



music worth sharing

Απαιτήσεις Χρηστών και Προδια- γραφές Συστήματος

Del.1.1

Version 0.1

Μαστραλέξη Χριστίνα Μαρία cmastral@ece.auth.gr
Μουρατίδης Αναστάσιος atmourat@ece.auth.gr
Σαουλίδης Νικόλαος saouling@ece.auth.gr
Χαρίτου Ελένη charitoue@ece.auth.gr

Ημερομηνία 11/04/2020

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email
Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Αναστάσιος Μουρατίδης	mob	atmourat@ece.auth.gr
Νικόλαος Σαουλίδης	mob	saouling@ece.auth.gr
Χριστίνα Μαρία Μαστραλέξη	mob	cmastral@ece.auth.gr
Ελένη Χαρίτου	mob	charitoue@ece.auth.gr



Πίνακας Περιεχομένων

Πίνακας Περιεχομένων	3
Λίστα Σχημάτων.....	5
1 Απαιτήσεις Συστήματος	6
1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)	6
1.2 Χρήστες και εξωτερικά συστήματα	9
1.2.1 Χρήστες	9
1.2.2 Εξωτερικά Συστήματα	9
1.3 Σημαντικές μη λειτουργικές απαιτήσεις	10
1.3.1 Απαιτήσεις επίδοσης	10
1.3.2 Απαιτήσεις ασφάλειας (Security)	10
1.3.3 Απαιτήσεις Χρηστικότητα (Usability)	10
1.3.4 Απαιτήσεις πολιτικής (Policy - legal, corporate).....	11
1.4 Λεξικογραφικοί Προσδιορισμοί	11
2 Σενάρια Χρήσης	13
2.1 Διάγραμμα σεναρίων χρήσης.....	13
2.2 Feature: View News Feed	13
2.3 Feature: Search Music or Profiles	14
2.4 Feature: View Profile	15
2.5 Feature: Follow / Unfollow User.....	15
2.6 Feature: Update profile info	16
2.7 Feature: Create Public Playlist	17
2.8 Feature: Make Playlist Private	18
2.9 Feature: Edit Playlist	19
2.10 Feature: Delete Playlist.....	20
2.11 Feature: Control Music Player	21
2.12 Feature: Manage Play Queue	22
2.13 Feature: Upload Music	23
2.14 Feature: View Audience Stats	24
3 Επιδεικτικά γραφικά παράθυρα διεπαφής.....	25
3.1 Σελίδα σύνδεσης και εγγραφής της εφαρμογής.....	25
3.2 Η αρχική σελίδα της εφαρμογής	25
3.3 Σελίδα αναζήτησης και εμφάνισης αποτελεσμάτων	26
3.4 Σελίδα για το προφίλ του ακροατή	26
3.5 Σελίδα για το προφίλ του Καλλιτέχνη	27
3.6 Σελίδα για τη δημιουργία μιας Playlist	28
3.7 Σελίδα για την τροποποίηση μιας Playlist.....	28
3.8 Σελίδα επεξεργασίας Music Player.....	29
3.9 Σελίδα προβολής Play Queue	29



3.10 Σελίδα δημοσίευσης/μεταφόρτωσης μουσικής	30
3.11 Σελίδα τροποποίησης των ρυθμίσεων	30
4 Στατική Μοντελοποίηση	31
4.1 Πακέτα υψηλής προτεραιότητας	31
4.1.1 UserHandle Package	31
4.1.2 HomePage Package	38
4.1.3 Playlist Package	44
4.1.4 UploadMusic Package	52
4.1.5 PlayMusic Package	56
4.2 Πακέτα μεσαίας προτεραιότητας.....	65
4.2.1 Search Package.....	65
4.2.2 Profile Package	71
4.3 Πακέτα χαμηλής προτεραιότητας	80
4.3.1 Statistics Package	80
4.3.2 Settings Package.....	85

