



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ: ΣΥΓΧΡΟΝΑ ΘΕΜΑΤΑ
ΤΕΧΝΟΛΟΓΙΑΣ ΛΟΓΙΣΜΙΚΟΥ - ΛΟΓΙΣΜΙΚΟ ΓΙΑ ΚΙΝΗΤΕΣ
ΣΥΣΚΕΥΕΣ**

Τελική Εργασία Μαθήματος

Επί των φοιτητών: Μόσχου Δημήτριου - Π18209

Σιάτρα Απόστολου – Π18215

Πετρίδη Αχιλλέα – Π18211

Φύτρου Ευάγγελου – Π18220

Διδάσκοντες: Αλέπης Ευθύμιος

Πειραιάς, 2022

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ	3
1.1 Εκφώνηση εργασίας.....	3
1.2 Στόχοι της εργασίας	3
1.3 Ορισμός του προβλήματος προς ανάλυση.....	3
2. ΣΥΝΤΟΜΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ RUP.....	4
3. ΦΑΣΗ: ΕΝΑΡΞΗ (INCEPTION).....	6
3.1 Σύλληψη απαιτήσεων	6
3.2 Ανάλυση-Σχεδιασμός.....	6
4. ΦΑΣΗ: ΕΚΠΟΝΗΣΗ ΜΕΛΕΤΗΣ (ELABORATION).....	7
4.1 Ανάλυση-Σχεδιασμός.....	7
4.1.1 Διαγράμματα περιπτώσεων χρήσης (1 ^η έκδοση).....	7
4.1.2 Διαγράμματα τάξεων (1 ^η έκδοση)	7
4.2 Υλοποίηση- Έλεγχος	8
4.2.1 Υλοποίηση: 1 ^η εκτελέσιμη έκδοση.....	8
4.2.2 Αναφορά ελέγχου για την 1η εκτελέσιμη έκδοση	9
5. ΦΑΣΗ: ΚΑΤΑΣΚΕΥΗ (CONSTRUCTION).....	10
5.1 Ανάλυση-Σχεδιασμός.....	10
5.1.1 Διάγραμμα Περιπτώσεων Χρήσης- Use Case Diagram (2η έκδοση)	10
5.1.2 Διάγραμμα Τάξεων – Class Diagram (2η έκδοση).....	11
5.2 Υλοποίηση- Έλεγχος	13
5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση.....	13
5.2.2 Αναφορά για την τελική εκτελέσιμη έκδοση	13
6. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ	14
6.1 Σύντομη παρουσίαση του προγράμματος.....	14
6.2 Ανάλυση σεναρίων λειτουργίας	23

1. ΕΙΣΑΓΩΓΗ

1.1 Εκφώνηση εργασίας

Ειδικό θέμα εργασίας ομάδας 15

Η ομάδα σας καλείται να αναπτύξει μια εφαρμογή Android αναπαραγωγής μουσικής βάσει του προφίλ του χρήστη (προτιμήσεων) και βάσει της τοποθεσίας του. Ο χρήστης θα μπορεί να δηλώσει κάποιες βασικές προτιμήσεις και η εναλλαγή των μουσικών κομματιών θα γίνεται δυναμικά βάσει χαρακτηριστικών που θα έχουν αποθηκευτεί μαζί με τις διάφορες πληροφορίες των μουσικών αρχείων. Τα μουσικά αρχεία μπορούν να είναι αποθηκευμένα τοπικά στη συσκευή του χρήστη, είτε να βρίσκονται στο διαδίκτυο (προϋποθέτει πρόσβαση στο internet).

1.2 Στόχοι της εργασίας

Η παρούσα εργασία, πραγματεύεται την ανάλυση, σχεδίαση και κατασκευή μιας android εφαρμογής, η οποία αναπαράγει μουσικής βάσει του προφίλ του χρήστη (προτιμήσεων) και βάσει της τοποθεσίας του.

1.3 Ορισμός του προβλήματος προς ανάλυση

Το πρόβλημα που καλούμαστε να υλοποιήσουμε είναι μια android εφαρμογή η οποία θα λειτουργήσει ως music player. Υπάρχει ένα και μόνο είδος χρήστη ο οποίος πρέπει να δημιουργήσει τον λογαριασμό του.

2. ΣΥΝΤΟΜΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ RUP

Η διαδικασία Rational Unified Process είναι η διαδικασία που προτείνουν ο Booch, Rumbaugh και Jacobson για την ανάπτυξη λογισμικού.

Ο κύκλος ζωής λογισμικού προτείνεται να είναι επαναληπτικός. Η ανάπτυξη δηλαδή να προχωρεί σε μια σειρά επαναλήψεων μέχρι να εξελιχθεί το τελικό προϊόν.

Η διαδικασία Rational Unified Process αποτελείται από ένα σύνολο οδηγιών σχετικά με τις τεχνικές και οργανωτικές απόψεις της ανάπτυξης λογισμικού. Η διαδικασία αυτή αφορά κυρίως στην Ανάλυση Απαιτήσεων και στο Σχεδιασμό.

Ο κύκλος ζωής λογισμικού όπως προτείνεται από τη Rational Unified Process φαίνεται στο Σχήμα 2.1.

Η διαδικασία Rational Unified Process είναι δομημένη σε δύο διαστάσεις:

- 1) Χρόνο - Χωρισμός του κύκλου ζωής σε φάσεις και επαναλήψεις. 2) Τμήματα διαδικασίας - Καλά ορισμένες εργασίες.

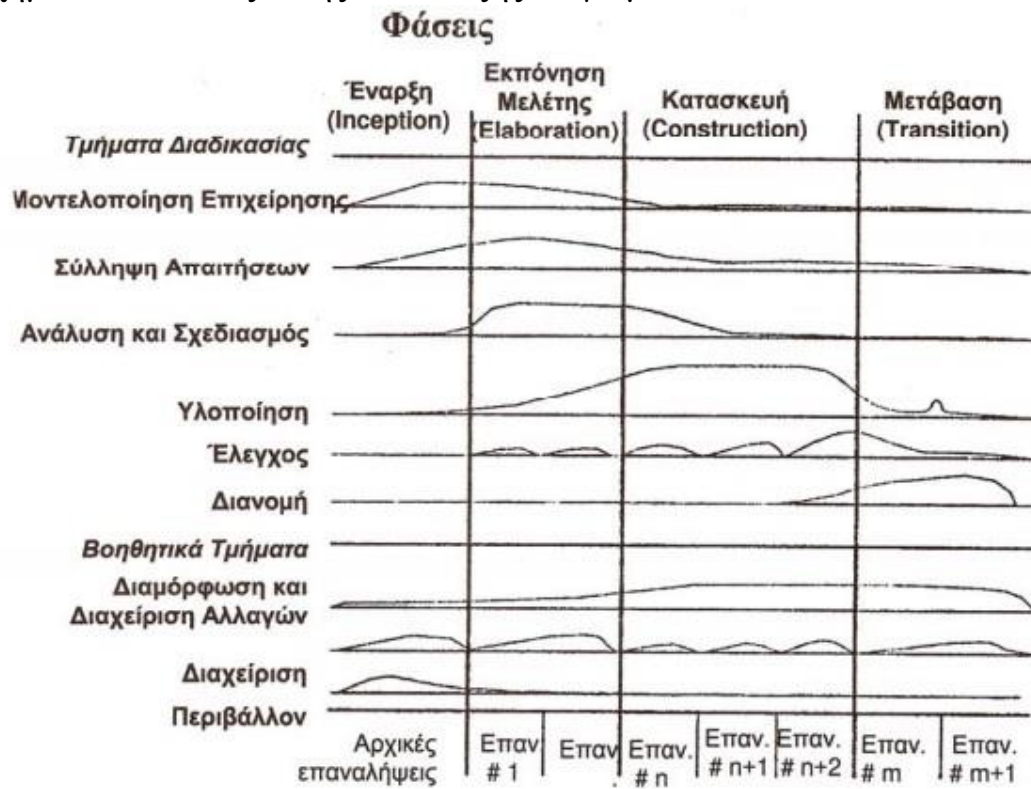
Η δόμηση ενός έργου σε σχέση με το χρόνο ακολουθεί τις εξής φάσεις που έχουν σχέση με το χρόνο:

- 1) Έναρξη (Inception): Καθορίζει την προοπτική του έργου.
- 2) Εκπόνηση μελέτης (Elaboration): Σχεδιασμός των απαιτούμενων δραστηριοτήτων και πόρων. Καθορισμός των χαρακτηριστικών και σχεδιασμός της αρχιτεκτονικής.
- 3) Κατασκευή (Construction): Ανάπτυξη του προϊόντος σε μια σειρά βηματικών επαναλήψεων.
- 4) Μετάβαση (Transition): Προμήθεια του προϊόντος στην κοινότητα χρηστών (παραγωγή, διανομή, εκπαίδευση).

