.



Εικόνα 17. Android Studio: Το activity_create_account.xml με τα πεδία που πρέπει να συμπληρωθούν από το χρήστη.

Στις εικόνες 14, 15 και 16 βλέπουμε την σελίδα καλωσορίσματος του χρήστη, ακολούθως την σελίδα εισόδου στο σύστημα και τέλος την σελίδα δημιουργίας νέου λογαριασμού. Στο σημείο αυτό να πούμε ότι έχουν προκύψει οι εξής ανάγκες. Θα πρέπει να δηλώσουμε στο manifest τα permissions τα οποία η εφαρμογή μας θα πρέπει να χρησιμοποιήσει ώστε αρχικά να μπορεί να συνδεθεί στον ιστό, όπως επίσης και τα permissions που χρειάζεται ώστε να μπορεί να χρησιμοποιεί τα location services.

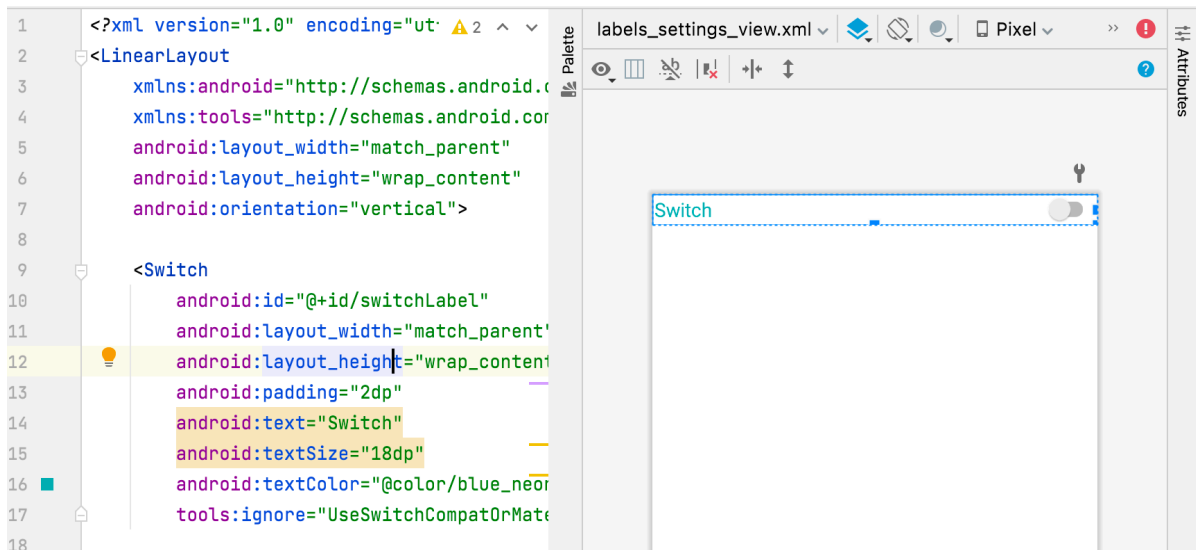
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

Εικόνα 18. Δήλωση των permission στο manifest.

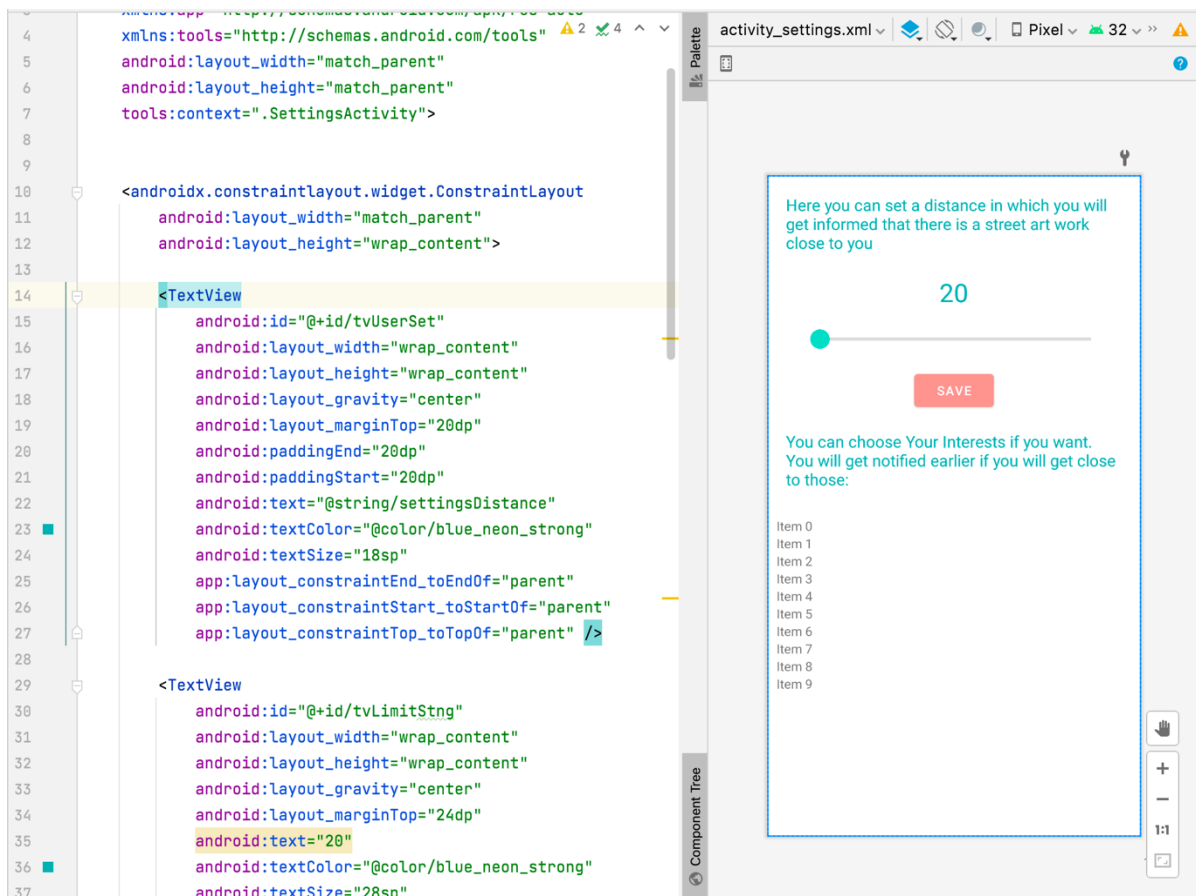
Στη συνέχεια στην CreateAccountActivity.java θα κάνουμε όλες τις απαραίτητες ενέργειες για την δημιουργία της βάσης δεδομένων από όπου ο χρήστης θα μπορεί να δημιουργεί λογαριασμό με username e-mail και password και ταυτόχρονα κατασκευάζουμε ένα καινούργιο φάκελο utilities όπου μέσα σε αυτόν δημιουργούμε μία καινούργια κλάση ως αντικείμενο, στην οποία θα αποθηκεύονται τα στοιχεία του χρήστη κατά την εγγραφή ή στη διαδικασία σύνδεσής του στο σύστημα ώστε με getInstance να μπορούμε σε οποιοδήποτε σημείο της εφαρμογής μας να αντλούμε στοιχεία του χρήστη. Τέτοιου είδους στοιχεία μπορεί να είναι πέρα από το username το email και το password του, η ημερομηνία εγγραφής ως timestamp ή προσωποποιημένες εγγραφές που μπορεί να επιλέξει στο επόμενο Activity μετά την εγγραφή του, που είναι το SettingsActivity.java.

Μία δεύτερη βασική λειτουργία κατά την σύλληψη της ιδέας της εφαρμογής ήταν να ζητάμε ένα όριο σε μέτρα από τον χρήστη, το οποίο θα χρησιμοποιείται ώστε όταν ο χρήστης πλησιάσει στο όριο που έχει θέσει μεταξύ ενός έργου Street Art, να λαμβάνει μία ειδοποίηση από την εφαρμογή ώστε να γνωρίζει πώς βρίσκεται σε κοντινή απόσταση και να μπορεί να το επισκεφθεί. Επίσης σκοπός μας είναι να χρησιμοποιήσουμε το machine learning kit της firebase σε κάθε φωτογραφία Street Art, ώστε να αντλήσουμε μέσω του labeler τα περιεχόμενα που έχει η κάθε εικόνα.

Τα παραπάνω labels θα τα αποθηκεύσουμε σε μία collection στην firebase ώστε να τα εμφανίσουμε ως button επιλογής στα settings. Για κάθε label που ο χρήστης θα έχει επιλεγμένο (true) θα αποθηκεύεται στο προσωπικό του document που έχουμε στη βάση, ώστε για όποιο Street Art περιέχει το αντίστοιχο label να τον ειδοποιούμε στη διπλάσια απόσταση από το όριο που έχει επιλέξει. Ο τρόπος για να τα ενσωματώσουμε στο Settings Activity επιτυγχάνεται με την χρήση RecyclerView. Θα πρέπει να δημιουργήσουμε ένα νέο Activity το οποίο θα περιέχει το μοντέλο του button label. Ακολούθως θα δημιουργήσουμε έναν καινούργιο φάκελο μέσα στον οποίον θα κατασκευάσουμε μία καινούργια κλάση SettingsRecyclerViewAdapter, στην οποία θα κάνουμε extend τον RecyclerViewAdapter. Χρησιμοποιώντας τις μεθόδους του RecyclerViewAdapter επιτυγχάνουμε την προβολή των button labels στη SettingsActivity.



Εικόνα 19. Android Studio: Το μοντέλο του button label που θα χρησιμοποιηθεί για την προβολή της λίστας μέσω του SettingsRecyclerViewAdapter.