Λίστα Σχημάτων

Σχήμα 1: Διάγραμμα σεναρίων χρήσης	13
Σχήμα 2: View News Feed	14
Σχήμα 3: Search Music or Profiles	15
Σχήμα 4: View Profile	15
Σχήμα 5: Follow / Unfollow User	16
Σχήμα 6: Update profile info	17
Σχήμα 7: Create Public Playlist	18
Σχήμα 8: Make Playlist Private	19
Σχήμα 9: Edit Playlist	20
Σχήμα 10: Add song	20
Σχήμα 11: Delete Playlist	21
Σχήμα 12: Control Music Player	22
Σχήμα 13: Manage Play Queue	23
Σχήμα 14: Upload Music	24
Σχήμα 15: View Audience Stats	24

Λίστα Εικόνων

Εικόνα 4.1: Διάγραμμα κλάσεων Πακέτου UserHandle	37
Εικόνα 4.2: Διάγραμμα κλάσεων του Πακέτου	43
Εικόνα 4.3: Διάγραμμα κλάσεων του Πακέτου Playlist	51
Εικόνα 4.4: Διάγραμμα κλάσεων του Πακέτου UploadMusic	55
Εικόνα 4.5: Διάγραμμα κλάσεων του Πακέτου PlayMusic	64
Εικόνα 4.6: Διάγραμμα Κλάσεων Πακέτου Search	70
Εικόνα 4.7: Διάγραμμα Κλάσεων Πακέτου Profile	79
Εικόνα 4.8: Διάγραμμα Κλάσεων Πακέτου Statistics	84
Εικόνα 4.9: Διάγραμμα Κλάσεων Πακέτου Settings	88

1 Απαιτήσεις Συστήματος

1.1 Λειτουργικές απαιτήσεις (Σενάρια χρήσης)

<ΛΑ-1>

Ο χρήστης πρέπει να μπορεί να αναζητήσει μουσική / προφίλ με μια φράση.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να πληκτρολογήσει μια φράση και να εμφανιστεί μια λίστα με τη σχετική μουσική και χρήστες που την περιέχουν.

User Priority (5/5): Η αναζήτηση μουσικής / προφίλ είναι υψίστης σημασίας για τον χρήστη για να μπορέσει να ακούσει/αποθηκεύσει τη μουσική που επιθυμεί και να βρει τα προφίλ που τον ενδιαφέρουν.

Technical Priority (5/5): Η αναζήτηση στην βάση δεδομένων είναι πολύ σημαντική για το σύστημα.

<ΛΑ-2>

Ο χρήστης πρέπει να μπορεί να δημιουργήσει μια Playlist στο προφίλ του.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να δημιουργήσει μια Playlist στο προφίλ του, η οποία είναι από προεπιλογή δημόσια, δίνοντας ένα όνομα, μια περιγραφή, και εάν επιθυμεί, να προσθέσει ένα ή περισσότερα τραγούδια.

User Priority (5/5): Η δημιουργία Playlist είναι υψίστης σημασίας για τον χρήστη, έτσι ώστε να οργανώσει τα τραγούδια κατά την προτίμησή του.

Technical Priority (4/5): Η δημιουργία/διαγραφή Playlist είναι ιδιαίτερα σημαντική για το σύστημα.

<ΛΑ-3>

Ο Ακροατής πρέπει να μπορεί να κάνει μία δημόσια Playlist ιδιωτική.

Περιγραφή: Ο Ακροατής πρέπει να μπορεί να κάνει μία δημόσια Playlist ιδιωτική, δηλαδή να επιλέξει να μην είναι ορατή στο προφίλ του στους υπόλοιπους χρήστες.

User Priority (4/5): Είναι αρκετά σημαντικό ο Ακροατής να μπορεί να επιλέξει αν η λίστα του θα είναι ορατή στους υπόλοιπους χρήστες.

Technical Priority (2/5): Η συγκεκριμένη απαίτηση δεν είναι ιδιαίτερα σημαντική για το σύστημα.