Η δόμηση έργου σύμφωνα με τη διάσταση των τμημάτων διαδικασίας περιλαμβάνει τις ακόλουθες δραστηριότητες:

- 1) Σύλληψη απαιτήσεων (Requirements capture): Μια αφήγηση του τι πρέπει να κάνει το σύστημα.
- 2) Ανάλυση και σχεδιασμός (Analysis and design): Μια περιγραφή του πως θα υλοποιηθεί το σύστημα.
- 3) Υλοποίηση (Implementation): Η παραγωγή του κώδικα.
- 4) Έλεγχος (Test): Η επαλήθευση του συστήματος.

Σχήμα 2.1: Κύκλος Ζωής Ανάπτυξης Λογισμικού



3. ΦΑΣΗ: ΕΝΑΡΞΗ (INCEPTION)

3.1 Σύλληψη απαιτήσεων

Για την παρούσα εργασία, θα χρειαστούν: α) η γλώσσα προγραμματισμού Java, β) το προγραμματιστικό εργαλείο android studio, γ) η βάση δεδομένων Firebase, δ) η authentication λειτουργία της firebase καθώς και ε) η storage λειτουργία της.

3.2 Ανάλυση-Σχεδιασμός

Η εφαρμογή θα περιλαμβάνει μία κατηγορία χρηστών: τους χρήστες. Οι χρήστες θα μπορούν:

- Να δημιουργούν λογαριασμό
- Να συνδέονται και να αποσυνδέονται
- Να ακούν μουσική
- Να επιλέγουν το αγαπημένο τους είδος μουσικής
- Να επεξεργάζονται τις πληροφορίες του προφίλ τους

Η εφαρμογή θα μπορεί:

- Να έχει διγλωσσική πρόσβαση (Ελληνικά, Αγγλικά)
- Να χρησιμοποιεί αισθητήρες της συσκευής (μικρόφωνο, αισθητήρας εγγύτητας)

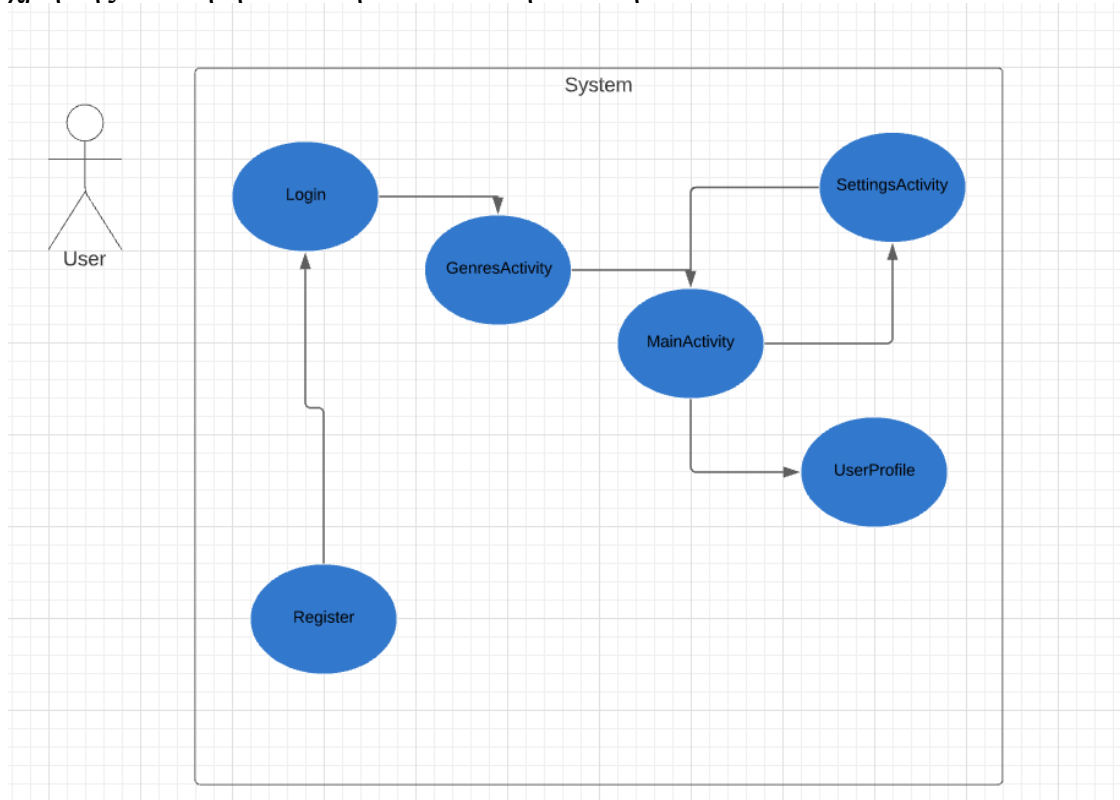
4. ΦΑΣΗ: ΕΚΠΟΝΗΣΗ ΜΕΛΕΤΗΣ **(ELABORATION)**

Η εκπόνηση μελέτης είναι ο σχεδιασμός των απαιτούμενων δραστηριοτήτων και πόρων και καθορισμός των χαρακτηριστικών και σχεδιασμός της αρχιτεκτονικής.

4.1 Ανάλυση-Σχεδιασμός

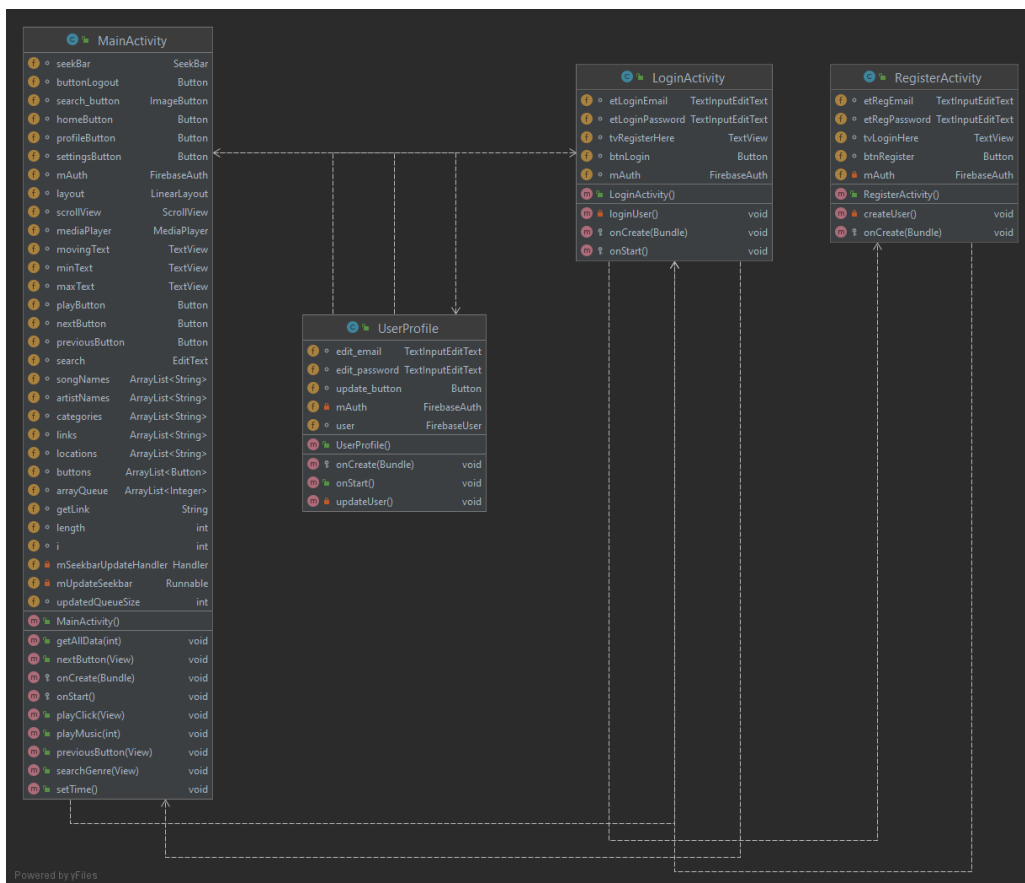
4.1.1 Διαγράμματα περιπτώσεων χρήσης (1^η έκδοση)

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος περιπτώσεων χρήσης. Αυτή η έκδοση δεν είναι η τελική.