Εικόνα 20. Android Studio: Το activity_settings.xml με χρήση Seekbar για την επιλογή της απόστασης και τη recycler view όπου θα προβληθούν τα labels buttons

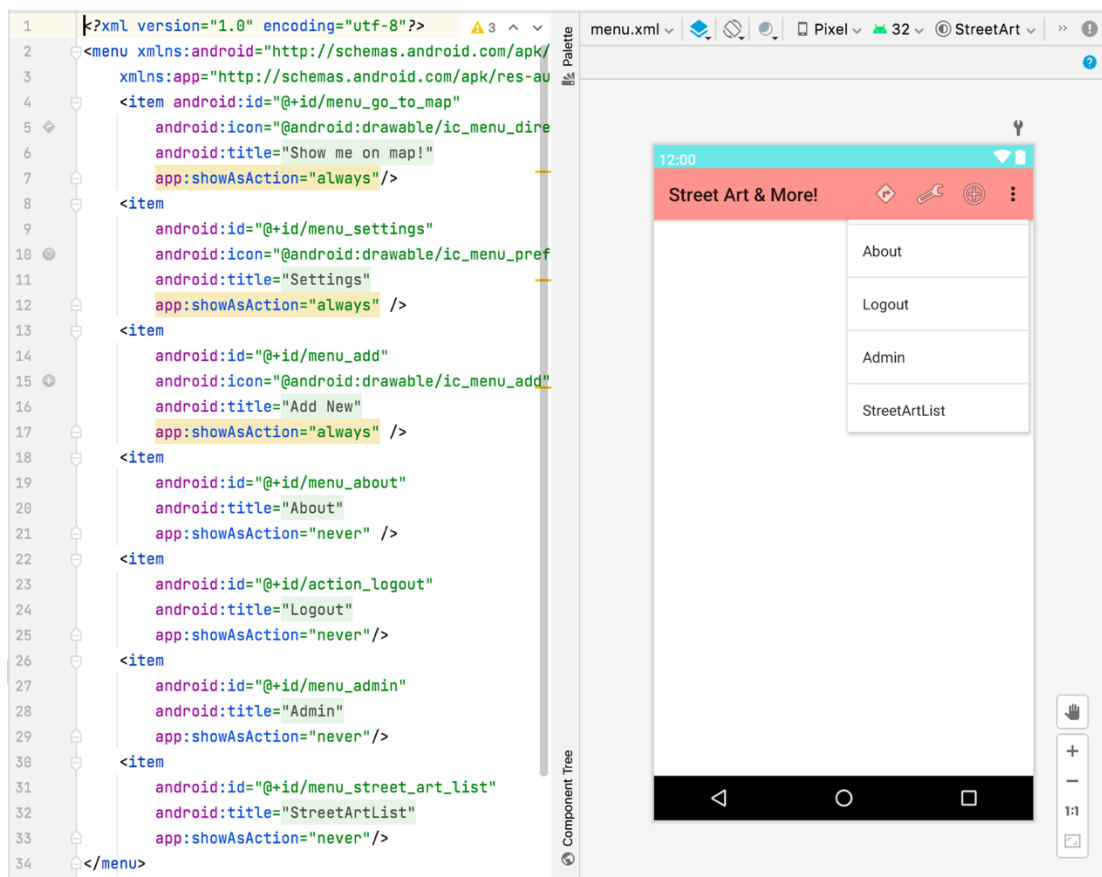
Στη συνέχεια ο χρήστης θα οδηγηθεί στο κεντρικό Activity της εφαρμογής όπου θα υπάρχει μία λίστα όπως αναφέραμε παραπάνω με τα Street Art έργα. Θα πρέπει επίσης να κατασκευάσουμε ένα μενού από το οποίο ο χρήστης θα μπορεί να μεταφερθεί σε άλλα Activity της εφαρμογής, όπως ο χάρτης που θα περιέχει markers με τις τοποθεσίες των Street Art έργων, τα settings, μία επιλογή about ώστε να του δώσουμε πληροφορίες για εμάς και για την εφαρμογή και η επιλογή logout.

Κατά την υλοποίηση προέκυψε άλλη μία ιδέα για μία επιπλέον λειτουργία. Τη λειτουργία Add News Street Art (AddNewActivity). Σκεφτήκαμε πώς θα ήταν πολύ χρήσιμο αρχικά για τον admin να μπορεί να προσθέσει ένα έργο Street Art απευθείας με κινητό του. Φωτογραφίζοντας το και περνώντας τις κατάλληλες πληροφορίες να είναι σε θέση να το αποθηκεύει στη βάση δεδομένων.

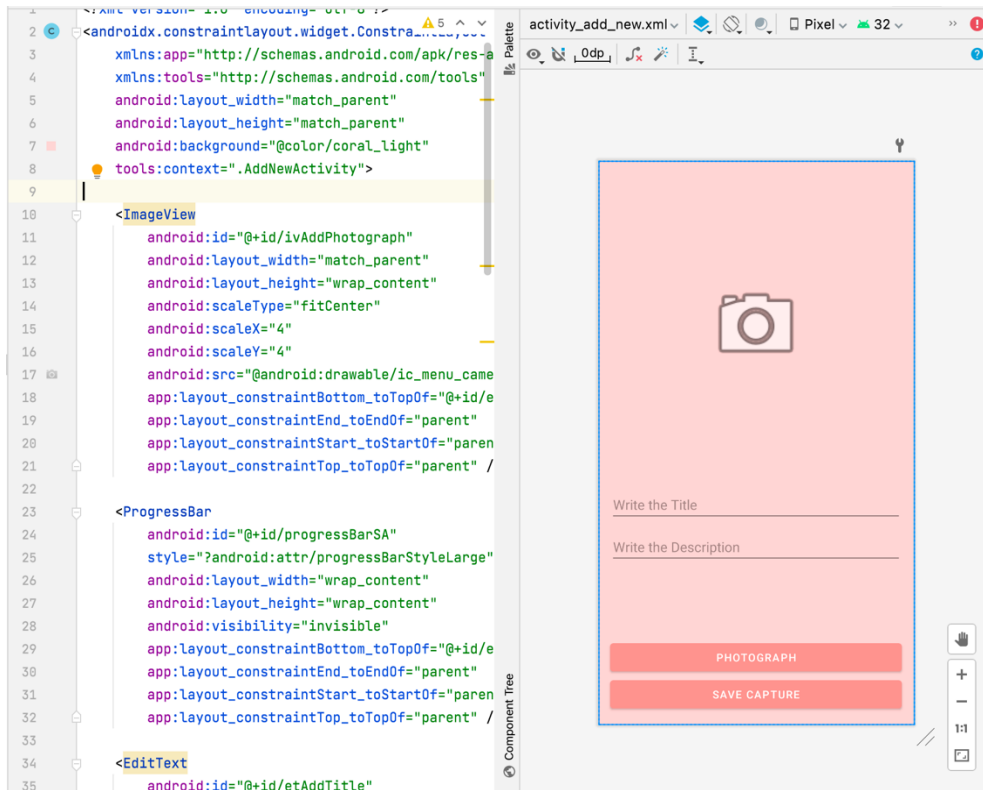
Εν τέλει αυτή η λειτουργία απλοποίησε σημαντικά τον τρόπο αποθήκευσης των έργων στην βάση καθώς όταν ο χρήστης την χρησιμοποιήσει, ταυτόχρονα εκτελούνται κατάλληλες μέθοδοι για να βρει η εφαρμογή τις συντεταγμένες της συσκευής και με την χρήση του Geocoder εντοπίζει την διεύθυνση μας. Ακολουθώντας κατά την λήψη της εικόνας χρησιμοποιώντας τη μέθοδο της firebase MLkit αντλούμε και τα labels από αυτή ώστε να είμαστε σε θέση να τα αποθηκεύσουμε στη βάση.

Μετά τη δημιουργία της συμπεράναμε ότι είναι αρκετά διαδραστική για τον χρήστη οπότε επιλέξαμε να μην την περιορίσουμε στον admin. Τέλος κατασκευάσαμε και ένα ακόμα Activity στο οποίο έχει πρόσβαση μόνο ο admin της εφαρμογής ώστε να μπορεί να ελέγχει μέσω δυο SeekBar πόσοι χρήστες εγγράφηκαν και πόσα Street Art έργα προστέθηκαν για το διάστημα που θα ορίσει από 0 έως 30 ημέρες.

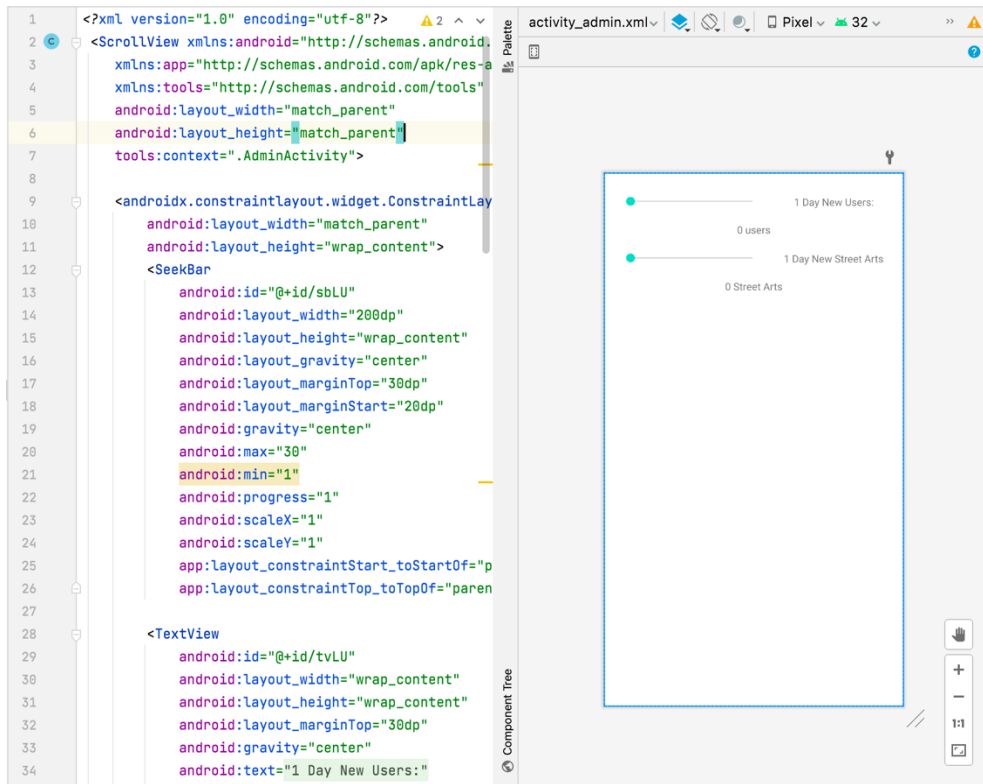
Ακολουθούν οι εικόνες των activity που περιγράψαμε:



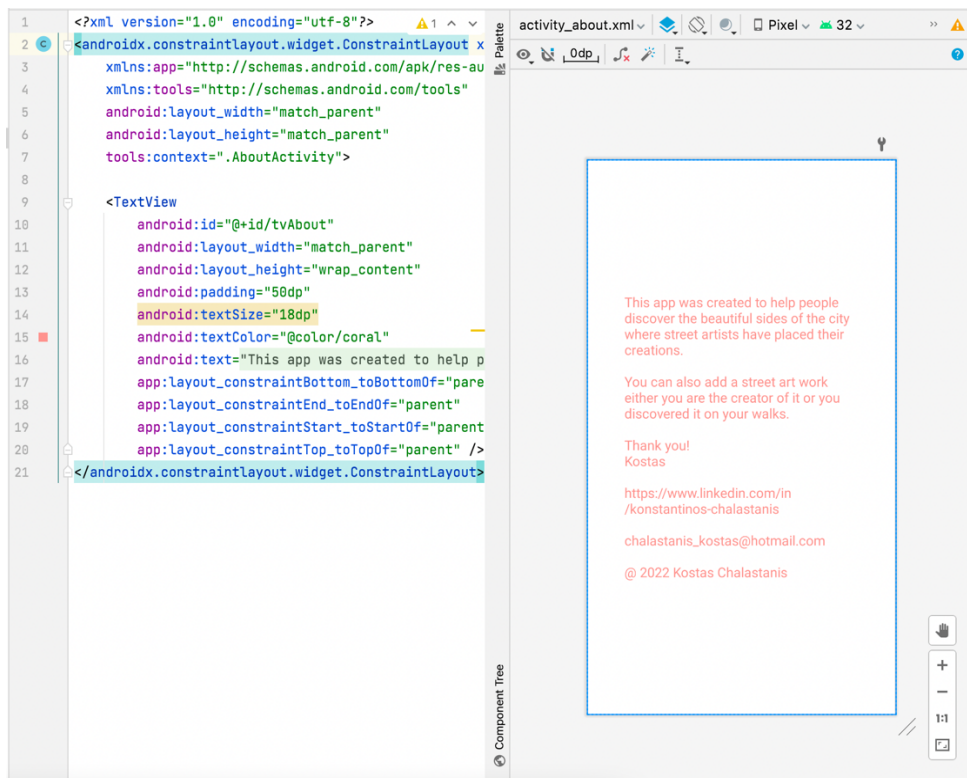
Εικόνα 21. Android Studio: Το menu.xml



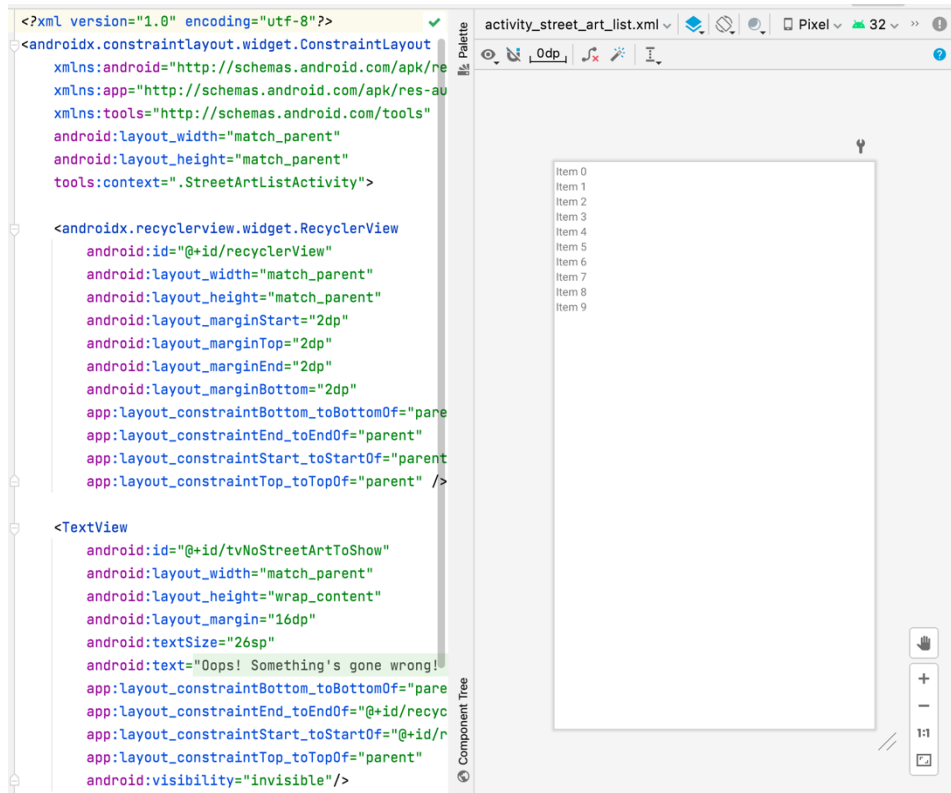
Εικόνα 22. Android Studio: Το activity_add_new.xml



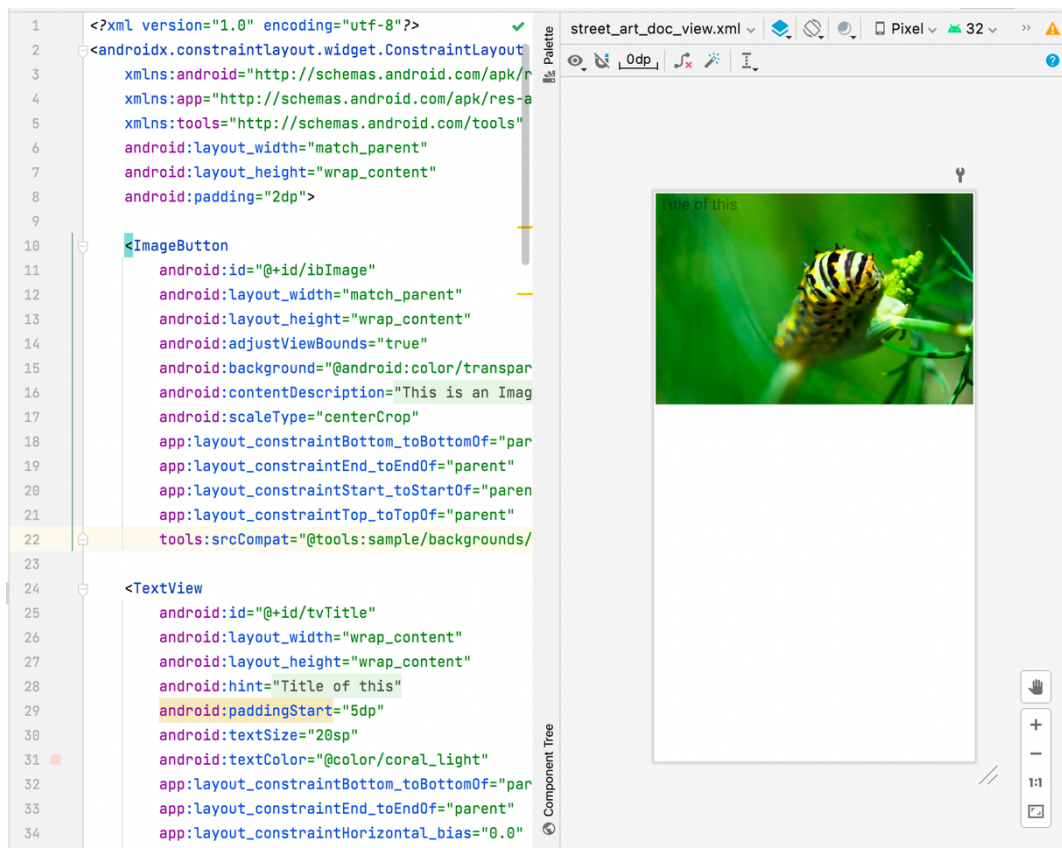
Εικόνα 23. Android Studio: Το activity_admin με δυο SeekBar.



Εικόνα 24 Android Studio: Το activity_about.xml.



Εικόνα 25. Android Studio: Το activity_street_art_list.xml. Περιέχει ένα recycler view



Εικόνα 26 Android Studio: Το street_art_doc_view.xml. Το μοντέλο για την λίστα του StreetArtListActivity.java

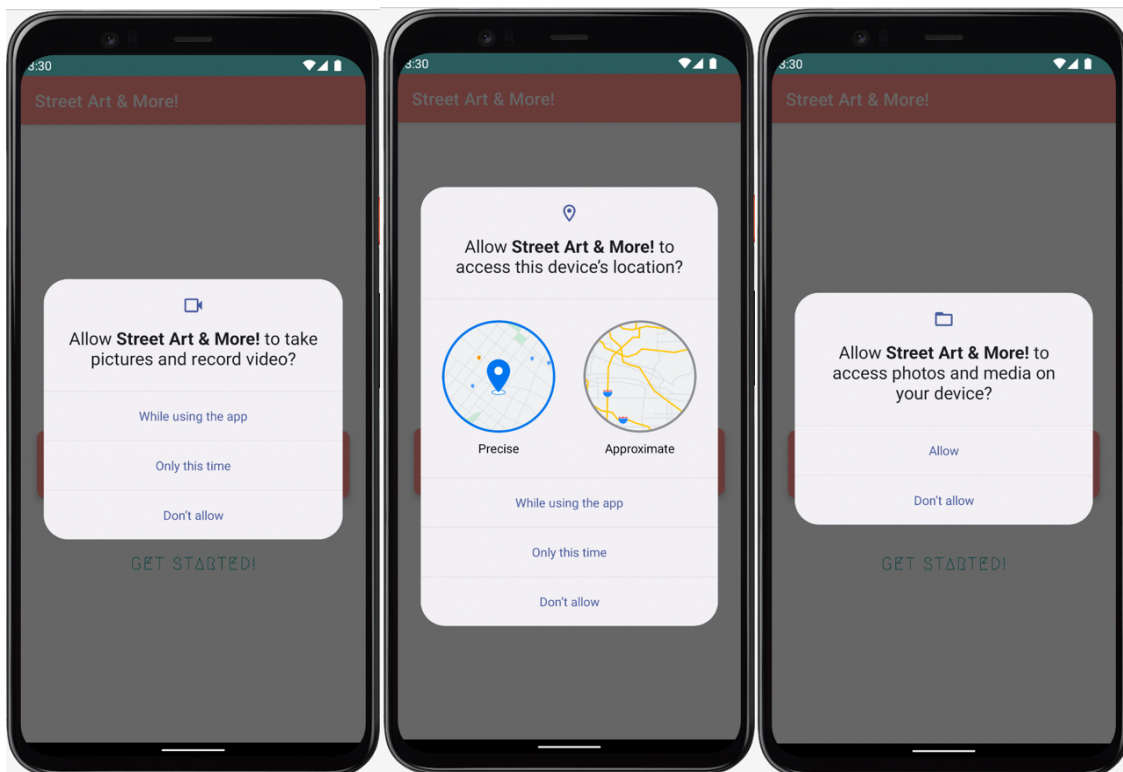
Όπως βλέπουμε στην εικόνα 24 και 25 το StreetArtListActivity μοιάζει πολύ απλό καθώς έχει έναν recycler view και το αντίστοιχο μοντέλο που χρειάζεται ώστε μέσω του adapter να προστεθούν στη λίστα τα έργα Street Art. Πάραυτα παρουσιάζει ενδιαφέρον προγραμματιστικά καθώς χρησιμοποιείται άλλο ένα κενό μοντέλο ώστε μέσω ενός Alert dialog να εμφανίσουμε την εικόνα σε μεγαλύτερο μέγεθος και να κατεθύνουμε τον χρήστη στην αντίστοιχη τοποθεσία εφόσον το επιλέξει. Ταυτόχρονα γίνεται κλήση σε μία ακόμη κλάση που έχουμε κατασκευάσει, την οποία την χρησιμοποιούμε ώστε να μπορούμε να έχουμε πρόσβαση στην τοποθεσία της συσκευής ακόμη και όταν ο χρήστης έχει εγκαταλείψει τη χρήση της. Με αυτό τον τρόπο Πετυχαίνουμε να ειδοποιήσουμε τον χρήστη στην περίπτωση που βρεθεί κοντά σε κάποιο Street Art έργο. Επίσης έγινε η χρήση της βιβλιοθήκης Picasso για την λήψη και προβολή των εικόνων στη λίστα και στο Alert dialog, οπότε και χρειάστηκε να δηλωθεί στα dependencies του build.gradle καθώς αποτελεί εξωτερικό πακέτο εντολών. Επιπλέον για τις λειτουργίες που κατασκευάσαμε παραπάνω απαραίτητο ήταν να δηλωθούν στο manifest τα permissions για την χρήση της camera όπως και τα δικαιώματα ανάγνωσης και εγγραφής στην συσκευή.