<ΛΑ-4>

Ο χρήστης πρέπει να μπορεί να επεξεργαστεί μια Playlist.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να προσθέσει ή να αφαιρέσει τραγούδια της επιλογής του σε μια Playlist που έχει δημιουργήσει ή να προσθέσει τραγούδι σε μια Playlist που δεν υπάρχει, δημιουργώντας την εκείνη την στιγμή. Επίσης, μπορεί να αλλάξει τα στοιχεία της Playlist, δηλαδή το όνομα και την περιγραφή και τέλος, μπορεί να διαγράψει τη Playlist.

User Priority (5/5): Η επεξεργασία των Playlists του χρήστη είναι υψίστης σημασίας, μιας και του προσφέρει τη δυνατότητα να οργανώνει και να αποθηκεύει τη μουσική σύμφωνα με τις προτιμήσεις του.

Technical Priority (5/5): Η απαίτηση αυτή είναι υψίστης σημασίας για τη λειτουργικότητα του συστήματος.

<ΛΑ-5>

Ο χρήστης πρέπει να μπορεί να ταξινομήσει την ουρά αναπαραγωγής.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να επιλέξει την σειρά με την οποία θα αναπαραχθούν τα κομμάτια της Playlist του, τροποποιώντας έτσι ανά πάσα στιγμή την ουρά αναπαραγωγής.

User Priority (4/5): Η ταξινόμηση της ουράς αναπαραγωγής είναι αρκετά σημαντική για τον χρήστη, μιας και του δίνει τη δυνατότητα να προσαρμόσει τη σειρά αναπαραγωγής των κομματιών μιας ήδη υπάρχουσας λίστας όπως αυτός/ή επιθυμεί.

Technical Priority (3/5): Η απαίτηση αυτή είναι μεσαίας σημασίας για το σύστημα.

<ΛΑ-6>

Ο Ακροατής πρέπει να μπορεί να κάνει Follow / Unfollow κάποιον χρήστη.

Περιγραφή: Ο Ακροατής πρέπει να μπορεί να ακολουθήσει Καλλιτέχνη πατώντας την επιλογή Follow. Επίσης θα μπορεί να αναιρέσει την επιλογή του, πατώντας Unfollow.

User Priority (4/5): Η μη εκπλήρωση μιας τέτοιας λειτουργικότητας δεν είναι καταστροφική, αφού ο Ακροατής μπορεί ακόμα να περιηγείται χειροκίνητα στα προφίλ χρηστών. Ωστόσο, η δυνατότητα να ακολουθεί ο Ακροατής χρήστες δίνει νόημα στην ύπαρξη του *News Feed*, το οποίο είναι αρκετά σημαντικό και χαρακτηριστικό για το Mob.

Technical Priority (4/5): Η απαίτηση αυτή είναι ιδιαίτερα σημαντική για το σύστημα.

<ΛΑ-7>

Ο χρήστης πρέπει να μπορεί να επεξεργαστεί τις πληροφορίες του προφίλ του.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να επεξεργαστεί το προφίλ του, δηλαδή να ενημερώσει φωτογραφία προφίλ, όνομα, περιγραφή και τα προφίλ του από άλλα κοινωνικά δίκτυα.

User Priority (3/5): Η επεξεργασία του προφίλ είναι επιθυμητή για το χρήστη, ώστε να συνδέεται με άτομα που έχουν κοινά ενδιαφέροντα.

Technical Priority (2/5): Η απαίτηση αυτή δεν είναι ιδιαίτερα σημαντική για το σύστημα.

<ΛΑ-8>

Ο Ακροατής πρέπει να μπορεί να λάβει ενημερώσεις για την πρόσφατη δραστηριότητα των χρηστών που ακολουθεί στο News Feed του.



Περιγραφή: Ο Ακροατής θα μπορεί να ενημερώνεται για την προσθήκη καινούργιας μουσικής από καλλιτέχνες ή την δημιουργία/ενημέρωση μιας Playlist από άλλους χρήστες που ακολουθεί.