4.1.2 Διαγράμματα τάξεων (1^η έκδοση)

Παρακάτω φαίνεται η πρώτη έκδοση του διαγράμματος τάξεων.



4.2 Υλοποίηση- Έλεγχος

4.2.1 Υλοποίηση: 1^η εκτελέσιμη έκδοση

Με γνώμονα τα προαναφερθέντα διαγράμματα πραγματοποιήθηκε η πρώτη εκτελέσιμη έκδοση της εφαρμογής, λαμβάνοντας υπόψιν και την ανάλυση απαιτήσεων.

Για την εύρυθμη λειτουργία της εφαρμογής, δημιουργήθηκε Βάση Δεδομένων μέσω της Firebase καθώς και το authentication των χρηστών έγινε μέσω αυτής.

Η Βάση Δεδομένων αποτελείται από δυο πίνακες:

- **songs:** ο πίνακας αυτός περιέχει τις πληροφορίες για όλα τα τραγούδια. Αποθηκεύει τις πληροφορίες για το όνομα του καλλιτέχνη “artist”, την κατηγορία του τραγουδιού “category”, το link του τραγουδιού που το συνδέει στην firebase storage “link”, την τοποθεσία για την οποία είναι διαθέσιμο το τραγούδι “location” καθώς και το όνομα του τραγουδιού “name”.
- **user_pref:** αυτός ο πίνακας χρησιμοποιείται για την αποθήκευση της προτιμήσεως του χρήστη πάνω στο είδος μουσικής.

Η υλοποίηση της διεπαφής με τον χρήστη έγινε με την δημιουργία .xml αρχείων. Τα αρχεία αυτά βρίσκονται μέσα στον φάκελο <layout>.

Αυτά τα αρχεία είναι:

- activity_login.xml: Χρησιμοποιείται για την σύνδεση των χρηστών στην εφαρμογή.
- activity_register.xml: Χρησιμοποιείται για την εγγραφή των χρηστών στην εφαρμογή.
- activity_user_profile.xml: Χρησιμοποιείται για την επεξεργασία του προφίλ του χρήστη.
- activity_main.xml: Εδώ εκτελούνται οι βασικές λειτουργίες της εφαρμογής. Παρουσιάζονται στον χρήστη όλα τα αποτελέσματα τα οποία μπορεί να φιλτράρει μέσω της αναζήτησης με είδος μουσικής και ακούσει μουσική.

4.2.2 Αναφορά ελέγχου για την 1η εκτελέσιμη έκδοση

Παρόλο που η πρώτη εκτελέσιμη έκδοση περιέχει όλα τα βασικά στοιχεία της εφαρμογής, χρειάζεται βελτίωση και προσθήκη επιπρόσθετων λειτουργιών.

Οι λειτουργίες που οφείλουν να υλοποιηθούν στην τελική εκτελέσιμη έκδοση είναι:

- Activity, για την δημιουργία προτιμήσεων του χρήστη.
- Activity, ρυθμίσεων για την έξοδο του χρήστη από την εφαρμογή και ενεργοποίηση αποτελεσμάτων βάση προτιμήσεων.

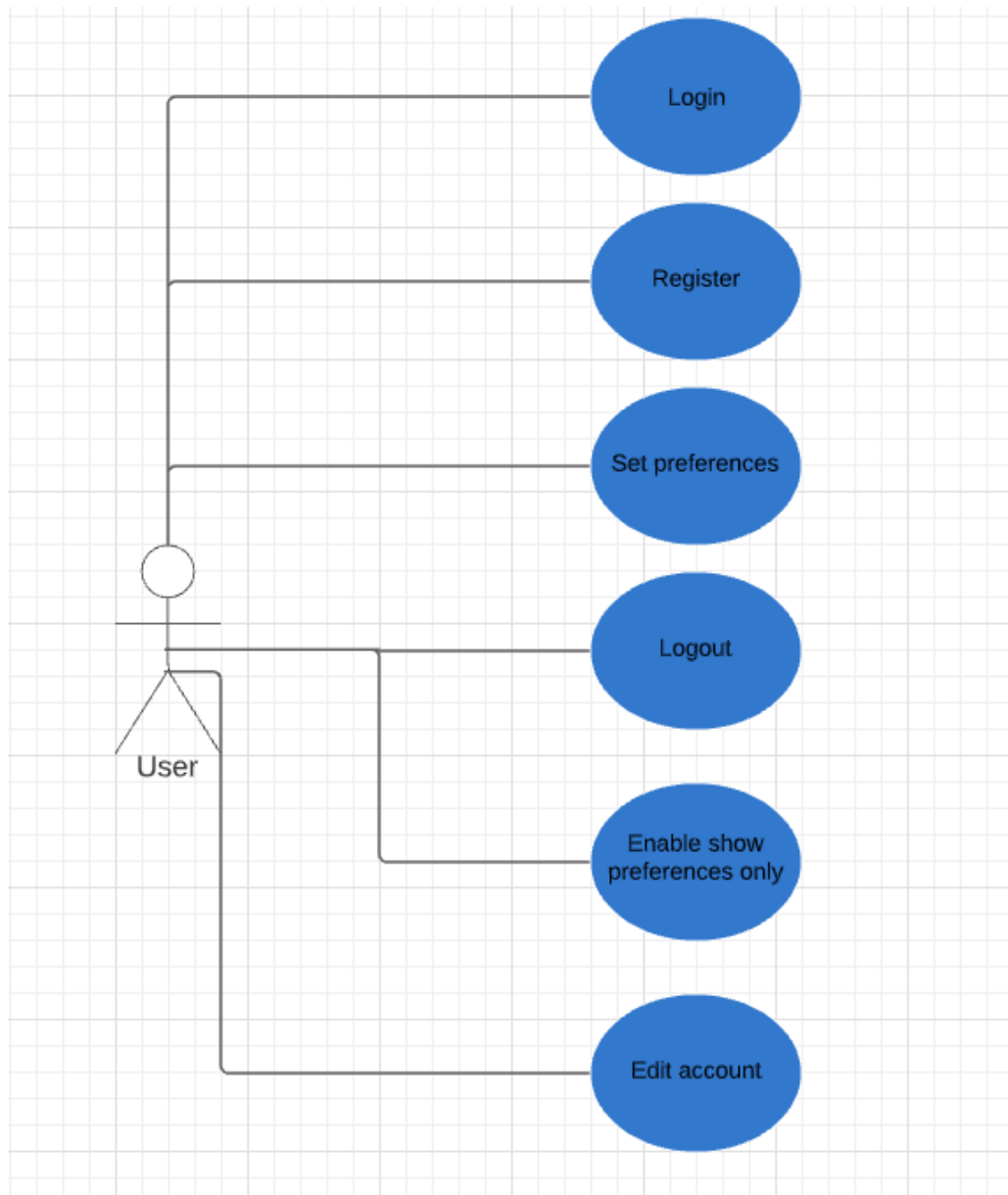
5. ΦΑΣΗ: ΚΑΤΑΣΚΕΥΗ (CONSTRUCTION)

5.1 Ανάλυση-Σχεδιασμός

Σε αυτή τη φάση της εφαρμογής, στόχος είναι η καλύτερη οργάνωση του κώδικα και η προσθήκη περαιτέρω ενεργειών που δεν κατάφεραν να υλοποιηθούν στην πρώτη εκτελέσιμη έκδοση.