4 Προσομοίωση της εφαρμογής

Σε αυτό το κεφάλαιο θα προσπαθήσουμε να αναδείξουμε τις λειτουργίες της εφαρμογής. Θα προσπαθήσουμε να πάρουμε την θέση του χρήστη και να δείξουμε όσο πιο αναλυτικά μπορούμε τα στάδια και τα σενάρια τα οποία ο χρήστης μπορεί να ακολουθήσει. Αρχικά αν επιλέξει να χρησιμοποιήσει την εφαρμογή ως επισκέπτης, όπου και του προσφέρουμε περιορισμένες δυνατότητες, ακολούθως αφότου δημιουργήσει νέο λογαριασμό και εισέλθει στη λίστα των έργων Street Art και τέλος θα προσπαθήσουμε μέσω του user interface να εκτελέσουμε όλες τις επιλογές που έχει. Επίσης θα συνδεθούμε και ως admin user για να δείξουμε το Activity στο οποίο ο διαχειριστής μπορεί να δει ένα σύνολο πληροφοριών χωρίς να χρειαστεί να επισκεφτεί την βάση μέσω browser.

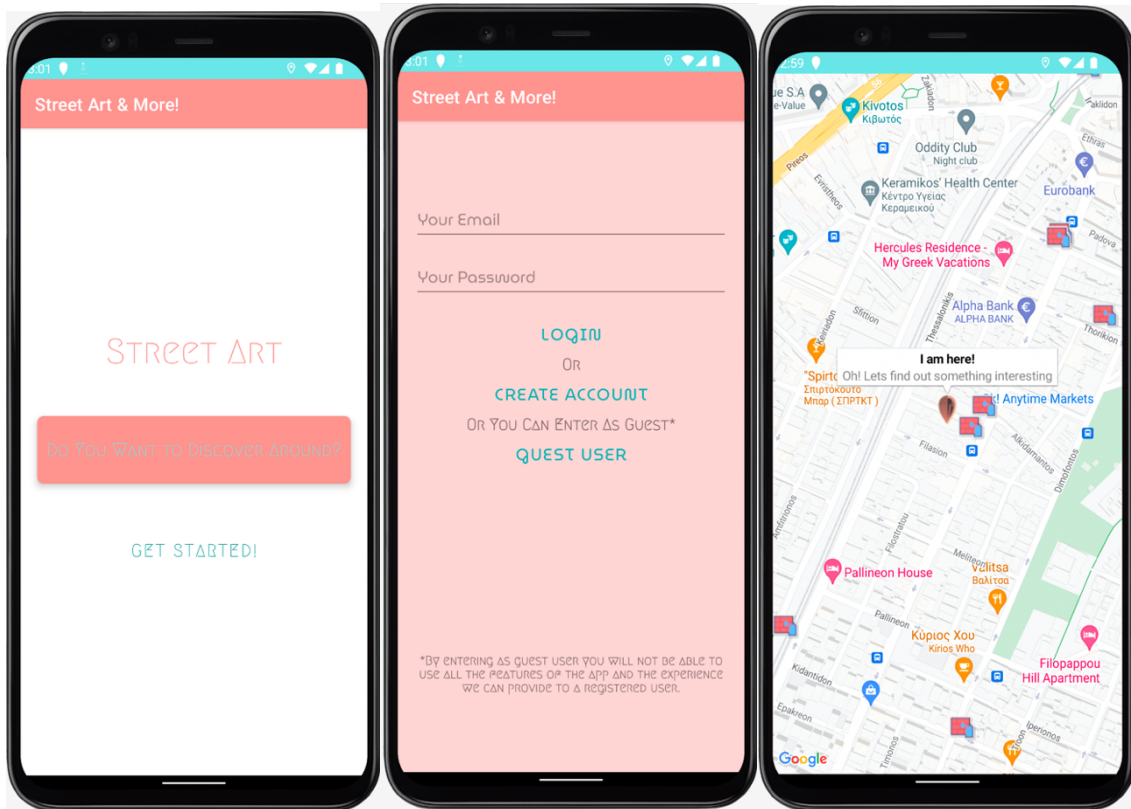
4.1 Εκκίνηση της εφαρμογής για πρώτη φορά

Κατά την εκκίνηση της εφαρμογής για πρώτη φορά θα αναλυθούν τα παράθυρα στα οποία ο χρήστης καλείται να αποδειχθεί ή όχι τα permissions που ζητά η εφαρμογή.



Εικόνα 27. Η εφαρμογή κατά την εκκίνηση θα ζητήσει δικαιώματα από το χρήστη.

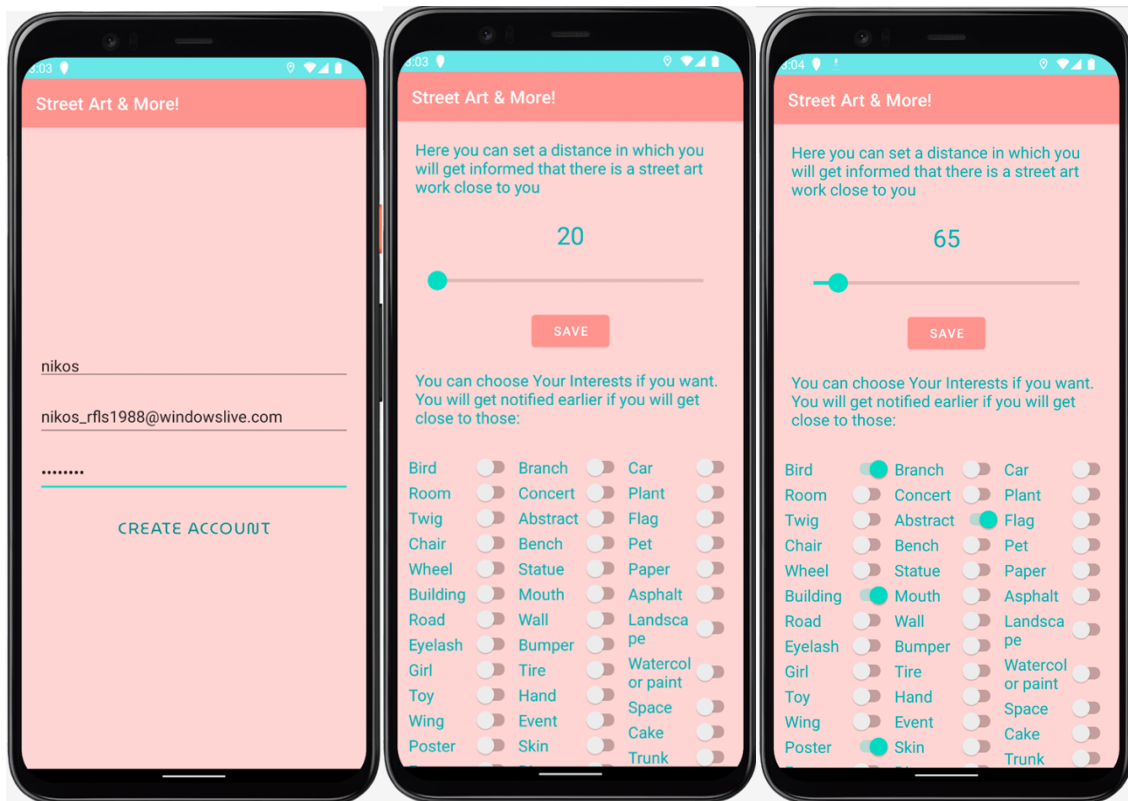
Στη συνέχεια αφότου ο χρήστης επιλέξει τα δικαιώματα που θα δώσει στην εφαρμογή θα εμφανιστεί το activity_main.xml το οποίο τον καλωσορίζει στην εφαρμογή. Υπάρχει μόνο μία επιλογή, το κουμπί get started. Πατώντας το get started ο χρήστης θα οδηγηθεί στη σελίδα activity_login.xml. Λόγω του ότι εξετάζουμε το ενδεχόμενο που χρησιμοποιεί την εφαρμογή για πρώτη φορά ο χρήστης έχει δύο επιλογές. Η μία είναι να εισέλθει ως guest user και η δεύτερη να επιλέξει create new account. Αρχικά θα δούμε την επιλογή που θα εισέλθει ως guest user.



Εικόνα 28. Ο χρήστης επιλέγει να εισέλθει ως guest

Παρατηρούμε ότι κατευθύνεται στην σελίδα χαρτών όπου έχει πρόσβαση στους markers των έργων Street Art και με αυτόν τον τρόπο μπορεί να τα αναζητήσει manual αλλά όπως βλέπουμε δεν υπάρχει κάποιο μενού με το οποίο μπορεί να χρησιμοποιήσει άλλες λειτουργίες. Έτσι Αν τον ευχαριστήσει η συγκεκριμένη εμπειρία ίσως επιστρέψει στην αρχική ώστε να κάνει λογαριασμό.