User Priority (4/5): Αυτή η λειτουργία είναι πολύ σημαντική για τον Ακροατή, μιας και του δίνει εύκολη και γρήγορη πρόσβαση στη νέα δραστηριότητα χρηστών που τον ενδιαφέρουν.

Technical Priority (3/5): Η απαίτηση αυτή είναι μεσαίας σημασίας για το σύστημα.

<ΛΑ-9>

Ο Καλλιτέχνης πρέπει να μπορεί να προσθέσει μουσική.

Περιγραφή: Ο Καλλιτέχνης πρέπει να μπορεί να προσθέσει μουσική στο προφίλ του, η οποία καθίσταται διαθέσιμο προς ακρόαση για τους χρήστες.

User Priority (5/5): Η προσθήκη νέας μουσικής από τους Καλλιτέχνες είναι ένα από τα βασικά στοιχεία της λειτουργικότητας της εφαρμογής και είναι απαραίτητη ώστε να έχει νόημα η δημιουργία της πλατφόρμας.

Technical Priority (5/5): Η απαίτηση αυτή είναι υψίστης προτεραιότητας καθώς είναι προϋπόθεση για την επιθυμητή λειτουργία της εφαρμογής.

<ΛΑ-10>

Ο Καλλιτέχνης πρέπει να μπορεί να βλέπει τα στατιστικά για το ακροατήριό του.

Περιγραφή: Ο Καλλιτέχνης πρέπει να μπορεί να βλέπει τα στατιστικά σχετικά με τη μουσική του, δηλαδή τον αριθμό των ακροάσεων κάθε κομματιού και ποσοστά σχετικά με την ηλικία, φύλο και χώρα προέλευσης των ακροατών.

User Priority (4/5): Τα στοιχεία αυτά για τον Καλλιτέχνη αντικατοπτρίζουν την απήχηση του στους χρήστες της εφαρμογής, επομένως πρόκειται για αναγκαία λειτουργία.

Technical Priority (3/5): Η απαίτηση αυτή είναι μεσαίας σημασίας για το σύστημα.

<ΛΑ-11>

Ο χρήστης πρέπει να μπορεί να χειρίζεται το Music Player.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να επιλέγει αναπαραγωγή / παύση / επόμενο / προηγούμενο και να μετακινεί το Slider στο σημείο αναπαραγωγής που επιθυμεί.

User Priority (5/5): Η απαίτηση αυτή είναι εξαιρετικά σημαντική για τον χρήστη γιατί του επιτρέπει να ελέγχει την αναπαραγωγή της μουσικής.

Technical Priority (5/5): Η απαίτηση αυτή είναι υψίστης προτεραιότητας για το σύστημα.

<ΛΑ-12>

Ο χρήστης πρέπει να μπορεί να ακούσει μουσική πατώντας τον τίτλο.

Περιγραφή: Ο χρήστης πατώντας τον τίτλο του τραγουδιού/ άλμπουμ / Playlist που επιθυμεί, γίνεται αναπαραγωγή του, αφού προστίθεται αυτόματα στην ουρά αναπαραγωγής και στο Music Player.

User Priority (5/5): Η απαίτηση αυτή είναι εξαιρετικά σημαντική για τον χρήστη γιατί του επιτρέπει να ελέγχει την αναπαραγωγή της μουσικής.

Technical Priority (4/5): Η απαίτηση αυτή είναι ιδιαίτερα σημαντική για το σύστημα.

<ΛΑ-13>

Ο χρήστης πρέπει να μπορεί να δει το προφίλ άλλων χρηστών.

Περιγραφή: Ο χρήστης πρέπει να μπορεί να δει το προφίλ άλλων χρηστών πατώντας πάνω στο όνομα του εκάστοτε χρήστη.

User Priority (5/5): Η απαίτηση αυτή είναι εξαιρετικά σημαντική για τον χρήστη, γιατί του επιτρέπει να περιηγείται στα προφίλ των χρηστών που αναζητεί.

Technical Priority (4/5): Η απαίτηση αυτή είναι αρκετά σημαντική για το σύστημα.