5.1.1 Διάγραμμα Περιπτώσεων Χρήσης- Use Case Diagram (2η έκδοση)

Παρακάτω φαίνεται η 2η και τελική έκδοση του διαγράμματος περιπτώσεων χρήσης. Περιέχει όλες τις πιθανές ενέργειες που μπορούν να τελέσουν οι χρήστες τις εφαρμογής:



5.1.2 Διάγραμμα Τάξεων – Class Diagram (2η έκδοση)

Παρακάτω παρατίθεται η τελική έκδοση του διαγράμματος τάξεων:

MainActivity		
locman	LocationManager	
REQ_LOC_CODE	int	
REQUEST_CODE_SPEECH_INPUT	int	
seekBar	SeekBar	
search_button	ImageButton	
profileButton	Button	
settingsButton	Button	
mAuth	FirebaseAuth	
layout	LinearLayout	
scrollView	ScrollView	
mediaPlayer	MediaPlayer	
preference	String	
uid	String	
movingText	TextView	
minText	TextView	
maxText	TextView	
playButton	Button	
nextButton	Button	
previousButton	Button	
search	EditText	
songNames	ArrayList<String>	
artistNames	ArrayList<String>	
categories	ArrayList<String>	
links	ArrayList<String>	
locations	ArrayList<String>	
buttons	ArrayList<Button>	
arrayQueue	ArrayList<Integer>	
getLink	String	
country	String	
length	int	
i	int	
mySensorManager	SensorManager	
myProximitySensor	Sensor	
powerManager	PowerManager	
proximityFlag	boolean	
mSeekBarUpdateHandler	Handler	
mUpdateSeekBar	Runnable	
updatedQueueSize	int	
proximitySensorEventListener	SensorEventListener	
MainActivity()		
currentCountry(Location)	void	
enableLoc()	void	
getAllData(int)	void	
nextButton(View)	void	
onActivityResult(int, int, Intent)	void	
onCreate(Bundle)	void	
onLocationChanged(Location)	void	
onPause()	void	
onProviderDisabled(String)	void	
onProviderEnabled(String)	void	
onRequestPermissionsResult(int, String[], int[])	void	
onResume()	void	
onStart()	void	
playClick(View)	void	
playMusic(int)	void	
previousButton(View)	void	
searchGenre()	void	
setTime()	void	
settings(View)	void	
voiceRec(View)	void	
wakeLock	WakeLock?	

LoginActivity		
etLoginEmail	TextInputEditText	
etLoginPassword	TextInputEditText	
tvRegisterHere	TextView	
btnLogin	Button	
userIds	ArrayList<String>	
email	String	
password	String	
mAuth	FirebaseAuth	
LoginActivity()		
loginUser()	void	
onCreate(Bundle)	void	
onRequestPermissionsResult(int, String[], int[])	void	
onStart()	void	

RegisterActivity		
etRegEmail	TextInputEditText	
etRegPassword	TextInputEditText	
tvLoginHere	TextView	
btnRegister	Button	
mAuth	FirebaseAuth	
RegisterActivity()		
createUser()	void	
onCreate(Bundle)	void	

UserProfile		
edit_email	TextInputEditText	
edit_password	TextInputEditText	
update_button	Button	
mAuth	FirebaseAuth	
user	FirebaseUser	
UserProfile()		
onCreate(Bundle)	void	
onStart()	void	
updateUser()	void	

GenresActivity		
readyButton	Button	
genresSpinner	Spinner	
myRef	DatabaseReference	
mAuth	FirebaseAuth	
GenresActivity()		
onCreate(Bundle)	void	

SettingsActivity		
buttonLogout	Button	
mAuth	FirebaseAuth	
switch1	Switch	
switcher	Boolean	
SettingsActivity()		
logout(View)	void	
onCreate(Bundle)	void	

5.2 Υλοποίηση- Έλεγχος

5.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση

Με γνώμονα τις τελικές εκδόσεις των διαγραμμάτων, πραγματοποιήθηκε η τελική εκτελέσιμη έκδοση της εφαρμογής, λαμβάνοντας υπόψιν και την πρώτη εκτελέσιμη έκδοση.

Αλλαγές στην Βάση Δεδομένων δεν πραγματοποιήθηκαν και έτσι η δομή της παρέμεινε ίδια με αυτήν που παρουσιάστηκε στην πρώτη εκτελέσιμη έκδοση.

Στην υλοποίηση της διαδικτυακής διεπαφής προστέθηκαν τα παρακάτω καινούργια αρχεία xml:

- activity_genres.xml: Χρησιμοποιείται για την δημιουργία προτιμήσεων του χρήστη.
- activity_settings.xml: Χρησιμοποιείται για την έξοδο του χρήστη από την εφαρμογή και την ενεργοποίηση για προβολή βάση προτιμήσεων.

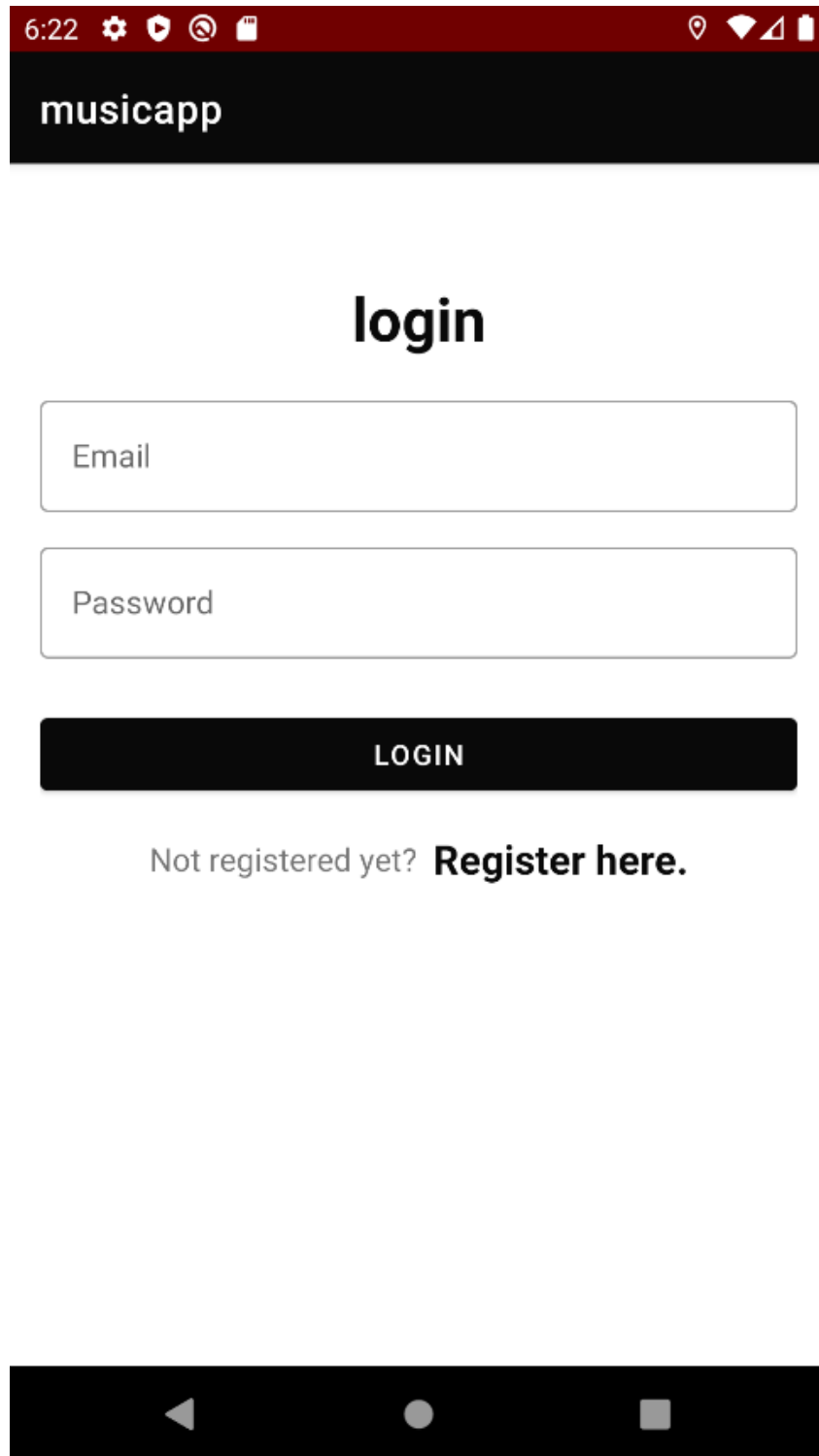
5.2.2 Αναφορά για την τελική εκτελέσιμη έκδοση

