ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής



Εργασία Μαθήματος

Σύγχρονα Θέματα Τεχνολογία Λογισμικού

|  |  |
| --- | --- |
| **Αριθμός εργασίας – Τίτλος εργασίας** | Απαλλακτική εργασία 2018-2019: Υλοποίηση  Android εφαρμογής αναπαραγωγής μουσικής |
| Ονόματα ομάδας | ΚΟΤΤΑΡΙΔΗΣ ΑΘΑΝΑΣΙΟΣ, ΔΙΑΤΣΙΓΚΟΣ ΙΩΑΝΝΗΣ, ΤΣΙΛΗΣ ΔΗΜΗΤΡΙΟΣ |
| Αρ. Μητρώων | Π15059, Π15037, Π11159 |
| Ημερομηνία παράδοσης | 15-2-2019 |

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

**1.** Εισαγωγή………………………………………………………………………………………..…………………………..

**2.** Ερώτημα 1: Ανάλυση αποτίσεων και UML………………………………..…………………………………

**2.1.** Ανάλυση απαιτήσεων εργασίας.........................................................................

**2.2.** Διαγράμματα UML.............................................................................................

**3.** Eρώτημα 2 Ανάπτυξη 4ον φάσεων RUP…………………………………..…………………………………..

**4.** Τεκμηρίωση κώδικα……………………………………………………………………………………………..

**4.1** Java Class**.** .....................................................................................................

**4.2.** Java Activities..............................................................................................

**4.3.** User Interface…………………………………………………………………………………………

**5.** Aποθηκευση Δεδομένων…………………………………………………………………………………………………..

1. Εισαγωγή

Η εργασία αυτή αποτελεί μια εφαρμογή δυναμικής αναπαραγωγής τραγουδιών που βρίσκονται αποθηκευμένα είτε online είτε τοπικά στον αποθηκευτικό χώρο του κινητού, η αναπαραγωγή γίνεται βάση της τοποθεσίας των προτιμήσεων του χρήστη. Ποιο συγκεκριμένα το Project έχει υλοποιηθεί σε γλώσσα JAVA για ανάπτυξη Android εφαρμογών με τη βοήθεια του εργαλείου Android studio. Για αποθήκευση των δεδομένων της εφαρμογής χρησιμοποιήθηκαν:

* Firebase για user authentication, αποθήκευση των online playlist καθώς και των αρχείων που αποτελούν της playlist (εικόνες και τραγούδια)
* Shared Preferences για αποθήκευση των ρυθμίσεων του profile του χρήστη όπως Music Preferences και Points Of Interest καθώς και των πληροφοριών που εισάγει ο χρήστης στο σύστημα για τα τραγούδια που είναι αποθηκευμένα στον τοπικό αποθηκευτικό χώρο του κινητού

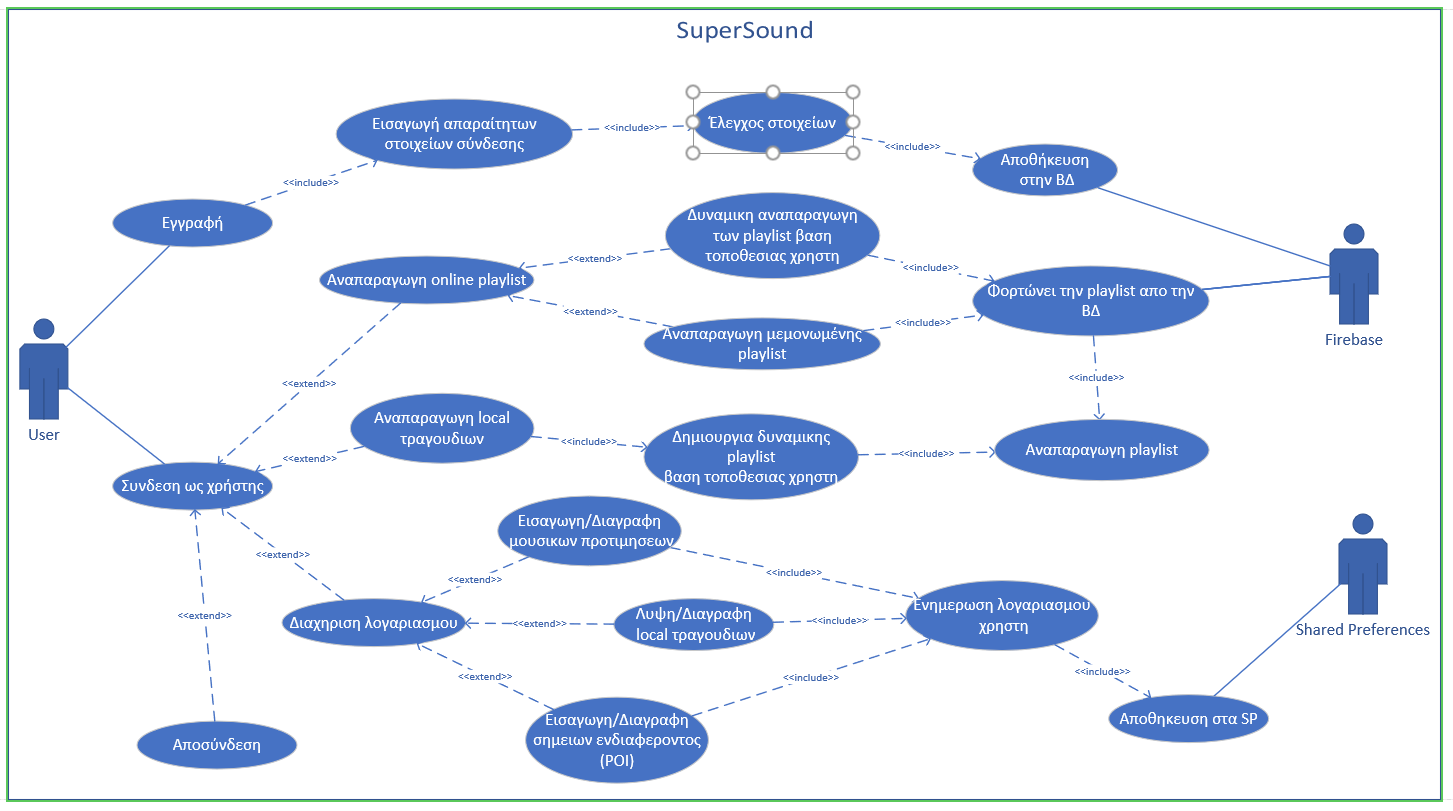
Η εφαρμογή τρέχει κανονικά σε οποιαδήποτε Android συσκευή με λογισμικό API 23 και μετα αρκεί να είναι συνδεδεμένη στο διαδίκτυο.