1.2 Χρήστες και εξωτερικά συστήματα

Κατά τη διάρκεια της εγγραφής τους οι χρήστες χωρίζονται σε Listeners και Artists. Και για τις δύο κατηγορίες έχει υλοποιηθεί μια βασικά κοινή λειτουργικότητα. Ο κάθε χρήστης όμως βάσει του ρόλου του έχει και δικά του προνόμια και λειτουργίες σε σχέση με τον άλλον.

1.2.1 Χρήστες

➤ Χρήστης - User

Τα άτομα που εγγράφονται στο Mob. Μοντελοποιεί την προαναφερθείσα κοινή λειτουργικότητα.

➤ Ακροατής

Η κατηγορία αυτή, περιλαμβάνει τα άτομα που μπορούν να δημιουργήσουν Playlists (ιδιωτικές ή δημόσιες) στο προφίλ τους, να ακολουθούν άλλους χρήστες και να λαμβάνουν ενημερώσεις πρόσφατης δραστηριότητας στο News Feed τους για αυτούς.

➤ Καλλιτέχνης

Η κατηγορία αυτή αναφέρεται στους μουσικούς που χρησιμοποιούν το Mob ως πλατφόρμα προώθησης της δουλειάς τους. Αυτοί έχουν τη δυνατότητα να ανεβάζουν τραγούδια / άλμπουμ και να δημιουργούν Playlists, όπως και να έχουν πρόσβαση σε στοιχεία που αφορούν το ακροατήριό τους.

1.2.2 Εξωτερικά Συστήματα

Βάση Δεδομένων

Είναι απαραίτητο εξωτερικό σύστημα για την ύπαρξη της εφαρμογής, καθώς σε αυτήν αποθηκεύονται και υπάρχουν όλα τα τραγούδια / άλμπουμ / Playlist, τα οποία θα ακούν και θα διαχειρίζονται οι χρήστες. Περιλαμβάνει, επίσης, τα στοιχεία των χρηστών και διασυνδέεται με κατάλληλο λογισμικό για την εξαγωγή των στατιστικών αποτελεσμάτων που θα εμφανίζονται στους καλλιτέχνες.

1.3 Σημαντικές μη λειτουργικές απαιτήσεις

1.3.1 Απαιτήσεις επίδοσης

<ΜΛΑ-1>

Το σύστημα πρέπει να μπορεί να υποστηρίξει 50000 χρήστες εξασφαλίζοντας μέγιστο χρόνο απόκρισης 200 ms, σε μια LTE σύνδεση.

Περιγραφή: Οι ενέργειες του χρήστη πρέπει να έχουν αποτέλεσμα σε λιγότερο από 200 ms έτσι ώστε να είναι ομαλή η περιήγησή του στο Mob.

User Priority (5/5): Η εμπειρία του χρήστη κατά την περιήγηση και την ακρόαση μουσικής στο Mob είναι βασική προτεραιότητα.

Technical Priority (5/5): Η απαίτηση αυτή επιδρά σε όλες τις παραμέτρους του συστήματος. Ο αριθμός των ταυτόχρονων χρηστών αναμένεται αρκετά υψηλός, οπότε το bandwidth πρέπει να είναι αρκεδόντως μεγάλο για να υποστηρίξει μια streaming υπηρεσία σαν το Mob.

1.3.2 Απαιτήσεις ασφάλειας (Security)

<ΜΛΑ-2>

Το σύστημα πρέπει να εντοπίζει και να αποτρέπει μη εξουσιοδοτημένες προσπάθειες εισόδου.

Περιγραφή: Το σύστημα πρέπει να εντοπίζει και να αποτρέπει προσπάθειες εισόδου κατά την ταυτοποίηση του χρήστη (λάθος συνδυασμός email / username - κωδικού πρόσβασης)

User Priority (5/5): Η προστασία του λογαριασμού είναι υψίστης σημασίας για την διαφύλαξη των προσωπικών δεδομένων του χρήστη.