Μέσα από την διαδικασία της RUP, δημιουργήθηκε το τελικό αποτέλεσμα της android εφαρμογής αναπαραγωγής μουσικής βάση προτιμήσεων και τοποθεσίας. Όλες οι εκκρεμότητες που αναλύθηκαν παραπάνω, διευθετήθηκαν. Το πρόγραμμα είναι έτοιμο προς χρήση. Παρέχει όλα τα βασικά ζητούμενα που αναφέρθηκαν κατά την φάση της έναρξης, καθώς και επιπλέον λειτουργίες οι οποίες προέκυψαν κατά την φάση της ανάλυσης απαιτήσεων.

6. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΤΗ

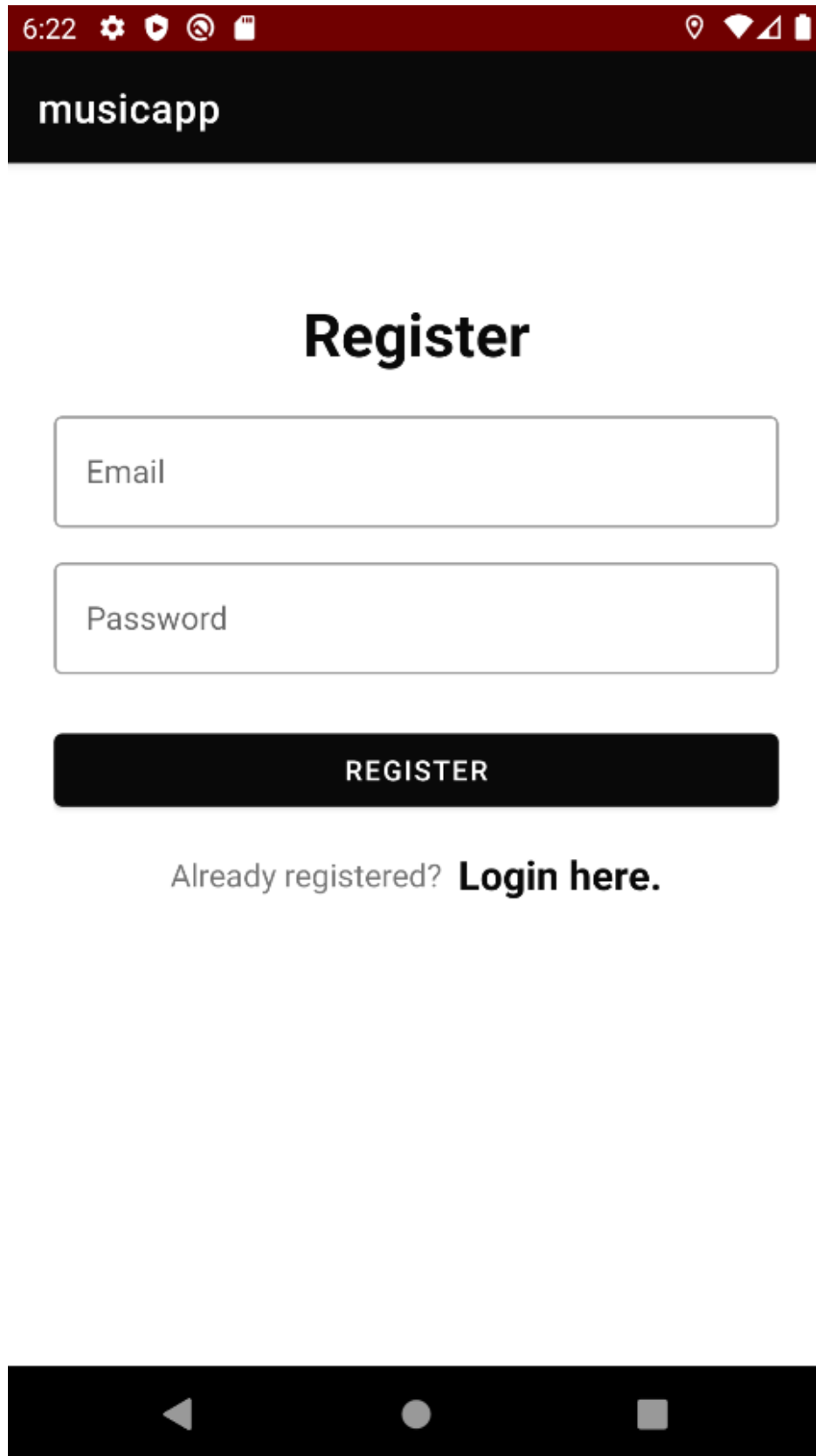
6.1 Σύντομη παρουσίαση του προγράμματος

Αρχική σελίδα/ σελίδα εισόδου:



The screenshot shows the login interface of an application named 'musicapp'. At the top, there is a dark red status bar with the time '6:22' and various icons. Below it, a black header bar displays the app name 'musicapp' in white. The main content area is white and features the word 'login' in a large, bold, black font. Underneath, there are two white input fields with rounded corners: the first is labeled 'Email' and the second is labeled 'Password'. Below these fields is a solid black button with the word 'LOGIN' in white, uppercase letters. At the bottom of the form, there is a link that reads 'Not registered yet? **Register here.**'. The entire screen is framed by a black Android navigation bar at the very bottom, which contains the back, home, and recent apps icons.

Σελίδα εγγραφής:



The screenshot shows a mobile application interface for 'musicapp'. At the top, there is a status bar with the time 6:22 and various icons. Below the status bar, the app name 'musicapp' is displayed in white text on a dark background. The main content area has a white background and features the title 'Register' in a large, bold, black font. Below the title, there are two input fields: 'Email' and 'Password', both with rounded corners and light gray borders. Underneath these fields is a solid black button with the word 'REGISTER' in white, uppercase letters. Below the button, the text 'Already registered? Login here.' is displayed, with 'Login here.' in a bold black font. At the bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.