1. . Ερώτημα 1: Ανάλυση αποτίσεων και UML
   1. Ανάλυση απαιτήσεων εργασίας

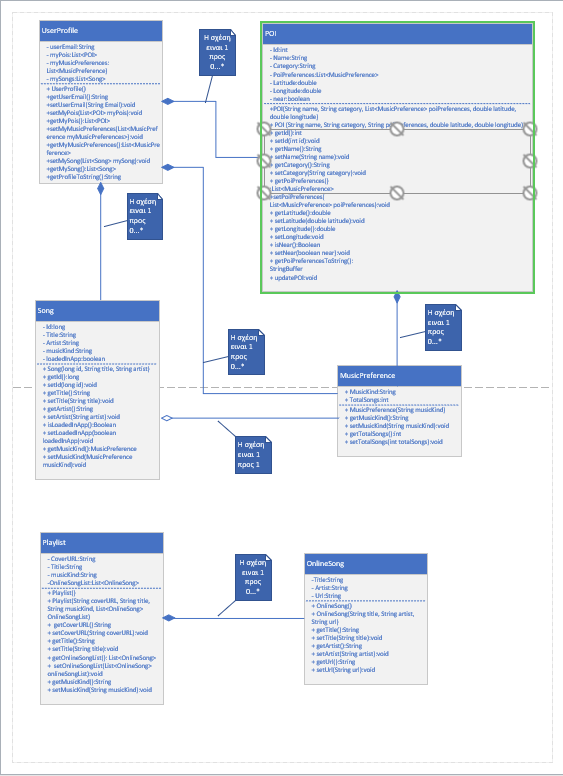
Στην εργασία μας καλούμαστε να υλοποιήσουμε μια Android εφαρμογή αναπαραγωγής μουσικής βάσει του προφίλ του χρήστη (προτιμήσεων) και βάσει της τοποθεσίας του. . Στην εφαρμογή μας ο χρήστης έχει την δυνατότητα να Εγγραφεί και να Συνδεθεί στο σύστημα, να διαχειρίζεται το Profile του (δηλαδή να εισάγει και να διαγραφεί είδη μουσικής στης προτιμήσεις του, να κατευνάζει στην εφαρμογή τραγούδια που είναι αποθηκευμένα στο κινητό χαρακτηρίζοντας κάθε τραγούδι σε ποιο είδος των προτιμήσεων του ανήκει ώστε να μπορεί να αναπαραχθεί δυναμικά, αλλά και να διαγραφεί αυτά τα τραγούδια αν το επιθυμεί ο χρήστης και τέλος να προσθέτει και να τροποποιεί τα σημεία ενδιαφέροντος του «POIs» στα οποία θα εισάγει μια λίστα από ειδή μουσικής που επιθυμεί να ακούει όταν βρίσκετε σε αυτήν τη τοποθεσία) , να αναπαράγει δυναμικά σε μορφή playlist τα τραγούδια που αντιστοιχούν στα ειδή προτιμήσεων του χρήστη σε μια συγκεκριμένη τοποθεσία , να αναπαράγει δυναμικά της onlinePlaylist που είναι αποθηκευμένες στη Firebase των οποίον το είδος μουσικής αντιστοιχεί σε κάποια από της προτίμησης του χρήστη σε μια συγκεκριμένη τοποθεσία, επίσης μέσο της εφαρμογής παρέχετε η δυνατότητα στον χρήστη να αναγάγει και ξεχωριστά οποιαδήποτε από της διαθέσιμες online playlists επιθυμεί.