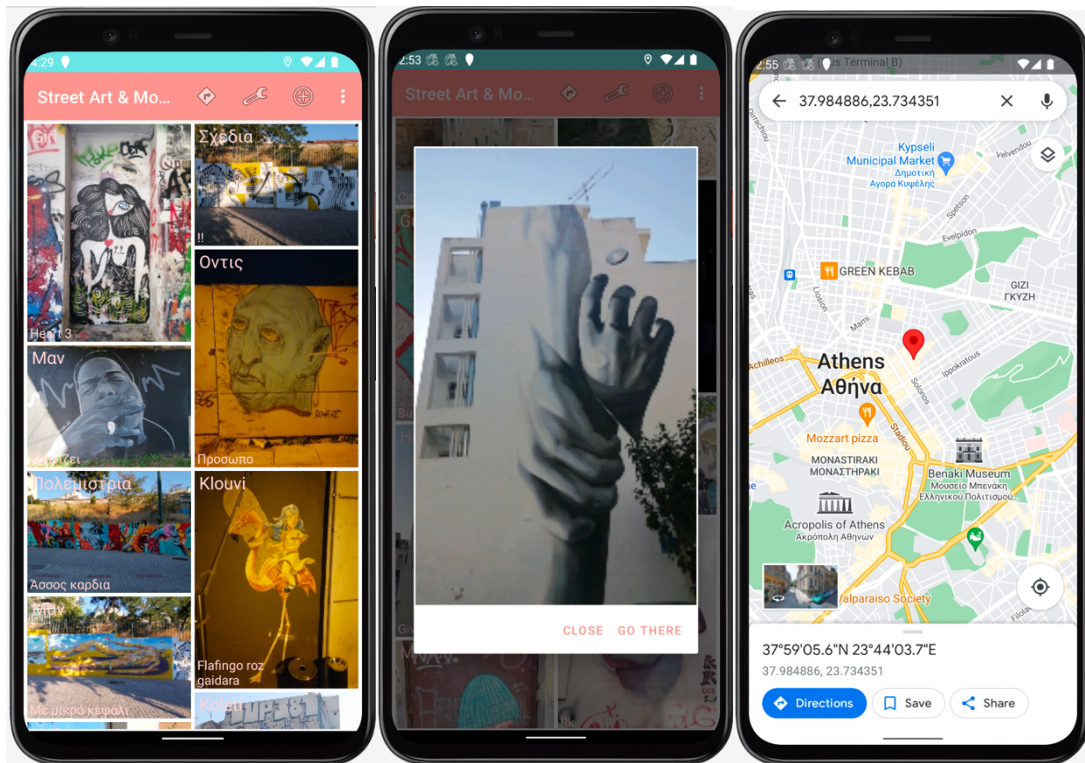
Στις επόμενες εικόνες θα δούμε την διαδικασία που θα κάνει ο χρήστης εάν επιλέξει να πατήσει το κουμπί create account. Θα κατευθυνθεί στη σελίδα δημιουργίας λογαριασμού και εάν ακολουθήσει τη διαδικασία σωστά θα οδηγηθεί στο activity_settings.xml



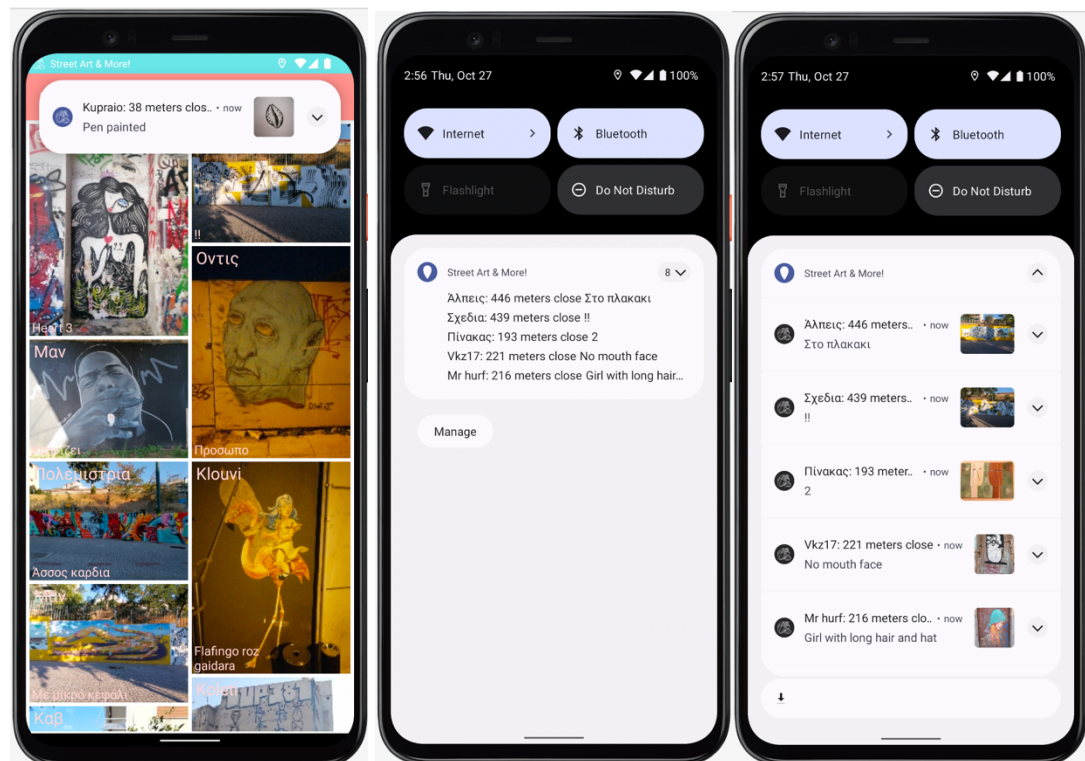
Εικόνα 29. Εγγραφή νέου χρήστη και ρύθμιση των αρχικών του settings

Σε αυτό το σημείο θα χρειαστεί να αποθηκεύσει τα Settings του πατώντας το button save και ακολούθως θα οδηγηθεί στο `activity_street_art_list`. Σε αυτό το activity μπορεί να δει το σύνολο των έργων κάνοντας scroll down και μία από τις επιλογές του είναι να πατήσει πάνω σε ένα πλακίδιο για να το δει σε μεγάλο μέγεθος. Δεύτερη επιλογή εάν το επιθυμεί είναι στο πλακίδιο που έχει αναδυθεί να πατήσει την επιλογή GO THERE. πατώντας το Θα μεταφερθεί στην εφαρμογή χαρτών της Google όπου μπορεί να πάρει οδηγίες κατεύθυνσης ώστε να το παρατηρήσει στην οδό που βρίσκεται.

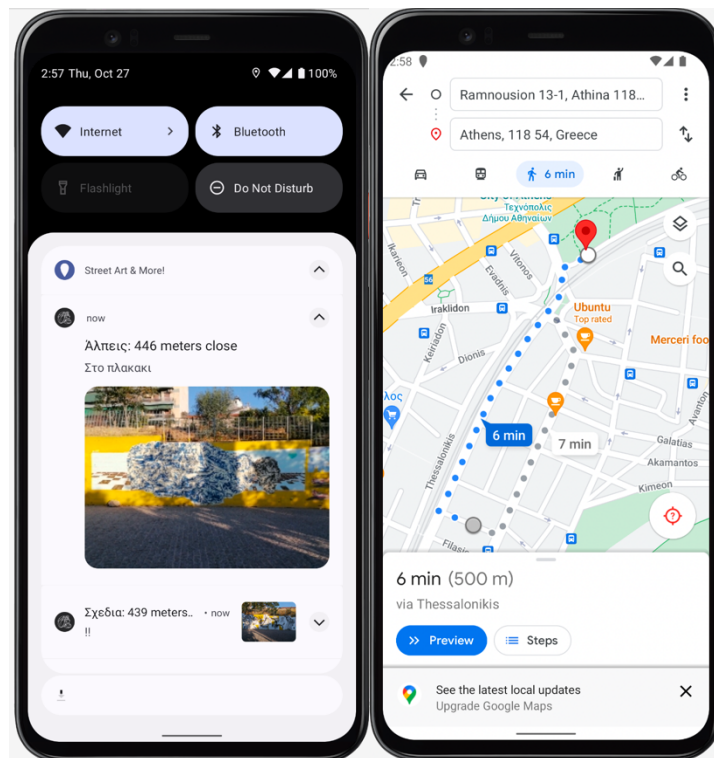
Παρακάτω στην εικόνα 31 βλέπουμε τα notifications που έχουν εμφανιστεί καθώς στο παρασκήνιο τρέχει το location service, στο οποίο γίνεται έλεγχος για την θέση την οποία βρίσκεται ο χρήστης. Στην περίπτωση που υπάρχουν έργα Street Art για προβολή σε απόσταση μικρότερη ή ίση με το distance ή με το $distance * 2$ για τα labels που έχει θέσει στα settings θα τον ειδοποιήσει.