6:22

musicapp

Register

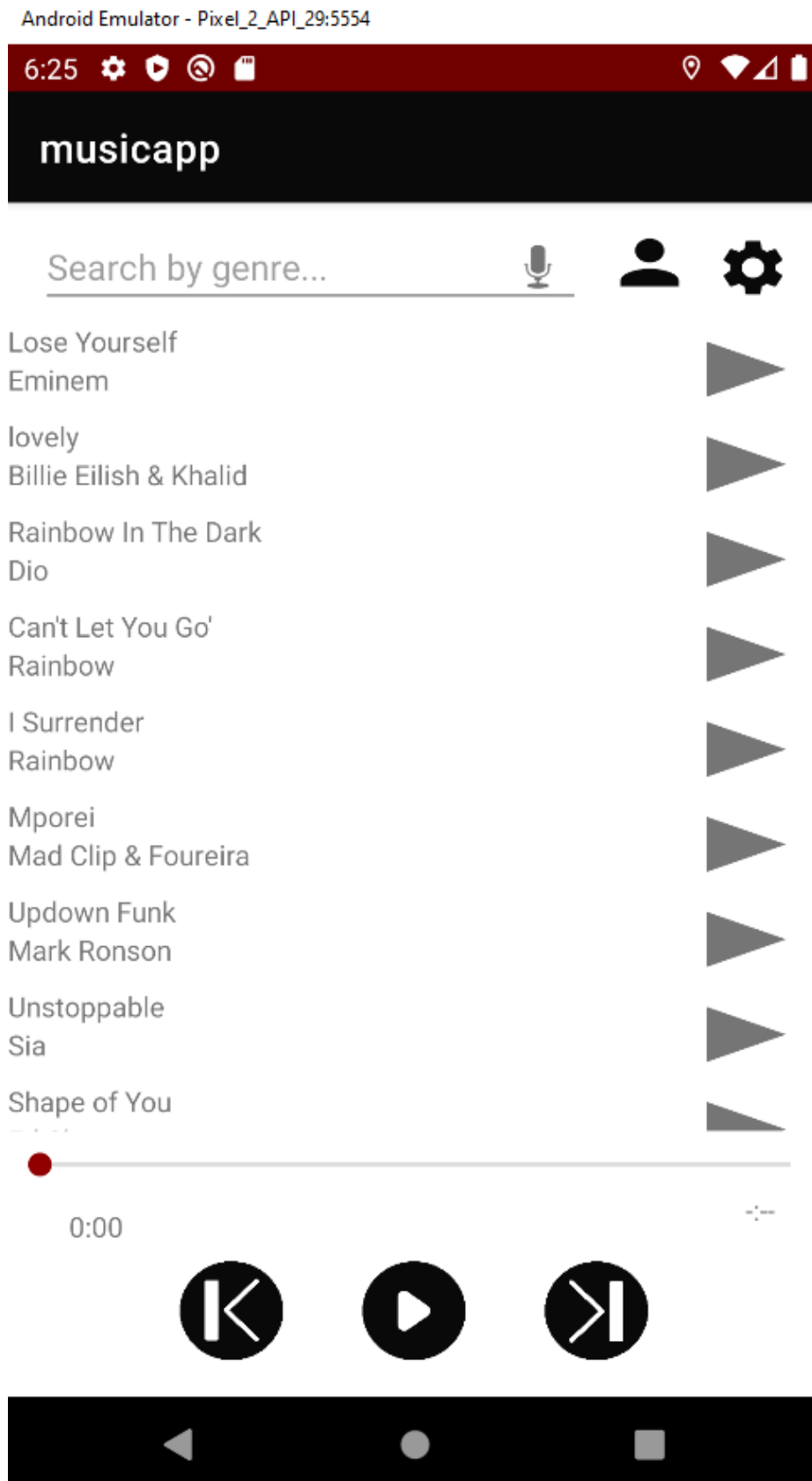
Email

Password

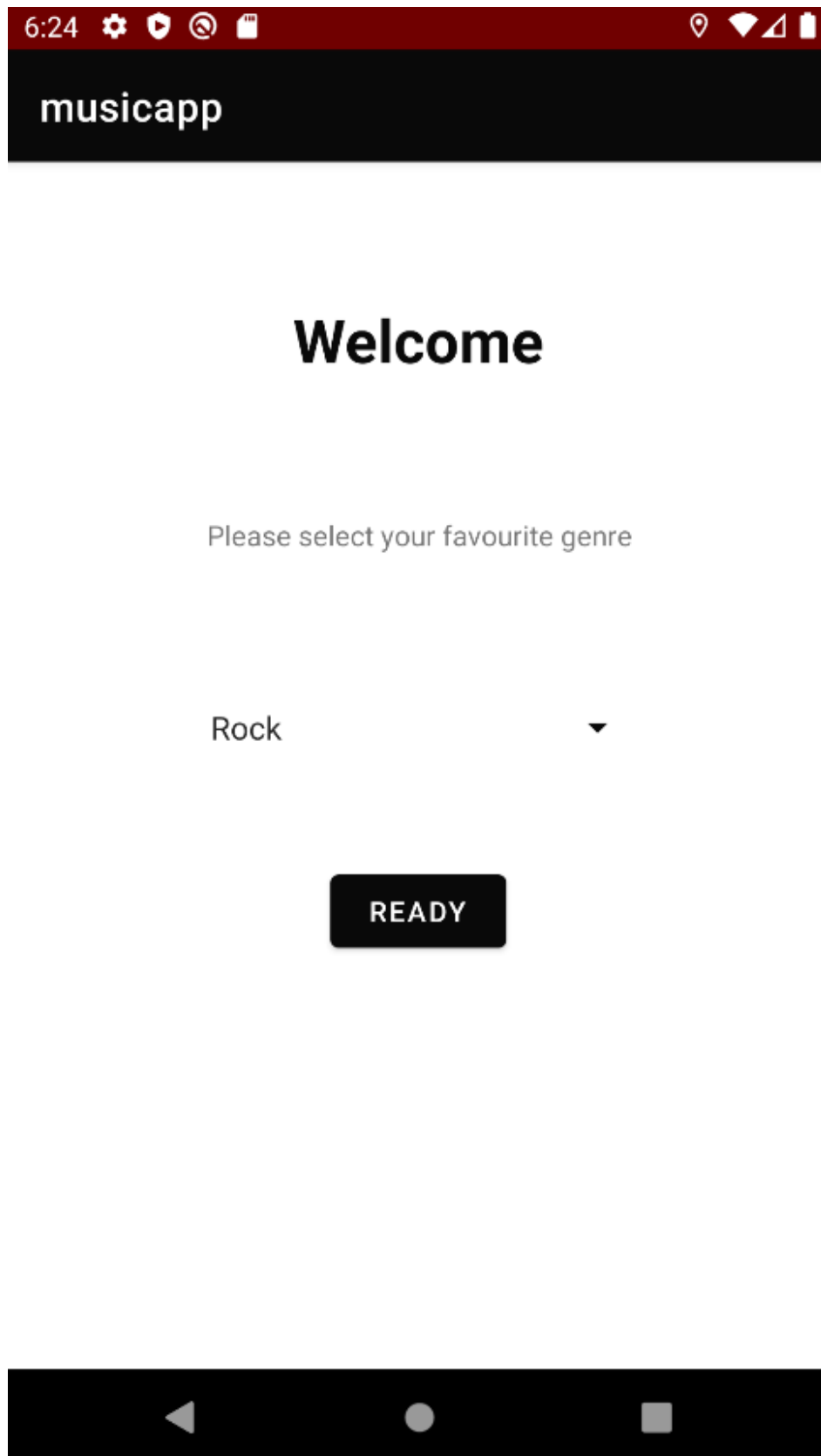
REGISTER

Already registered? **Login here.**

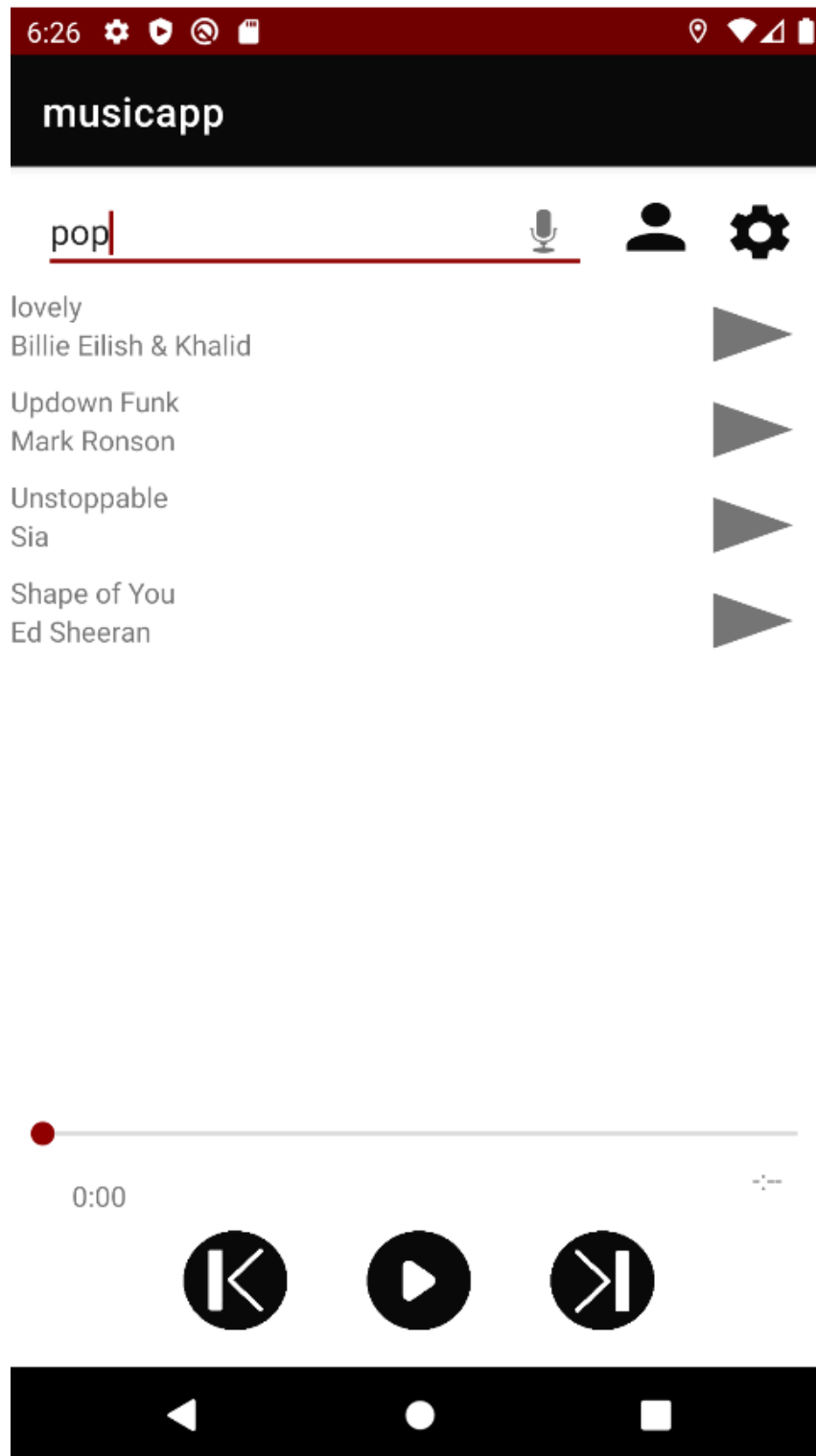
Αρχική σελίδα μετά την είσοδο:



Σελίδα δημιουργίας προτιμήσεων :



Σελίδα αποτελεσμάτων αναζήτησης βάση είδους :



Σελίδα επεξεργασίας πληροφοριών του χρήστη:

The screenshot shows a mobile application interface for editing user information. At the top, a red status bar displays the time 6:27 and various system icons. Below this, a black header bar contains the text "musicapp" in white. A gray bar with a person icon is positioned below the header. The main content area features a large gray circle representing a user profile picture. Below the profile picture, there are two input fields: the first is labeled "Email" and contains the text "email@mail.com"; the second is labeled "Password". A gray button with the text "UPDATE" is located below the input fields. The bottom of the screen shows a black navigation bar with standard Android icons.