Technical Priority (4/5): Πρέπει να προβλεφθεί η χρήση τεχνικών προστασίας από μη εξουσιοδοτημένες προσπάθειες εισόδου.

1.3.3 Απαιτήσεις Χρηστικότητας (Usability)

<ΜΛΑ-3>

Το σύστημα πρέπει να έχει ικανότητα απομνημόνευσης της τελευταίας κατάστασης πριν τον τερματισμό της εφαρμογής.

Περιγραφή: Το σύστημα πρέπει να απομνημονεύει τις προσωπικές ρυθμίσεις του χρήστη, όπως την τελευταία κατάσταση του music player (τρέχον τραγούδι και λίστα / ουρά αναπαραγωγής)

User Priority (4/5): Η αποθήκευση της τελευταίας κατάστασης είναι πολύ σημαντική από πλευράς του χρήστη, καθώς του δίνει τη δυνατότητα να συνεχίσει τη χρήση του Mob από την τελευταία του έξοδο από την εφαρμογή, χωρίς να χάσει χρόνο. Επίσης, καλύπτει την περίπτωση ανεπιθύμητου τερματισμού της εφαρμογής λόγω κάποιου εξωτερικού σφάλματος.

Technical Priority (3/5): Το σύστημα πρέπει να αποθηκεύει σε μια προσωρινή τοποθεσία ανά πάσα στιγμή την κατάσταση στην οποία βρίσκεται η εφαρμογή. Η απαίτηση αυτή είναι μεσαίας σημασίας για το σύστημα.

**<ΜΛΑ-4>**

Ο ρυθμός των σφαλμάτων κατά την αναπαραγωγή ενός τραγουδιού δεν θα πρέπει να ξεπερνά το 1%.

Περιγραφή: Το ποσοστό των σφαλμάτων κατά την αναπαραγωγή θα πρέπει να είναι το ελάχιστο δυνατό.

User Priority (5/5): Αυτή η απαίτηση είναι πολύ σημαντική για τον χρήστη, καθώς ένας μεγαλύτερος ρυθμός σφαλμάτων μπορεί να καταστρέψει την εμπειρία του.

Technical Priority (5/5): Αυτή η απαίτηση είναι υψίστης σημασίας για το σύστημα.

1.3.4 Απαιτήσεις πολιτικής (Policy - legal, corporate)**<ΜΛΑ-5>**

Η πολιτική απορρήτου των προσωπικών δεδομένων του χρήστη ορίζεται από το άρθρο 15 του GDPR.

Περιγραφή: Τα δεδομένα των χρηστών προστατεύονται από το άνωθεν άρθρο και προορίζονται αποκλειστικά για την εξαγωγή ανώνυμων στατιστικών για τους καλλιτέχνες που προωθούν τη μουσική τους μέσω του Mob. Ο χρήστης μπορεί να ενημερωθεί σχετικά με το άρθρο στην ενότητα 'Privacy Policy' των ρυθμίσεων της εφαρμογής.

User Priority (5/5): Είναι προτεραιότητα να διασφαλιστεί ότι ο χρήστης κατανοεί ποια προσωπικά δεδομένα που τον αφορούν συλλέγονται, τους λόγους για τους οποίους συλλέγονται και χρησιμοποιούνται. Επίσης, είναι σημαντικό να γίνουν κατανοητά τα δικαιώματα και οι επιλογές των χρηστών σχετικά με τα προσωπικά δεδομένα που τους αφορούν και τα οποία συλλέγονται και υπόκεινται σε επεξεργασία.

Technical Priority (1/5): Αυτή η απαίτηση δεν επηρεάζει με κάποιο τρόπο το σύστημα, άρα είναι ασήμαντη ως προς αυτό.

1.4 Λεξικογραφικοί Προσδιορισμοί**< Αρχική σελίδα >**

Είναι η σελίδα που εμφανίζεται όταν ο χρήστης ανοίγει την εφαρμογή, η οποία περιλαμβάνει την μπάρα αναζήτησης και το News Feed.