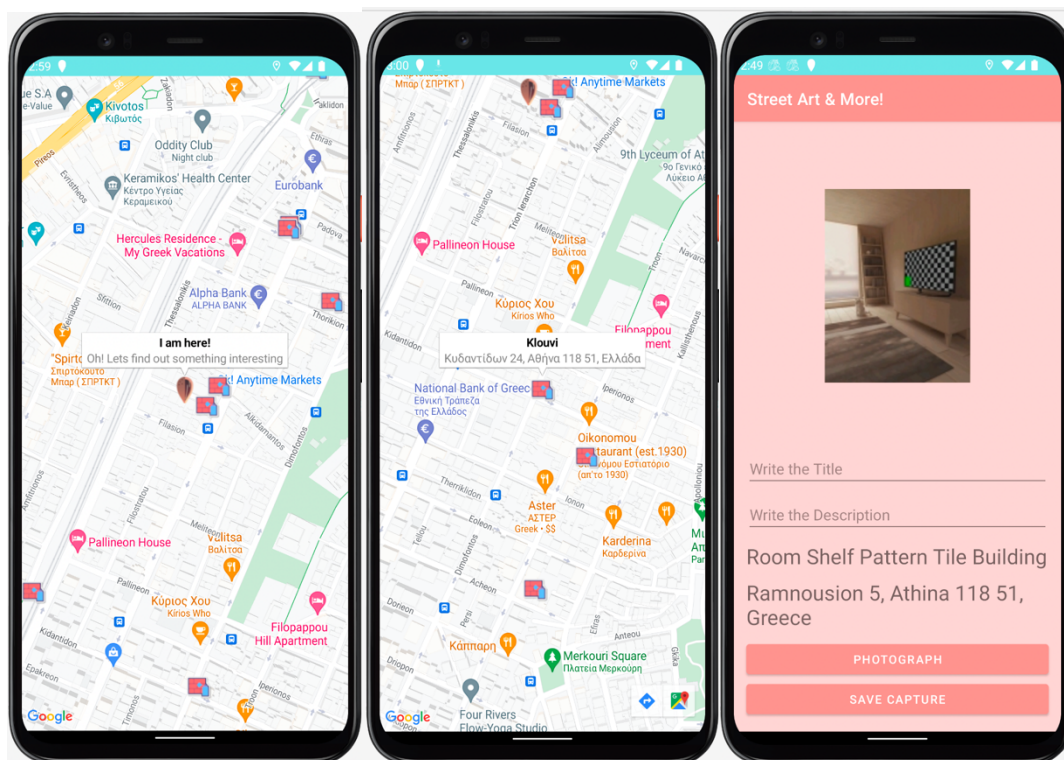
Εικόνα 30. Επιλογή ενός πλακιδίου από τη λίστα.



Εικόνα 31. Εισερχόμενες ειδοποιήσεις και προβολή αυτών.



Εικόνα 32. Ανάπτυξη notification και λήψη οδηγιών μόλις ο χρήστης κάνει κλικ πάνω του

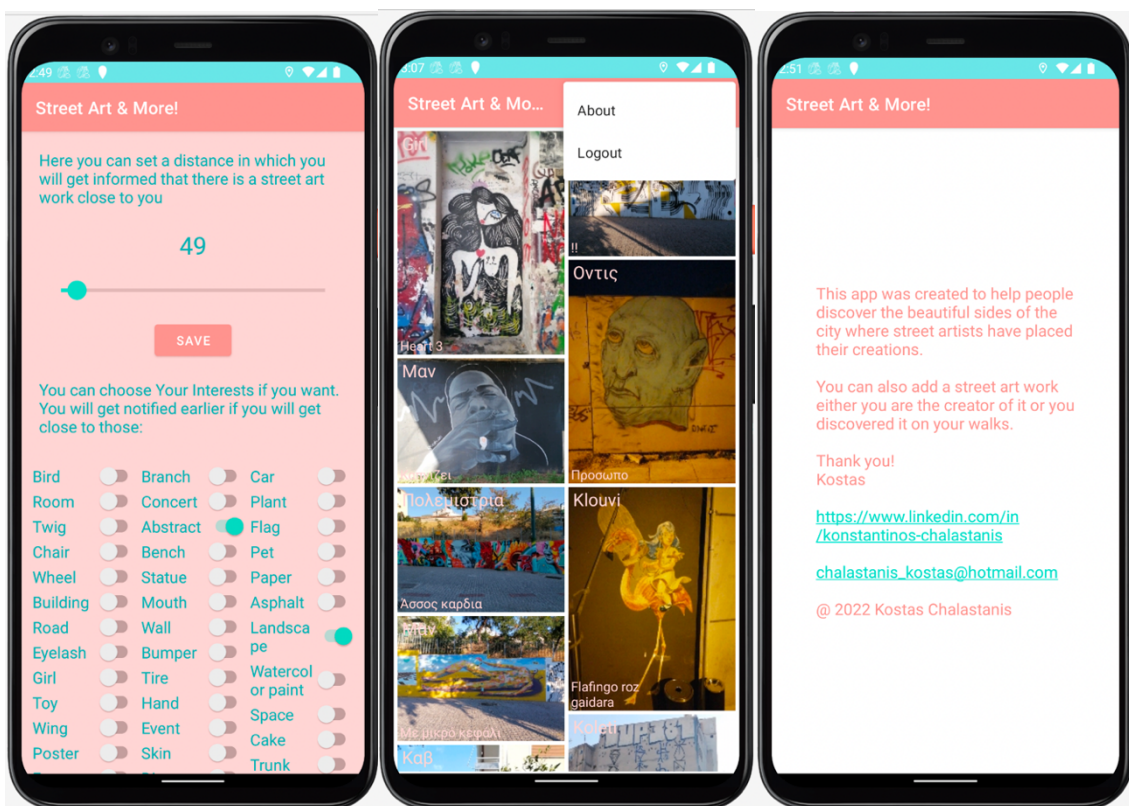


Εικόνα 33. Επιλογές μενού Map και Add New Street Art

Ανάπτυξη android εφαρμογής Street Art οδηγός πόλης

Στην εικόνα 33 βλέπουμε στην οθόνη το αποτέλεσμα της επιλογής του χρήστη να πατήσει το πλήκτρο του χάρτη. Αρχικά εστιάζει στο σημείο που βρίσκεται και στην δεύτερη εικόνα με τον χάρτη φαίνεται το μήνυμα μετά από την επιλογή ενός marker που έχει κάνει ο χρήστης. Όπως φαίνεται κάτω δεξιά στην δεύτερη εικόνα του χάρτη της εικόνας 33 όταν ο χρήστης επιλέγει έναν marker ακολούθως μπορεί να πατήσει το κουμπί directions.

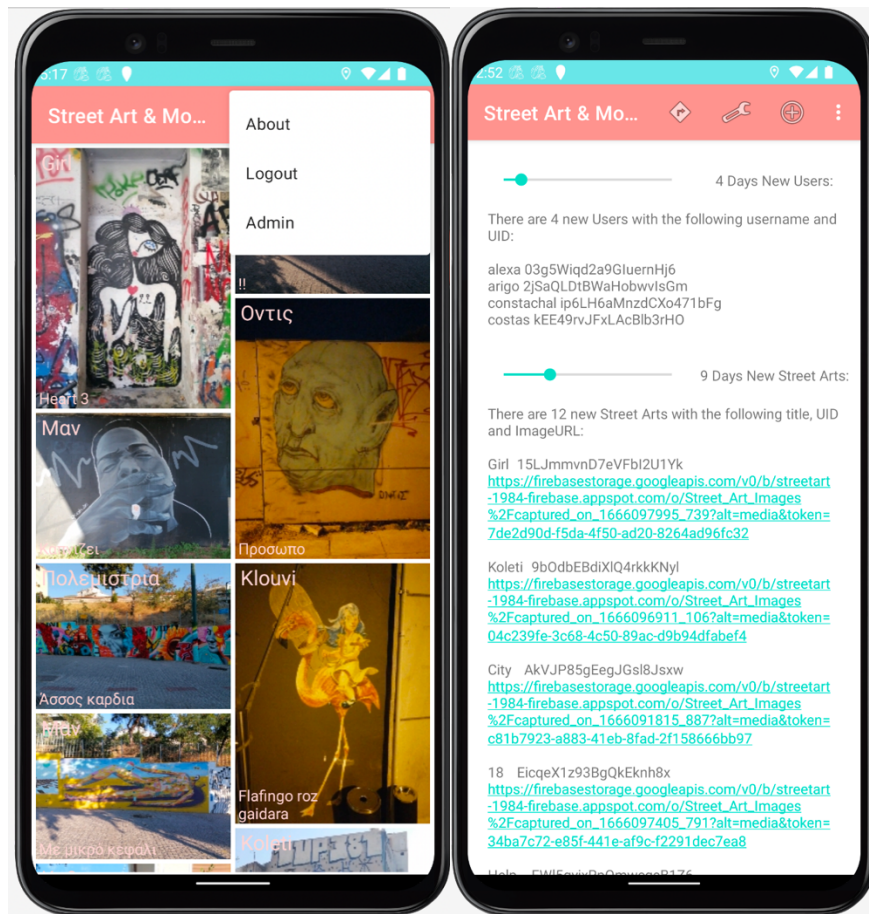
Στην τρίτη εικόνα της εικόνας 33 βλέπουμε το αποτέλεσμα της επιλογής του χρήστη να πατήσει το πλήκτρο της πρόσθεσης νέου Street Art. Όπως θα παρατηρήσουμε στο τελευταίο textview έχει εμφανιστεί η διεύθυνση του χρήστη η οποία έχει αποδοθεί μέσω των location services και του geocoder όπως αναφέραμε νωρίτερα. Ακριβώς πάνω από τη διεύθυνση βλέπουμε στο αντίστοιχο textview τα labels που αντλήθηκαν από την φωτογραφία που τράβηξε ο χρήστης. Στη συνέχεια για να ανεβάσει επιτυχώς στη βάση το συγκεκριμένο Street Art θα πρέπει να συμπληρώσει και τα δύο παιδιά title και description.



Εικόνα 34. Επιλογές Settings και About από το μενού.

4.2 Είσοδος στην εφαρμογή ως admin

Τέλος Όσον αφορά το χρήστη βλέπουμε στην εικόνα 34 το αποτέλεσμα της επιλογής το button settings την ανάδυση του μενού και την επιλογή about παρακάτω στην εικόνα 35 θα δούμε πώς διαμορφώνεται το μενού για τον admin user όπου προστίθεται μία ακόμη επιλογή admin. Πατώντας ο χρήστης την επιλογή admin έχει πρόσβαση σε μία σελίδα στην οποία μπορεί να κάνει έλεγχο για το πόσοι χρήστες έχουν προστεθεί στην βάση δεδομένων χρηστών της εφαρμογής σε ένα διάστημα από 0 έως 30 τελευταίες ημέρες. Αντίστοιχα μπορεί να κάνει τον ανάλογο έλεγχο και για τα Street Art έργα τα οποία πιθανόν έχουν προστεθεί για ανάλογο διάστημα.



Εικόνα 35. Μενού admin user και activity_admin

5 Μέθοδοι και μέρη του κώδικα

Σε αυτό το κεφάλαιο θα παρουσιάσουμε μέρη του κώδικα και μεθόδους που χρησιμοποιήσαμε σε διάφορα σημεία της εφαρμογής. Σκοπός είναι μέσα από την παρουσίασή τους να αναδείξουμε τον τρόπο με τον οποίον εργαστήκαμε ώστε να επιτύχουμε διαφορές λειτουργίες της εφαρμογής με στόχο την κατανόησή τους.