6:27

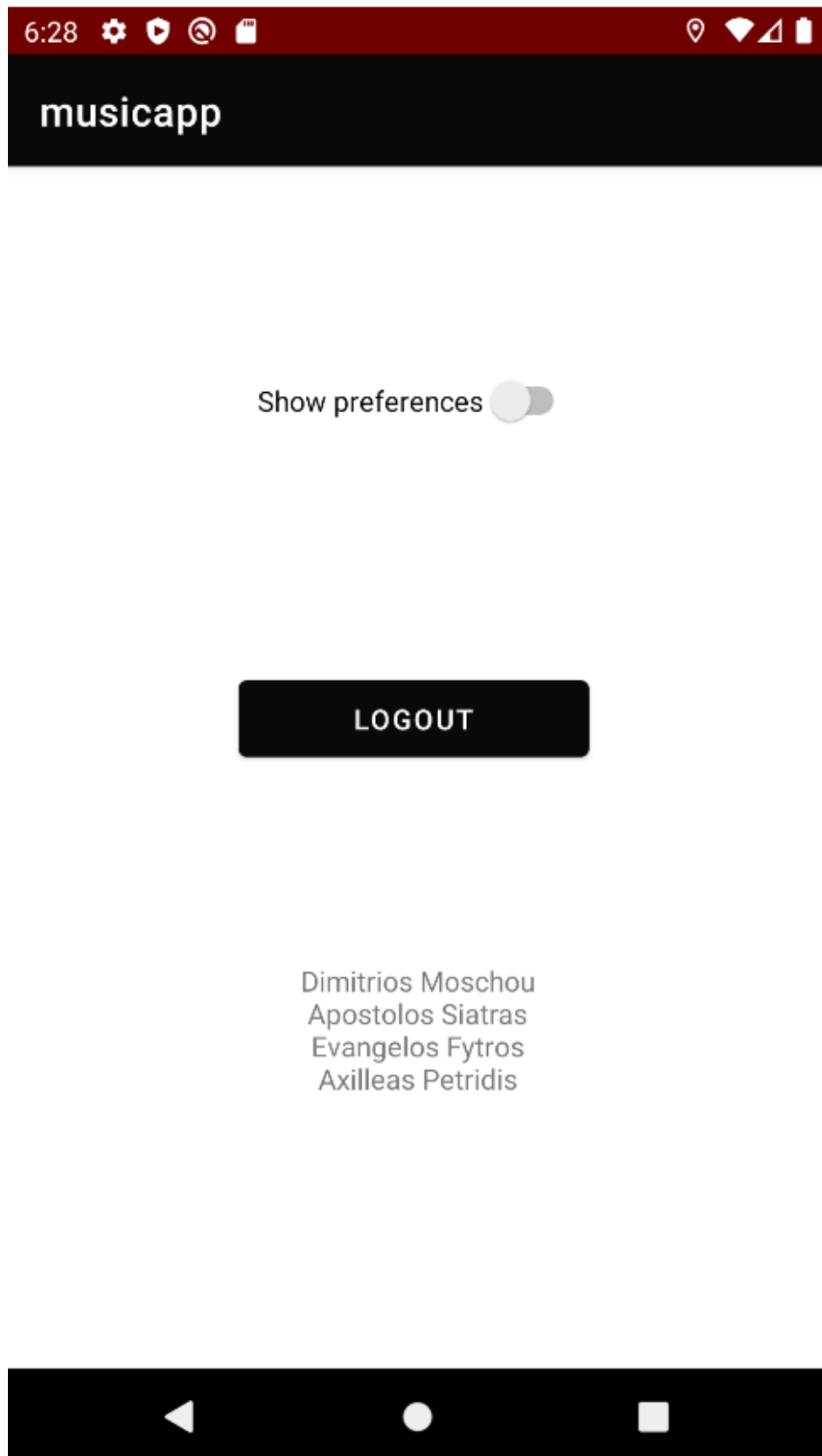
musicapp

Email
email@mail.com

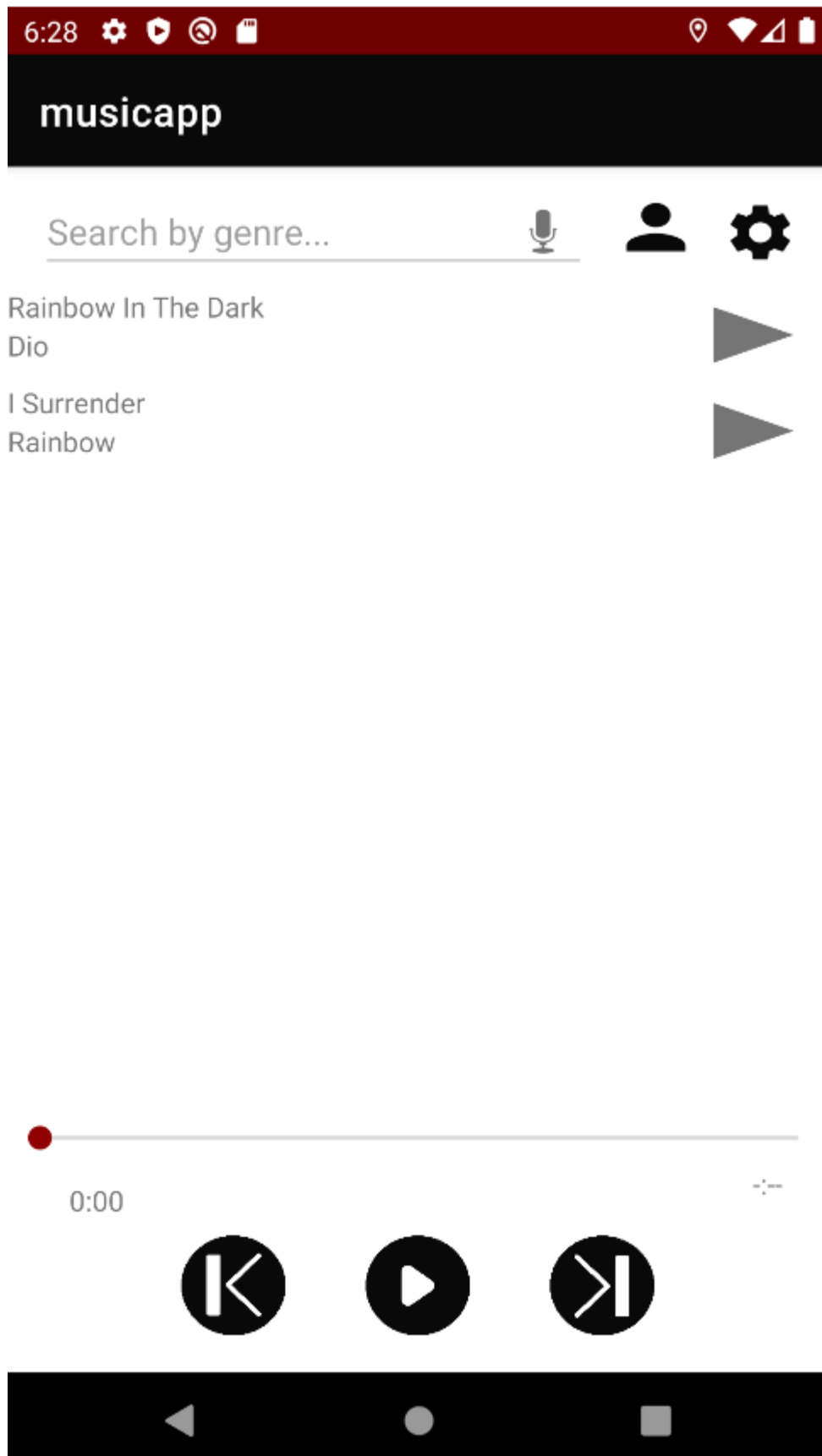
Password

UPDATE

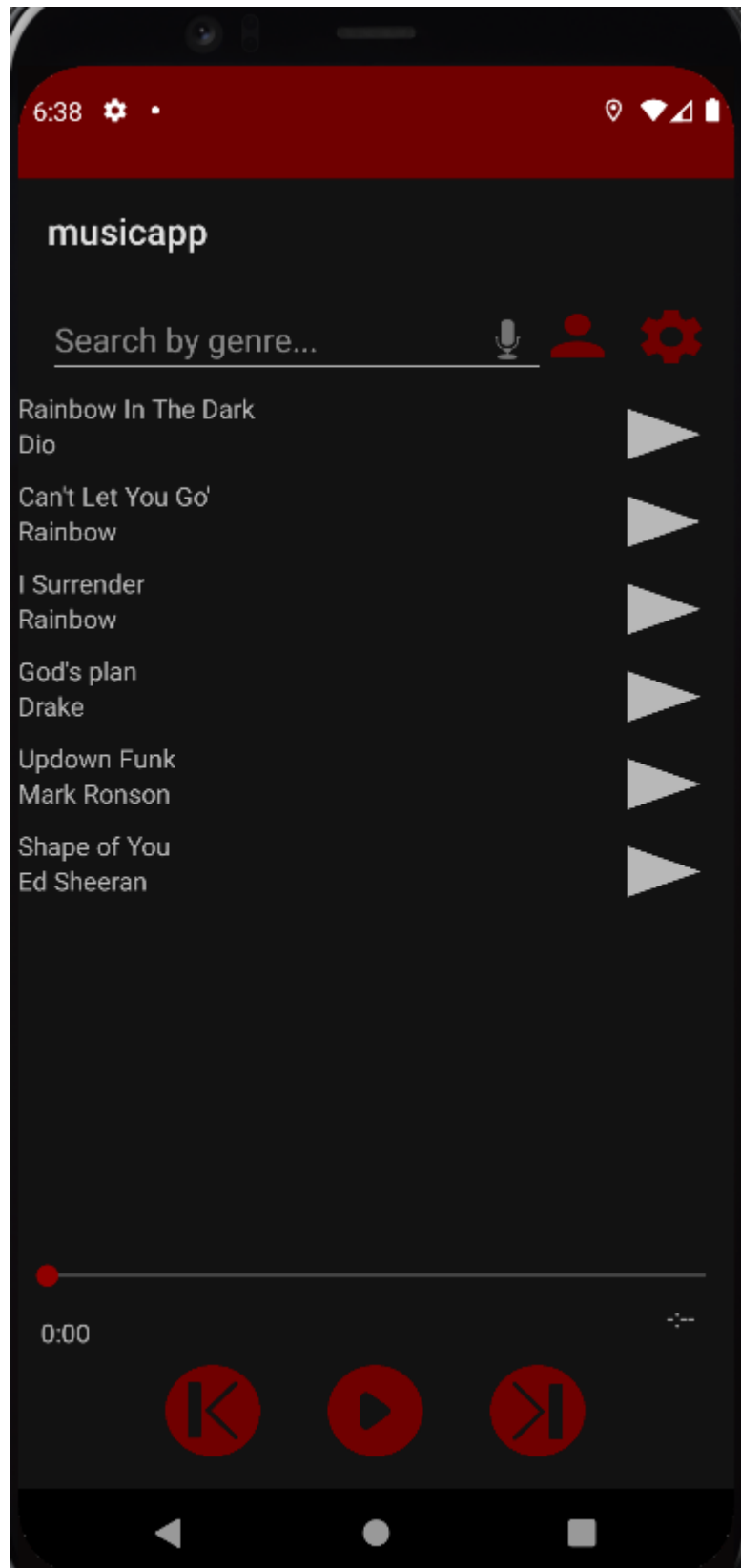
Σελίδα ρυθμίσεων :



Σελίδα εισόδου με ενεργοποιημένα εξατομικευμένα αποτελέσματα:



Αρχική σελίδα εφαρμογής με ενεργοποιημένη την σκοτεινή εμφάνιση:



6.2 Ανάλυση σεναρίων λειτουργίας

Αρχικά, εμφανίζεται η σελίδα σύνδεσης, στην οποία υπάρχει η φόρμα για σύνδεση στην εφαρμογή (login), καθώς και η επιλογή για την εγγραφή ενός νέου χρήστη στην εφαρμογή (register).

Στη συνέχεια, εμφανίζονται όλα τα διαθέσιμα τραγούδια τα οποία μπορεί να ακούσει ο χρήστης, ανάλογα με την χώρα στην οποία βρίσκεται, χρησιμοποιώντας το GPS.

Ο χρήστης, πατώντας στις ρυθμίσεις μπορεί να επιλέξει την εμφάνιση των τραγουδιών ανάλογα με τις προτιμήσεις του κάθε χρήστη, καθώς και να αποσυνδεθεί από την εφαρμογή.

Με την είσοδο στο προφίλ, ο χρήστης μπορεί να δει το email του και να αλλάξει το password του λογαριασμού του.

Χρησιμοποιούνται 2 αισθητήρες του κινητού για την βελτίωση της εμπειρίας του χρήστη:

- Μικρόφωνο: Έχει υλοποιηθεί λειτουργία voice recognition για την φωνητική αναζήτηση του είδους των τραγουδιών.
- Αισθητήρας εγγύτητας: Όταν ο χρήστης γυρίζει το κινητό του ανάποδα στο τραπέζι ή το βάζει στην τσέπη του, η οθόνη απενεργοποιείται αυτόματα.