< Προφίλ >

Είναι η σελίδα στην οποία βρίσκονται οι προσωπικές πληροφορίες του χρήστη και Playlists που έχει δημιουργήσει ο χρήστης. Στο προφίλ του Καλλιτέχνη, υπάρχει η επιλογή να προσθέσει μουσική και να προβάλλει τα στατιστικά του.

< Follow(Ακολουθώ) >

Ο χρήστης μπορεί να ακολουθήσει άλλους χρήστες και καλλιτέχνες, κάτι που σημαίνει ότι θα βρίσκονται στη λίστα των following και η πρόσφατη δραστηριότητά τους θα εμφανίζεται στο News Feed στην αρχική σελίδα.

< Μπάρα αναζήτησης >

Η μπάρα αναζήτησης είναι ένα πλαίσιο όπου ο χρήστης μπορεί να πληκτρολογήσει μια λέξη και να εμφανιστούν τα σχετικά αποτελέσματα.

< Προσθέτω / Αφαιρώ >

Η λειτουργία “προσθέτω” είναι διαθέσιμη για οποιοδήποτε τραγούδι και έχει ως αποτέλεσμα την προσθήκη και αποθήκευση του συγκεκριμένου τραγουδιού σε μια ή περισσότερες Playlist του προφίλ χρήστη. Αντίστοιχα, η λειτουργία “αφαιρώ” είναι διαθέσιμη για όλα τα κομμάτια στις Playlists του χρήστη και συνεπάγεται τη διαγραφή του συγκεκριμένου τραγουδιού από μια συγκεκριμένη Playlist.

< Playlist >

Αποτελείται από ένα σύνολο τραγουδιών, με συγκεκριμένη σειρά. Έχει υποχρεωτικά τίτλο, και προαιρετικά περιγραφή και φωτογραφία.

< News Feed >

Το News Feed είναι ένα μέρος της “Αρχικής σελίδας” στο οποίο ο χρήστης λαμβάνει ενημερώσεις για την πρόσφατη δραστηριότητα των καλλιτεχνών και των χρηστών που ακολουθεί.

<Πρόσφατη Δραστηριότητα >

Περιλαμβάνει τις τελευταίες ενέργειες του χρήστη σχετικά με την ενημέρωση δημοσίων Playlist και τις τελευταίες μεταφορτώσεις του κάθε Καλλιτέχνη.

< Ουρά αναπαραγωγής >

Είναι η σειρά με την οποία θα αναπαραχθούν τα επόμενα τραγούδια και καθορίζεται από τη δυνατότητα του χρήστη να αναδιατάξει τα τραγούδια στην ουρά χειροκίνητα, να προσθέσει και να αφαιρέσει τραγούδια από αυτήν.

< Μουσική >

Είναι τα τραγούδια μεμονωμένα, τα άλμπουμ και οι Playlists.