5.1 Εκκίνηση της εφαρμογής και permissions

Όπως αναφέραμε παραπάνω κατά την εκκίνηση της εφαρμογής ο χρήστης καλείται να δώσει δικαιώματα στην εφαρμογή ώστε να χρησιμοποιήσει διάφορες λειτουργίες της συσκευής, να έχει πρόσβαση στο ίντερνετ και να μπορεί να εντοπίσει τις συντεταγμένες του χρήστη

```
private String[] PERMISSIONS;

// Ενέργειες για την αποδοχή των permission από το χρήστη
PERMISSIONS = new String[] {
    Manifest.permission.ACCESS_FINE_LOCATION,
    Manifest.permission.ACCESS_COARSE_LOCATION,
    Manifest.permission.READ_EXTERNAL_STORAGE,
    Manifest.permission.WRITE_EXTERNAL_STORAGE,
    Manifest.permission.CAMERA
};

if (!hasPermissions(context: this, PERMISSIONS)) {
    ActivityCompat.requestPermissions(activity: MainActivity.this, PERMISSIONS, requestCode: 1);
}

// Ελεγχος αν έχουμε Permissions
private boolean hasPermissions(Context context, String... PERMISSIONS) {
    if (context != null && PERMISSIONS != null) {
        for (String permission: PERMISSIONS){
            if (ActivityCompat.checkSelfPermission(context, permission) != PackageManager.PERMISSION_GRANTED) {
                return false;
            }
        }
    }
    return true;
}

// Μεθοδος ελέγχου αν έχουν αποδοθεί τα permissions
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (requestCode == 1) {
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(this, "Fine Location Permission is granted", Toast.LENGTH_SHORT).show();
            Log.d(tag: "MainActivity", msg: "Fine Location Permission is granted");
        } else {
            Toast.makeText(this, "Fine Location Permission is denied", Toast.LENGTH_SHORT).show();
            Log.d(tag: "MainActivity", msg: "Fine Location Permission is denied");
        }
    }
}
```

Εικόνα 36. Μεθοδολογία για την απόδοση δικαιωμάτων στην εφαρμογή

5.2 Δημιουργία νέου χρήστη και σύνδεση με τη βάση δεδομένων

Παρακάτω θα δούμε την κλάση στην StreetArtUser που δημιουργήσαμε μέσα σε ένα νέο φάκελο utilities, την οποία θα χρησιμοποιήσουμε ώστε με τις μεθόδους getInstance, getters, setters να έχουμε πρόσβαση στα στοιχεία του χρήστη σε όλο το app. Ακολούθως στο CreateNewAccountActivity θα δούμε πως κάνουμε σύνδεση με τον authentication server και τη βάση δεδομένων όπου δημιουργούμε καινούργια collection για τους users, έναν καινούριο user και τον αποθηκεύουμε στη βάση σε ένα νέο document.

```
// Κατασκευάσαμε αυτή τη κλάση με extend Application ώστε να περάσουμε τα στοιχεία του χρήστη από
// την createAccountActivity. Έτσι με getInstance θα μπορούμε να τα χρησιμοποιούμε σε όλο το app
public class StreetArtUser extends Application {

    private String username;
    private String userId;
    private String distance;
    private List<String> userLabels;
    private Timestamp dateAdded;

    public static StreetArtUser instance;
    public static StreetArtUser getInstance(){
        if(instance==null)
            instance = new StreetArtUser();
        return instance;
    }

    public StreetArtUser(){
    }

    public List<String> getUserLabels() { return userLabels; }

    public void setUserLabels(List<String> userLabels) { this.userLabels = userLabels; }

    public String getUsername() { return username; }

    public void setUsername(String username) { this.username = username; }

    public String getUserId() { return userId; }

    public void setUserId(String userId) { this.userId = userId; }

    public String getDistance() { return distance; }

    public void setDistance(String distance) { this.distance = distance; }

    public Timestamp getDateAdded() { return dateAdded; }

    public void setDateAdded(Timestamp dateAdded) { this.dateAdded = dateAdded; }
}
```

Εικόνα 37. Android Studio: Κώδικας της κλάσης StreetArtUser

```

private FirebaseAuth firebaseAuth;
private FirebaseAuth.AuthStateListener authStateListener;
private FirebaseUser currentUser;

// Firestore connection
private FirebaseFirestore db = FirebaseFirestore.getInstance();
// δημιουργώ το reference για την Users collection
private CollectionReference usersReference = db.collection( collectionPath: "Users");

```

Εικόνα 38. Android Studio: Δηλώσεις στην CreateAccountActivity για την σύνδεση με τον authentication server, την Firestore και η αναφορά της collection

```

    firebaseAuth = FirebaseAuth.getInstance();

    authStateListener = new FirebaseAuth.AuthStateListener() {
        @Override
        public void onAuthStateChanged(@NonNull FirebaseAuth firebaseAuth) {
            currentUser = firebaseAuth.getCurrentUser();
        }
    };

    // Όταν θα πατήσω το create καλώ τη μέθοδο createUserEmailAccount για να δημιουργήσω τον user
    btnCreateActDone.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(!TextUtils.isEmpty(actvEmailCreate.getText().toString())
                && !TextUtils.isEmpty(etPasswordCreate.getText().toString())
                && !TextUtils.isEmpty(etUsernameCreate.getText().toString())){

                String email = actvEmailCreate.getText().toString().trim();
                String password = etPasswordCreate.getText().toString().trim();
                String username = etUsernameCreate.getText().toString().trim();

                createUserEmailAccount(email,password,username);
            }
        }
    });
}

// Μεθοδος για να δημιουργήσω τον νέο user
private void createUserEmailAccount(String email, String password, String username){
    if(!TextUtils.isEmpty(email) && !TextUtils.isEmpty(password) && !TextUtils.isEmpty(username)){
        progressBar.setVisibility(View.VISIBLE);
        // Εδώ θα χρησιμοποιήσω τη μέθοδο createUserWithEmailAndPassword
        // της firebase για να δημιουργήσω τον user με τα credentials απο τα editTexts
        // που έχει συμπληρώσει προσθέτωνας τον addOnCompleteListener

```

```

firebaseAuth.createUserWithEmailAndPassword(email,password)
    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            // Μου επιστρέφει ένα task όπου ελέγχω αν είναι successful
            if(task.isSuccessful()){
                // Αν το task είναι successful μεταφέρω τον user στο SettingsActivity και ταυτόχρονα
                // στην firebase καταχωρώ τον νέο authenticated user στη Users Collection

                currentUser = firebaseAuth.getCurrentUser();
                assert currentUser != null;
                String currentUserId = currentUser.getId();
                List<String> firstEmptyLabelList = new ArrayList<>();
                String startDistance = "20";
                Timestamp addedDate = Timestamp.now();
                // Create a user Map so we can create a user in the users collection
                Map<String, Object> userObj = new HashMap<>();
                userObj.put( k: "userId", currentUserId);
                userObj.put( k: "username", username);
                userObj.put( k: "userLabels", firstEmptyLabelList);
                userObj.put( k: "distance", startDistance);
                userObj.put( k: "dateAdded", addedDate);
                // Save to firestore database

                usersReference.add(userObj).addOnSuccessListener(new OnSuccessListener<DocumentReference>() {
                    @Override
                    public void onSuccess(DocumentReference documentReference) {
                        // Εδώ αντλώ τα στοιχεία του user που μόλις έχω αποθηκεύσει στη βάση και σετάρω τον
                        // StreetArtUser ώστε να μπορώ να χρησιμοποιώ τα στοιχεία σε άλλα σημεία της εφαρμογής.
                        documentReference.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
                            @Override
                            public void onComplete(@NonNull Task<DocumentSnapshot> task) {
                                if(task.getResult().exists()){
                                    progressBar.setVisibility(View.INVISIBLE);

                                    String name = task.getResult().getString( field: "username");
                                    String userId = task.getResult().getString( field: "userId");
                                    String distance = task.getResult().getString( field: "distance");
                                    Timestamp dateAdded = task.getResult().getTimestamp( field: "dateAdded");
                                    List<String> userLabels = (List<String>) task.getResult().get("userLabels");

                                    StreetArtUser streetArtUser = StreetArtUser.getInstance();
                                    streetArtUser.setUserId(userId);
                                    streetArtUser.setUsername(name);
                                    streetArtUser.setUserLabels(userLabels);
                                    streetArtUser.setDistance(distance);
                                    streetArtUser.setDateAdded(dateAdded);

                                    Intent intent = new Intent( packageContext: CreateAccountActivity.this, SettingsActivity.class);
                                    intent.putExtra( name: "username", name);
                                    intent.putExtra( name: "userId",userId);
                                    intent.putExtra( name: "distance",distance);
                                    intent.putStringArrayListExtra( name: "userLabels", (ArrayList<String>) userLabels);
                                    startActivity(intent);
                                    CreateAccountActivity.this.finish();
                                }else{
                                    progressBar.setVisibility(View.INVISIBLE);
                                    Log.d(TAG, msg: "OnFailure: Something went wrong! Check 129 Line");
                                }
                            }
                        });
                    }
                });
            }
        }
    });