**2.2. Διαγράμματα UML**

**Use Case Diagram**



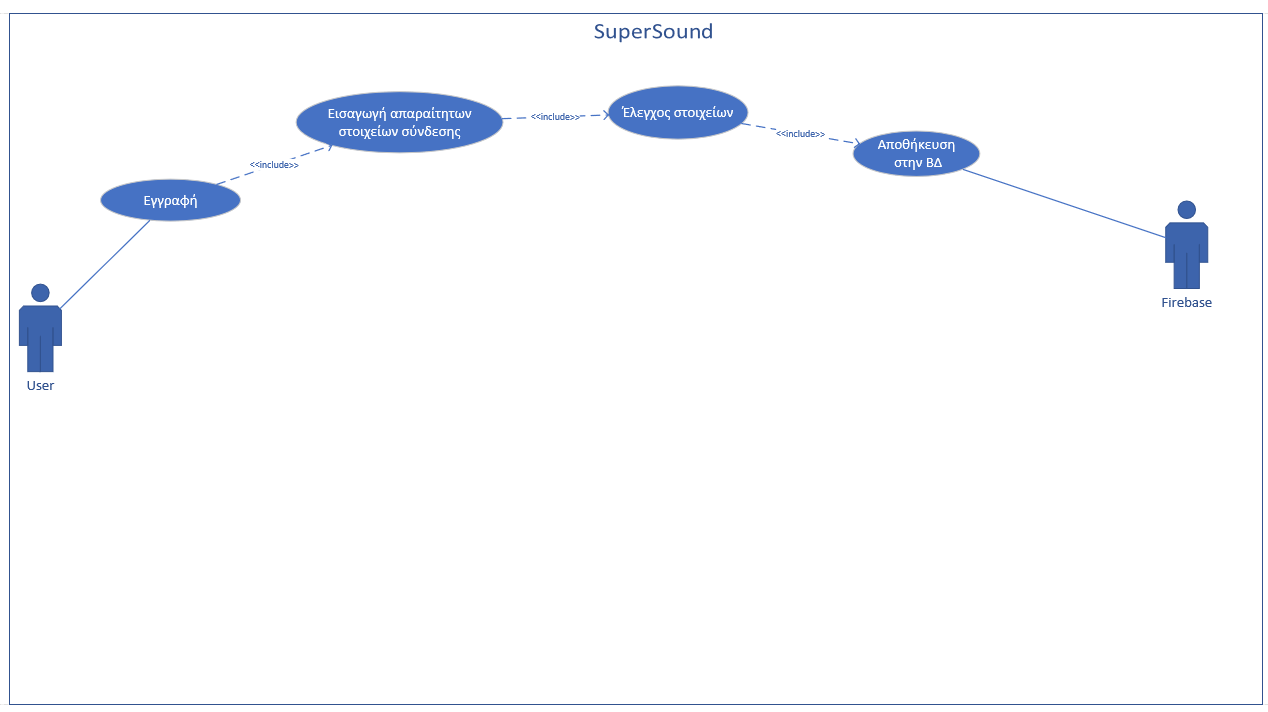
**Class Diagram**

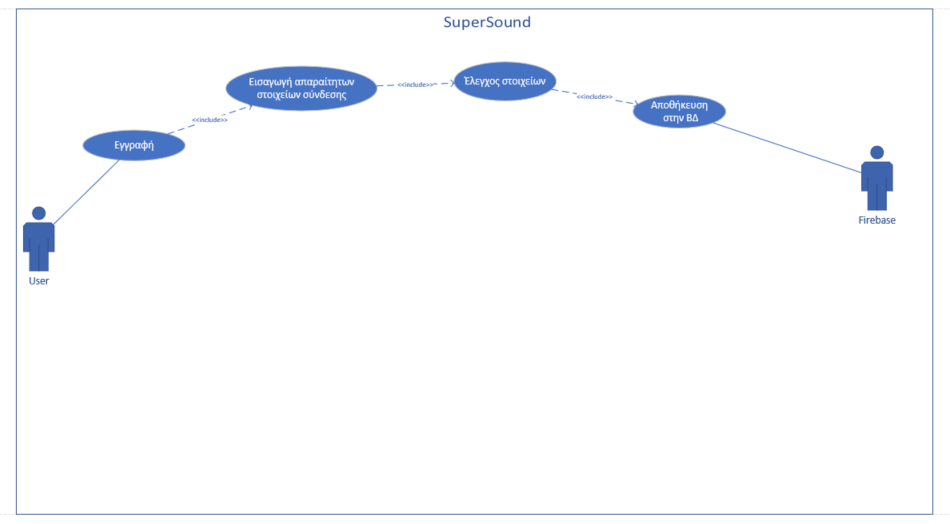


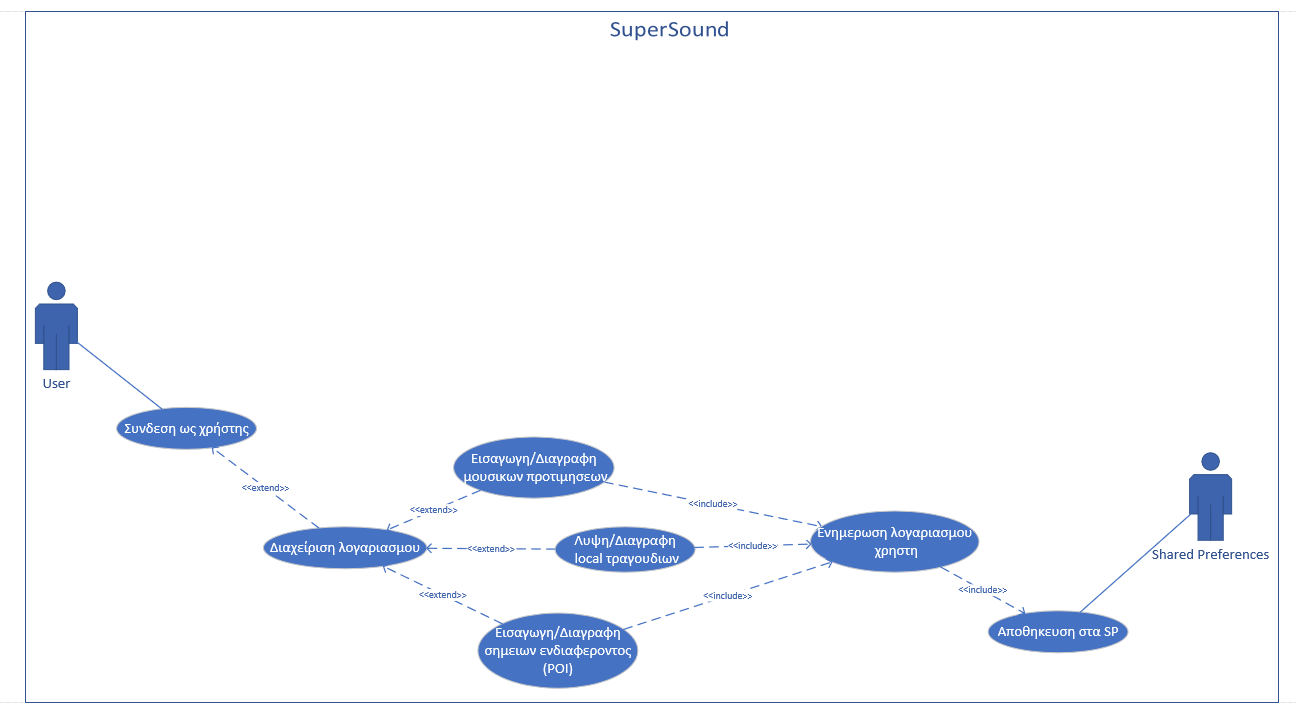
1. Eρώτημα 2 Ανάπτυξη 4ον φάσεων RUP

Διαγράμματα Use Case