```

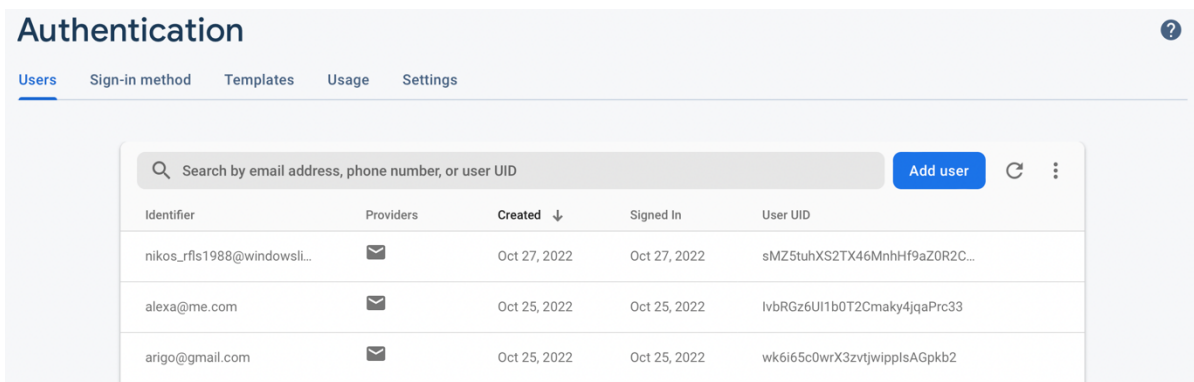


```

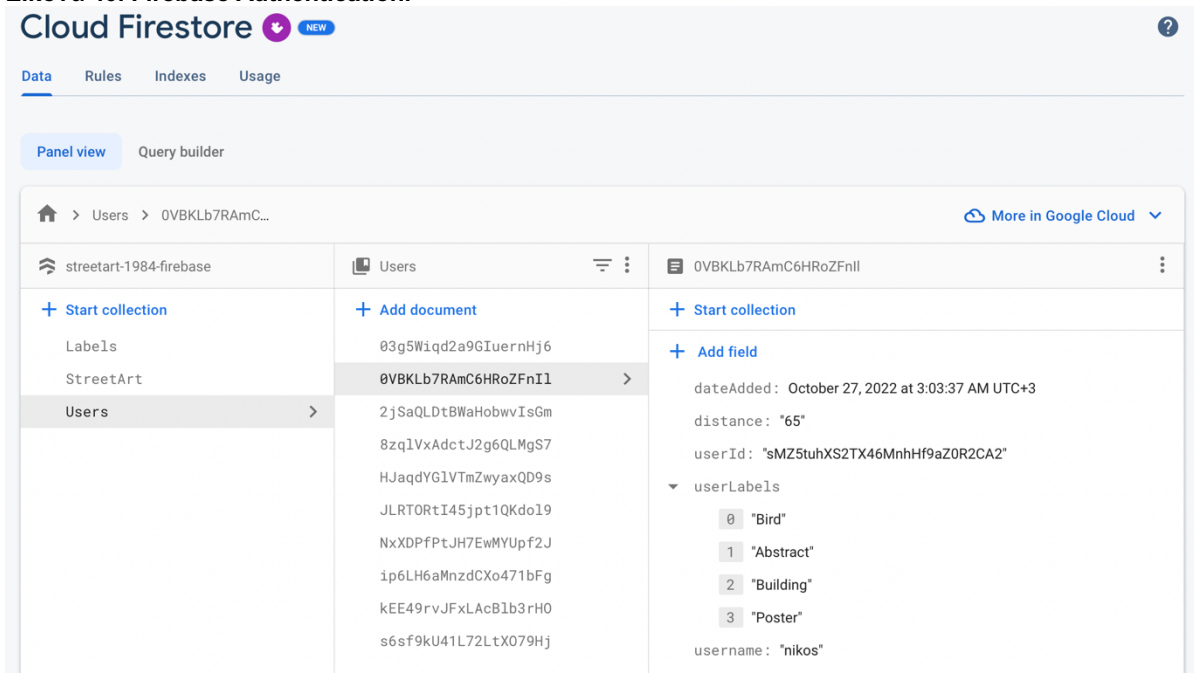
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            Log.d(TAG, msg: "On Failure: Check line 166 --> " + e);
        }
    });
} else {
    Log.d(TAG, msg: "OnFailure: Something went wrong! Check 103 Line");
}
}
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        Log.d(TAG, msg: "On Failure: Check Line 100 --> " + e);
    }
});
} else {
    //There is an empty field an actions needed
    Toast.makeText(context: CreateAccountActivity.this, text: "One or more inputs are empty!!!", Toast.LENGTH_LONG).show();
}
}
}

```

Εικόνα 39. Android Studio: Κώδικας της CreateAccountActivity με τη διαδικασία δημιουργίας του user



Εικόνα 40. Firebase Authentication.



Εικόνα 41. Firebase Cloud Firestore

5.3 Υπολογισμός της απόστασης μεταξύ ενός Street Art έργου

Όπως αναφέραμε σε προηγούμενο σημείο αυτής της εργασίας όταν ο χρήστης έχει πλέον δημιουργηθεί μεταφέρεται στη λίστα προβολής των έργων street art ταυτόχρονα και ως αποτέλεσμα τις κλήσης από το StreetArtActivity της κλάσης LocationService που έχουμε κατασκευάσει, στο παρασκήνιο γίνεται ένας έλεγχος εύρεσης των συντεταγμένων που βρισκόμαστε. Ταυτόχρονα κάθε φορά που γίνεται η λήψη των συντεταγμένων μας από το service, πραγματοποιείται ένας υπολογισμός μεταξύ της θέσης μας και των συντεταγμένων που βρίσκονται τα έργα StreetArt. Τότε με βάση το όριο που έχουμε θέσει στα settings, όταν θα βρεθούμε μέσα στην ακτίνα μεταξύ του έργου και της θέσης μας θα ενεργοποιηθεί η αποστολή notification με την εικόνα του έργου και πληροφορίες. Μόλις πατήσουμε πάνω στο notification θα μας μεταφέρει στο google maps με οδηγίες. Παρακάτω φαίνονται οι συναρτήσεις που χρησιμοποιούνται για τον υπολογισμό της απόστασης μεταξύ του έργου και της θέσης μας.

```
// Three functions bellow needed to calculate the distance between two locations
private double distance(double lat1, double long1, double lat2, double long2) {
    //Calculate Longitude difference
    double longDiff = long1 - long2;
    //Calculate distance
    double distance = Math.sin(deg2rad(lat1))
        * Math.sin(deg2rad(lat2))
        + Math.cos(deg2rad(lat1))
        * Math.cos(deg2rad(lat2))
        * Math.cos(deg2rad(longDiff));
    distance = Math.acos(distance);
    //Convert distance radian to degree
    distance = rad2deg(distance);
    //Distance in miles
    distance = distance * 60 * 1.1515;
    //Distance in kilometers
    distance = distance * 1.609344;
    //Distance in meters
    distance = distance * 1000;
    return distance;
}

//Convert radian to degree
private double rad2deg(double distance) { return (distance * 180.0 / Math.PI); }

//Convert degree to radian
private double deg2rad (double lat1) { return (lat1*Math.PI/180.0); }
```

Εικόνα 42. Συναρτήσεις υπολογισμού μεταξύ δύο συντεταγμένων.

5.4 Χρήση κλάσης Geocoder

Στην εικόνα 43 βλέπουμε πως μετατρέπονται οι συντεταγμένες μας σε διεύθυνση και πώς τοποθετείται σε ένα text view.

```

@Override
public void onLocationChanged(@NonNull Location location) {
    myLatitude = location.getLatitude();
    myLongitude = location.getLongitude();

    try {
        Geocoder geocoder = new Geocoder(context: AddNewActivity.this, Locale.getDefault());
        List<Address> addresses = geocoder.getFromLocation(myLatitude, myLongitude, maxResults: 1);
        String address = addresses.get(0).getAddressLine(index: 0);
        tvAddAddress.setText(address);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

Εικόνα 43. Χρήση της κλάσης Geocoder.

5.5 Χρήση ML kit labeler

Στον παρακάτω κώδικα φαίνεται η μέθοδος που χρησιμοποιούμε ώστε να αντλήσουμε τις ετικέτες που περιγράφουν το περιεχόμενο της φωτογραφίας μας χρησιμοποιώντας το interface της Google Firebase Machine Learning kit ImageLabeler.

```

ImageLabeler labeler;

// Methodos gia na adlisoume ta labels apo tin fwtografia
private void processImage() {
    labeler.process(inputImage)
        .addOnSuccessListener(new OnSuccessListener<List<ImageLabel>>() {
            @Override
            public void onSuccess(@NonNull List<ImageLabel> imageLabels) {
                String result="";
                labels = new ArrayList<>();
                for(ImageLabel label: imageLabels){
                    result = result + label.getText() + " ";
                    labels.add(label.getText().toString().trim());
                }
                Log.d(tag: "Add", msg: "Add: " + labels);
                tvAddLabels.setText(result);
                saveLabels();
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Log.d(TAG, msg: "onFailure: " + e.getMessage());
            }
        });
}
}

```

Εικόνα 44. Χρήση του ImageLabeler από το interface ML kit

6 Συμπεράσματα και μελλοντικές επεκτάσεις

Μετά την ολοκλήρωση της παραπάνω εργασίας μπορούμε να βγάλουμε κάποια περιορισμένα συμπεράσματα από την χρήση της εφαρμογής καθώς δεν έχει κυκλοφορήσει στο ευρύ κοινό αλλά μόνο σε έναν κλειστό κύκλο. Οι εντυπώσεις που λαμβάνουμε από τις αντιδράσεις των ανθρώπων που χρησιμοποίησαν την εφαρμογή είναι αρκετά θετικές. Γνώμη μας είναι πώς μία εφαρμογή τέτοιου είδους θα είχε θετικό αντίκτυπο στην κουλτούρα που αναπτύσσεται γύρω από τα έργα Street Art και πιστεύουμε πως θα προωθούσε δράσεις που θα χρωμάτιζαν τους γκρίζους τοίχους των πόλεων μας.

Υπάρχουν αρκετές σκέψεις για επεκτάσεις που μπορούν να γίνουν πάνω στην εφαρμογή με πρώτη την αυθεντικοποίηση των χρηστών μέσω των social media. Στη συνέχεια μπορούν να γίνουν οι απαραίτητες ρυθμίσεις για να ανέβει η εφαρμογή στο Play Store και πέρα από τις δικές μας σκέψεις είναι βέβαιο πώς κατά τη χρήση της εφαρμογής θα ανακύψουν νέες ιδέες για επεκτάσεις που μπορούν να εφαρμοστούν.

7 Βιβλιογραφικές Πηγές

1. Android OS: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))
2. Android Studio: https://en.wikipedia.org/wiki/Android_Studio
3. Android Studio: <https://developer.android.com/studio/intro>
4. Android NDK: <https://developer.android.com/ndk>
5. Java: https://www.java.com/en/download/help/whatis_java.html
6. Java: <https://el.wikipedia.org/wiki/Java>
7. Firebase: <https://firebase.google.com/>
8. Firebase Auth: <https://firebase.google.com/products/auth>
9. Firebase Firestore: <https://firebase.google.com/docs/firestore>
10. Firebase Storage: <https://firebase.google.com/docs/storage>
11. Firebase ML Kit: <https://firebase.google.com/docs/ml-kit>
12. UML Wikipedia: https://en.wikipedia.org/wiki/Unified_Modeling_Language
13. Google Maps Platform: <https://developers.google.com/maps/documentation/android-sdk/get-api-key>
14. Android Manifest: <https://developer.android.com/guide/topics/manifest/manifest-intro>
15. Picasso Library: <https://square.github.io/picasso/>
16. Android Location: <https://developer.android.com/training/location/background>
17. Android Location: <https://developer.android.com/training/location/request-updates>
18. Stackoverflow: <https://stackoverflow.com>
19. ML Kit Image Labeler: <https://firebase.google.com/docs/ml-kit/label-images>
20. Github: <https://github.com>