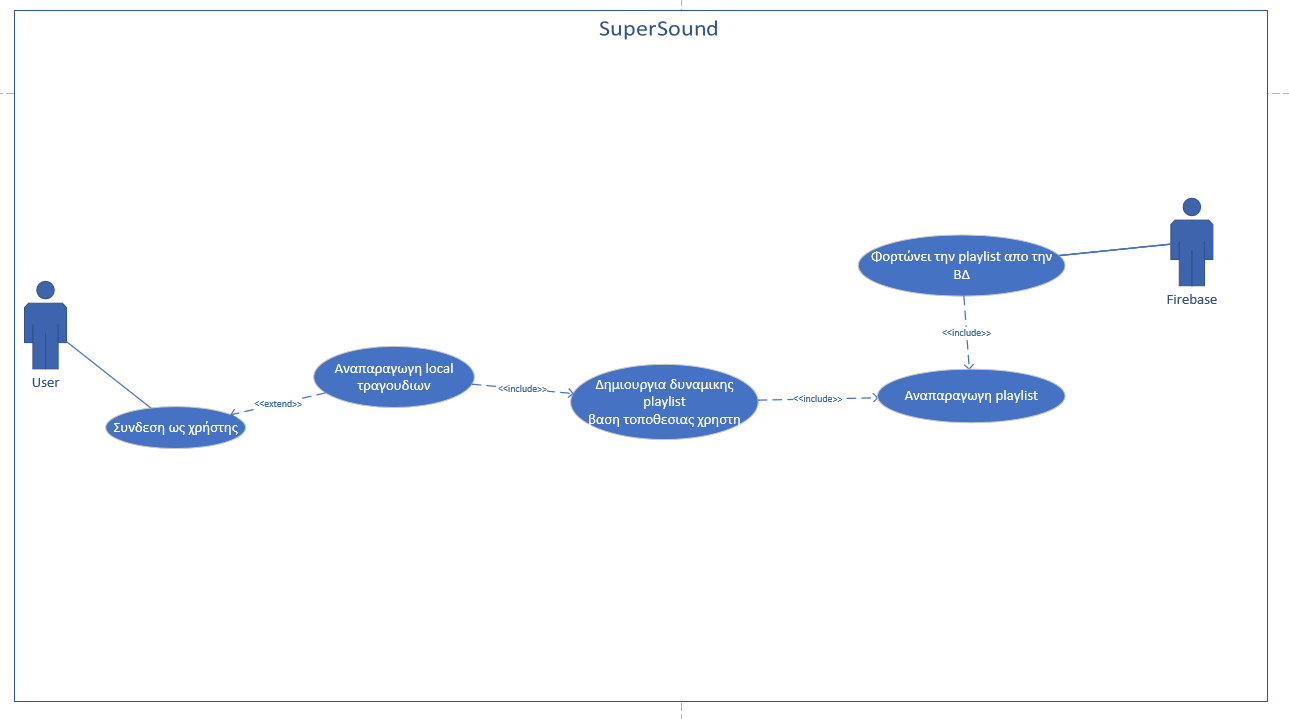
* 1η έκδοση

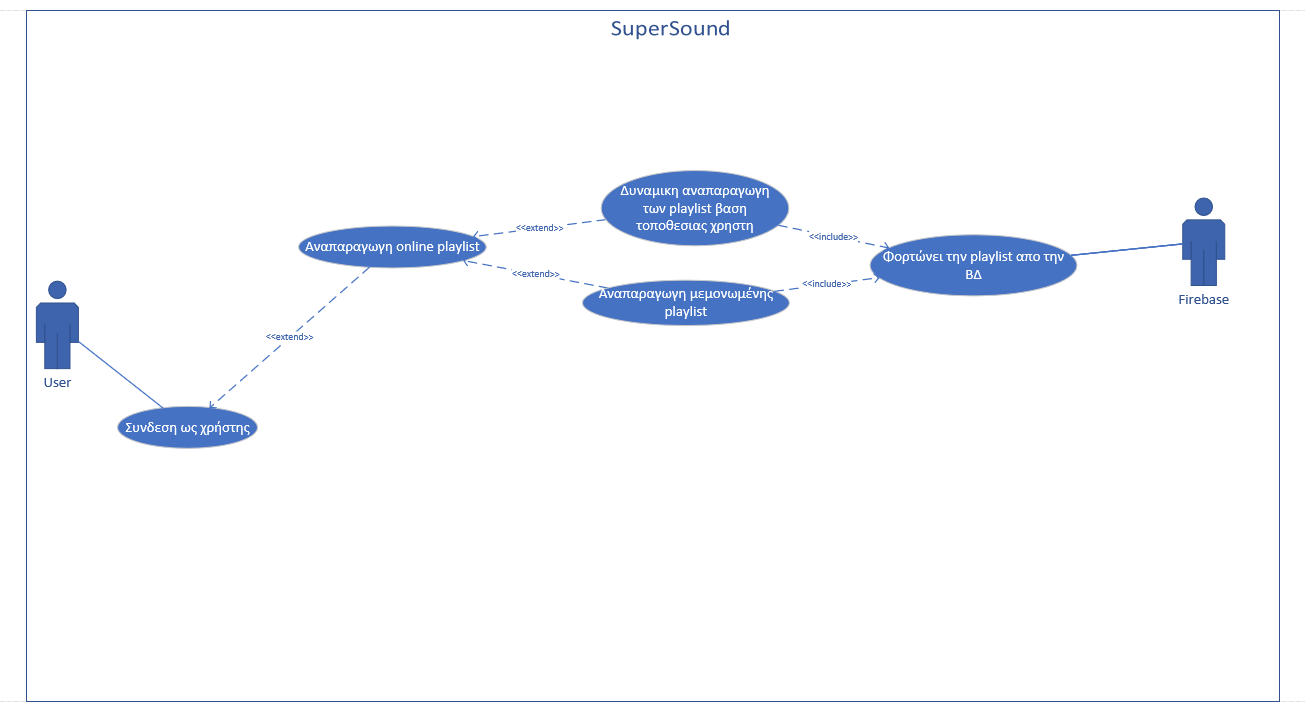


* 2η έκδοση Εγραφή Χρηστη 
* 2η έκδοση Διαχείριση Λογαριασμού



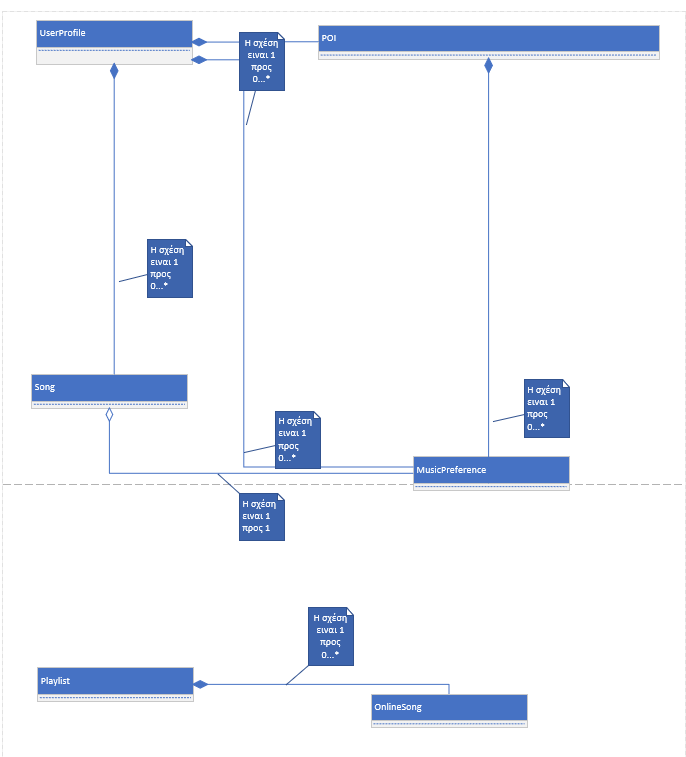
* 2η έκδοση Αναπαραγωγή Local τραγουδιών



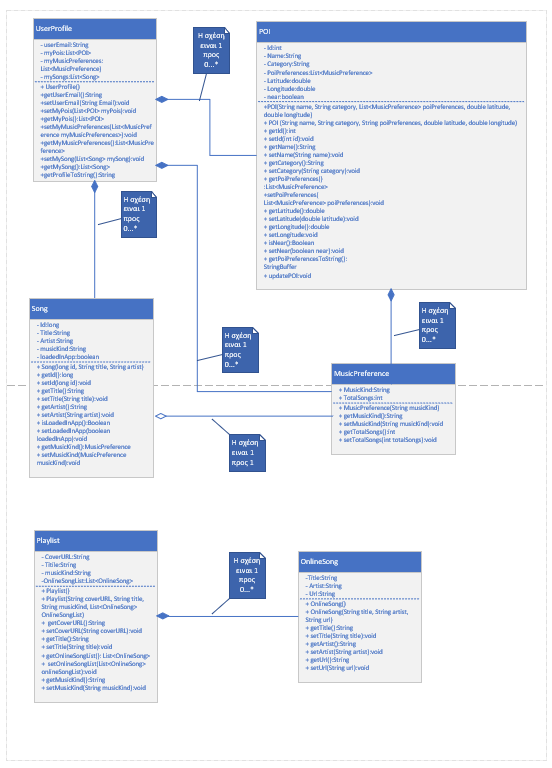
* 2η έκδοση Αναπαραγωγή Online Playlist

**Διαγράμματα Class**

* 1η έκδοση



* 2η έκδοση



1. Τεκμηρίωση Κώδικα
   1. Java Class

Για την υλοποίηση του Project χρειάστηκε να δημιουργήσουμε ορισμένες τάξης προκειμένου να βοηθηθούμε με την διαχείριση τον δεδομένων στο σύστημα

* ***UserProfile.java*** Η κλάση αυτή αφορά τους χρήστες του συστήματος και περιέχει το email του χρήστη (userEmail), μια λίστα με τα POIs που έχει εισάγει ο χρήστης στο σύστημα (myPois), μια λίστα με της μουσικές προτίμησης του χρήστη (myMusicPreferences), μια λίστα με τα τραγούδια που είναι αποθηκευμένα στη συσκευή και τα έχει κατεβάσει ο χρήστης στην εφαρμογή (mySong) καθώς και καταλλήλους getters και setters για την επεξεργασία αυτόν.
* ***MusicPreference.java*** Η κλάση αυτή αφορά τις μουσικές προτίμησης του χρήστη. Κάθε αντικείμενο αυτής της τάξης αντιπροσωπεύει ένα είδος μουσικής που έχει εισάγει σαν προτίμηση ο χρήστης στο σύστημα η κλάση αυτή περιέχει μια String μεταβλητή (MusicKind) που περιέχει το όνομα της κατηγορίας πχ Rock καθώς , έναν counter (TotalSongs) που μετράει ποσά τραγούδια αυτής της κατηγορίας έχει εισάγει ο χρήστης στην εφαρμογή καθώς και καταλλήλους getters και setters για την επεξεργασία αυτόν.
* ***Song.java*** η κλάση αυτή αναπαριστά τα τραγούδια που είναι αποθηκευμένα στην μνήμη του κινητού και κάθε αντικείμενο της περιέχει ένα id (id) το οποίο είναι το id που έχει το τραγούδι στη μνήμη, τον τίτλο του τραγουδιού (title), το όνομα του τραγουδιστή ή του συγκροτήματος (artist) ,το είδος του τραγουδιού το οποίο είναι κάποιο από αυτά που έχει δήλωση ο χρήστης (musicKind) , μια boolean μεταβλητή (loadedInApp) που υποδεικνύει αν το συγκεκριμένο τραγούδι έχει φορτωθεί στη εφαρμογή καθώς και καταλλήλους getters και setters για την επεξεργασία αυτόν.
* ***POI.java*** η κλάση αυτή αναπαριστά τα σημεία ενδιαφέροντος (POIs) που έχει εισάγει ο χρήστης στο σύστημα και περιέχει ένα id (id), το όνομα του POI (Name), τη κατηγορία στην οποία ανήκει το POI (Category), της γεωγραφικές συντεταγμένες του POI (Latitude και Longitude) , μια λίστα με της προτιμήσεις του χρήστη στη συγκεκριμένη τοποθεσία (PoiPreferences), μια boolean μεταβλητή (near) που υποδεικνύει αν ο χρήστης βρίσκετε μέσα στην ακτίνα του POI καθώς και καταλλήλους getters και setters για την επεξεργασία αυτόν.
* ***Playlist.java*** η κλάση αυτή αναπαριστά της playlist που διαβάζονται από την firebase και αποτελείτε από μια String μεταβλητή (coverUrl) που περιέχει το URL του εξοφλήσου της playlist, μια μεταβλητή (Title) που περιέχει τον τίτλο της playlist, μια μεταβλητή (MusicKind) που περιέχει το είδος των τραγουδιών της λίστας, μια λίστα από on line songs (OnlineSongList) που περιέχει τα τραγούδια της playlist, καθώς και καταλλήλους getters και setters για την επεξεργασία αυτόν.
* ***OnlineSong.java*** η κλάση αυτή αναπαριστά τα τραγούδια που είναι αποθηκευμένα στην firebase και περιέχει τον τίτλο τον τραγουδιών (Title), το όνομα του τραγουδιστή (Artist), το URL του τραγουδιού καθώς και καταλλήλους getters και setters για την επεξεργασία αυτόν.
  1. Java Activities

Για την υλοποίηση του Project χρειάστηκε να δημιουργήσουμε ορισμένα Activities. Τα Activities αυτά αλληλοεπιδρούν με τον χρήστη μέσο των xml layout που έχουν δημιουργηθεί.

* ***SignInActivity.java*** Αυτό το Activity χρησιμοποιείτε για την είσοδο του χρήστη στο σύστημα, δέχεται από τον χρήστη το email του και το password του και συνδέετε με την firebase προκειμένου να γίνει ταυτοποίηση με τα στοιχεία κάποιου ήδη καταχωρημένου χρήστη και τότε να τον μεταφέρει στη MainActivity.java εάν δεν είναι η πρώτη είσοδος του χρήστη στο σύστημα, αν ο χρήστης κάνει log in για πρώτη φορά τότε μεταφέρετε στην SetUpProfileActivity.java έτσι ώστε να εισάγει στο σύστημα τα απαραίτητα στοιχεία που χιάζετε για να λειτουργήσει. Το layout του Activity έχει υλοποιηθεί στο αρχείο activity\_sign\_in.xml.
* ***SignUpActivity.java*** Αυτό το Activity χρησιμοποιείτε για την εγγραφή του χρήστη στο σύστημα, δέχεται το username, το email και το password από τον χρήστη και τα αποθηκεύει στη firebase ελέγχοντας την εγκυρότητα τους. Το Layout αυτού του Activity έχει υλοποιηθεί στο αρχείο activity\_sign\_up.xml.
* ***MainActivity.java*** Αυτό το Activity αποτελεί την κεντρική σελίδα της εφαρμογής περιέχει ένα Drawer Menu μέσο του οπίου ο χρήστης μπορεί πραγματοποιήσει της έξεις λειτουργίες:
  1. Αποσύνδεση από τον λογαριασμό του. Αποσυνδέει τον χρήστη από τον λογαριασμό του και τον μεταφέρει στο SignInActiviti.java.
  2. Προσθήκη σημείου ενδιαφέροντος (POI). Αυτή η επιλογή μεταφέρει τον χρήστη στο POIsActivity.java
  3. Προθήκη ιδίους μουσικής στης προτιμήσεις του χρήστη (Music Preferences). ). Αυτή η επιλογή μεταφέρει τον χρήστη στο MusicPreferencesActivity.java.
  4. Προσθήκη τραγουδιού από τον αποθηκευτικό χώρο του κινητού στο σύστημα (My Music). ). Αυτή η επιλογή μεταφέρει τον χρήστη στο AddSongActivity.java.

Η υλοποιήσει αυτού του μενού πραγματοποιείτε με 2 αρχεία το nav\_header.xml το οποίο περιέχει την μορφοποίηση του drawer και το drawer\_menu\_xml το οποίο αποτελεί το layout των κατηγοριών του μενού. Επίσης σε αυτό το Activity γίνετε και η επιλογή εκτελέσεις playlist ο χρήστης μπορεί να επιλέξει να αναπαράγει δυναμικά βάση της τοποθεσίας του και των προτιμήσεων του τα τραγούδια που είναι αποθηκευμένα στο κινητό του ή να επιλέξει να αναπαράγει δυναμικά πάλι βάση της τοποθεσίας και των προτιμήσεων του της playlists που είναι αποθηκευμένες στην firebase. Επίσης ο χρήστης έχει την δυνατότητα και απλός να αναπαράγει κάποια από της διαθέσιμες playlists που είναι αποθηκευμένες στην firebase. Οπότε ο χρήστης επιλεγεί να απαγάγει κάποια playlists μεταφέρετε στο MusicPlayerActivity.java παίρνοντας μέσο του Intent τα απαραίτητα στοιχεία για να γίνει η αναπαραγωγή. Το layout του Activity έχει υλοποιηθεί στο αρχείο activity\_main.xml.

* ***SetUpProfileActivity.java*** Αυτό το Activity αλληλοεπιδρά με το activity\_set\_up\_profile.xml και απλά ενημερώνει των χρήστη ότι θα χρειαστεί να εισάγει κάποια στοιχεία προκείμενου να ρυθμίσει το profile του ώστε να μπορέσει να πραγματοποιηθεί δυναμική αναπαραγωγή τραγουδιών.
* ***MusicPreferencesActivity.java*** Αυτό το Activity αλληλοεπιδρά με το activity\_music\_preferences.xml σε αυτό το Activity ο χρήστης μπορεί να εισάγει κατηγορίες μουσικής που του αρέσουν. Οι κατηγορίες προβάλλονται σε ένα Recycler view το οποίο συνδέετε με αυτό το Activity μέσο τουMusicPreferenceAdapter.java. Επίσης για για να γίνει εισαγωγή κατηγορίας στο σύστημα ο χρήστης πρέπει να πατήσει πανό στο floating action button το οποίο εμφανίζει ένα custom dialog που υλοποιήσαμε και δέχεται κείμενο μέσο ενός Edit Text, η υλοποίηση των συγκεκριμενών dialog γίνετε μέσο της TextViewDialogClass.java η οποία έχει layout το custom\_text\_view\_dialog.xml.
* ***AddSongActivity.java*** Αυτό το Activity αλληλοεπιδρά με το activity\_add\_song.xml. Αρχικά ζητάει permission από τον χρήστη για να διαβάσει αρχαί από τον αποθηκευτικό χώρο του κινητού. Αν ο χρήστης δώσει άδεια τότε προβάλει όλα τα τραγούδια που είναι αποθηκευμένα στη συσκευή σε ένα Recycler view το οποίο συνδέετε με αυτό το Activity μέσο του AddSongsAdapter.java. Για να προσθέσει ο χρήστης ένα τραγούδι στην εφαρμογή αρκεί να πατήσει πάνω στο συγκεκριμένο item του recycler view τότε θα εμφανιστεί πάλι ένα custom dialog το οποίο περιέχει έναν Spinner ο οποίος έχει σαν αντικείμενα όλες της κατηγορίες τραγουδιών που έχει εισάγει ο χρήστης στο σύστημα, το συγκεκριμένο dialog υλοποιείτε μέσο του SpinnerDialogClass.java. Επίσης ο χρήστης έχει την δυνατότητα να αλλάξει το είδος κάποιο κατευνασμένου τραγουδιού απλά πατώντας πανό του και επιλέγοντας άλλο είδος από το dialog ή να διαγράψει κάποιο τραγούδι κάνοντας swipe left.
* ***POIsActivity.java*** Αυτό το Activity αλληλοεπιδρά με το activity\_pois.xml ο χρήστης σε αυτό το Activity μπορεί να ημερώνει και να προβάλει τα σημεία ενδιαφέροντος POIs. Τα σημεία διαφέροντος του χριστή προβάλλονται σε ένα Recycler view το οποίο συνδέετε με αυτό το Activity μέσο του POIsAdapter.java. Για να γίνει εισαγωγή ενός POI στο σύστημα ο χρήστης πατάει πανό στο floating action button το οποίο τον μεταφέρει στο AddPoiActivity οπού εκεί γίνεται η προσθήκη. Ο χρήστης έχει τη δυνατότητα να τροποποιήσει ένα POI απλά πατώντας πανό του, τότε μεταφέρετε πάλι στην AddPoiActivity αλλά για να πραγματοποιήσει Edit αυτή τη φορά, επίσης μπορεί να πραγματοποιήσει διαγραφή κάποιου POI κάνοντας swipe left.
* ***AddPoiActivity.java java*** Αυτό το Activity αλληλεπιδρά με το activity\_add\_poi.xml. Σε αυτότο Activity ο χρήστης μπορεί να εισάγει ένα σημείο ενδιαφέροντος POI ή να τροποποιήσει κάποιο είδη υπάρχουν. Για να γίνει εισαγάγει ο χρήστης εισάγει το όνομα του POI, της γεωγραφικές του συντεταγμένες, την κατηγορία του POI χρησιμοποιώντας πάλι ένα custom dialog το οποίο προβάλει όλες της διαθέσιμες κατηγορίες και δίνει στον χρήστη την δυνατότητα να επιλέξει μόνο μια, και της προτίμησης του χρήστη το συγκεκριμένο POI. Η επιλογή τον προτιμήσεων γίνεται μέσο ενός custom multi select dialog. Τα 2 αυτά custom dialogs είναι ορισμένα μέσα σε αυτό το Activity. Τέλος με το πάτημα το κουμπιού Save στέλνει στο POIsActivity το νέο POI χρησιμοποιώντας activity for results.
* ***MusicPlayerActivity.java*** Αυτό το Activity αλληλοεπιδρά με το activity\_music\_player.xml. Σε αυτό το activity αναπαράγοντες τα τραγούδια στην εφαρμογή. Υπάρχουν 3ης διαφορετικές λειτουργίες που υλοποιεί αυτό το Activity.
  1. Αναπαραγωγή δυναμικής playlist από τα τραγούδια που είναι αποθηκευμένα στο κινητό οπού τότε το activity δέχεται από την MainActivity τα τοπικά τραγούδια και της προτιμήσεις του χρήστη και τα POIs του και βάση της τοποθεσίας του διαμορφώνει την playlist καταλληλά ώστε να παίζει μόνο τραγούδια κατηγοριών του ανήκουν στης προτίμησης του για τη συγκεκριμένη τοποθεσία
  2. Αναπαραγωγή δυναμικά της playlist που είναι αποθηκευμένες στη Firebase πάλι βάση της τοποθεσίας του. Δέχεται δεδομένα από την MainActivity της OnlinePlaylist που ανήκουν στης προτίμησης του χρήστη καθώς και τις προτιμήσεις του χρήστη και τα POIs τις αναπαράγει δυναμικά βάση της τοποθεσίας του.
  3. Αναπαραγωγή κάποιας OnlinePlaylist μεμονωμένα ανεξάρτητα από την τοποθεσία του χρήστη.

Την πρώτη φορά που ο χρήστης επιθυμεί να αναπαράγει δυναμική playlist είτε online είτε από τραγούδια του τηλεφώνου ζητάτε από τον χρήστη permission ώστε να έχει η εφαρμογή πρόσβαση στην τοποθεσία του. Το Layout του Activity παραμένει ίδιο και για της 3ης κατηγορίες αναπαραγωγής. Το layout αποτελείτε από: ένα image view που περιέχει το εξοφλώ της playlist που παίζει, 2 text views που προβάλουν το όνομα του τραγουδιού και τον καλλίτεχνι, 3 κουμπιά (play/pause, next, previous) και μια seekBar με 2 text view που δείχνουν σε ποιο λεπτό βρίσκετε το τραγούδι και πόσος χρόνος απομένει μέχρι να ολοκληρωθεί. Για να υλοποιηθεί αυτί η λειτουργία έχει δημιουργηθεί άλλο ένα thread το οποίο τρέχει κάθε ένα δευτερόλεπτο και ενημερώνει τα 2 text views.

* 1. User Interface

Το User Interface του συστήματος υλοποιείτε από xml layouts τα οποία έχουν καταλύει μορφοποίηση προκείμενου να είναι φιλικές πως τον χρήστη. Τα xml αυτά αλληλοεπιδρούν με τα Activity προκειμένου να μπορούν να διαμορφώνονται δυναμικά . Κάθε Activity αλληλοεπιδρά με ένα xml layout όπως αναφέρθηκε στην προηγουμένη ενότητα. Έχουν δημιουργηθεί όμως και κάποια extra layouts τα οποία χρησιμοποιούνται για την αναπαράσταση τον αντικείμενόν στα Recycler view. Αρχικά στην εφαρμογή χρησιμοποιούνται και Horizontal και Vertical Recycler view. Τα Vertical Recycler views είναι 3:

1. Στο activity\_music\_preferences.xml το οποίο αποτελείτε από items που περνούν τη μορφοποίηση τους από το music\_preference\_item.xml .
2. Στο activity\_add\_songs.xml το οποίο αποτελείτε από items που περνούν τη μορφοποίηση τους από το song\_item.xml .
3. Στο activity\_pois.xml το οποίο αποτελείτε από items που περνούν τη μορφοποίηση τους από το poi\_item.xml .

Τα Horizontal Recycler views είναι 3 και όλα στο activity\_main.xml . Τα Recycler views περνούν την μορφοποίηση τους από το cover\_item.xml.

1. Αποθήκευση Δεδομένων.

Η αποθήκευση δεδομένων στην εφαρμογή γίνεται με χρησιμοποιώντας Firebase και Shared Preferences ενώ η αποστολή των δεδομένων από Activity σε Activity γίνεται μέσο Intent Bundle. Ποιο συγκεκριμένα:

* Στη Firebase είναι αποθηκευμένα
  + Οι χρήστες της εφαρμογής οι οποίοι αποθηκεύονται μέσο της λειτουργίας authentication της Firebase.
  + Όλα τα τραγούδια και οι εικόνες των playlist που υποστηρίζονται από την εφαρμογή τα οποία είναι αποθηκευμένα στο Storage της Firebase.
  + Όλες οι playlist που υποστηρίζονται από την εφαρμογή οι οποίες είναι αποθηκευμένες στο Firebase Realtime Database.
* Στα Shared Preferences είναι αποθηκευμένα
* Το profile των χρηστών (UserProfile).