< Music Player >

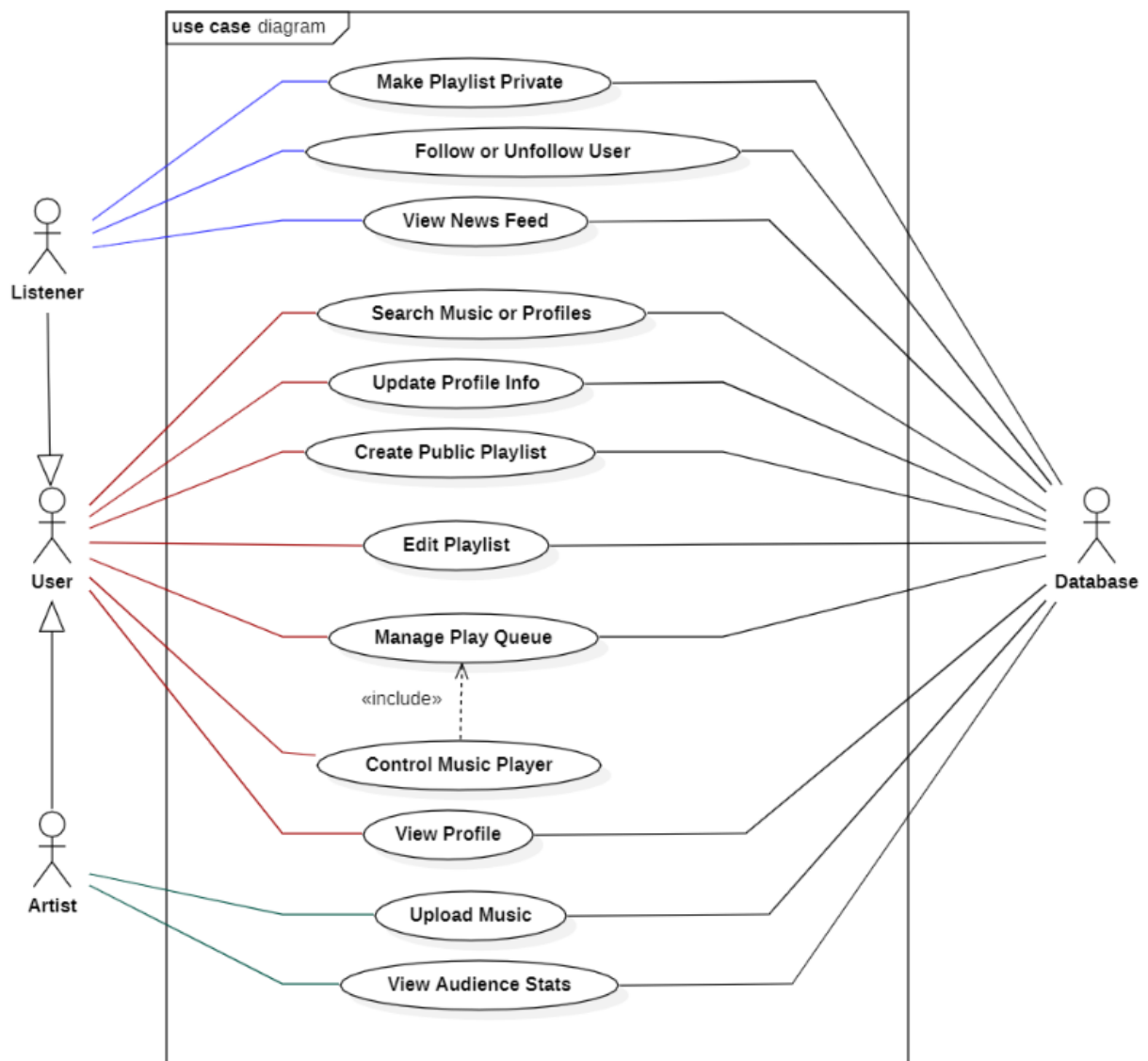
Πρόκειται για μια γραμμή εργαλείων της αναπαραγωγής, με την οποία ο χρήστης μπορεί ανά πάσα στιγμή να δει τον τίτλο του τραγουδιού που ακούει, να πατήσει παύση, συνέχιση αναπαραγωγής, επόμενο τραγούδι, προηγούμενο τραγούδι αλλά και να επιστρέψει/προχωρήσει σε οποιοδήποτε σημείο του τραγουδιού. Επιπλέον, μπορεί να επιλέγει ανάμεσα σε τυχαία ή μη αναπαραγωγή (Shuffle / Don't Shuffle) και σε επανάληψη ενός μεμονωμένου τραγουδιού (Repeat Track / Don't Repeat).

< Slider >

Είναι το εργαλείο του Music Player το οποίο δείχνει στον χρήστη κάθε στιγμή της αναπαραγωγής σε ποιο δευτερόλεπτο βρίσκεται και του δίνει την δυνατότητα να επιστρέψει/προχωρήσει σε οποιοδήποτε σημείο του τραγουδιού επιθυμεί με την κύλιση αυτού.

2. Σενάρια Χρήσης

2.1 Διάγραμμα σεναρίων χρήσης



Σχήμα 1: Διάγραμμα σεναρίων χρήσης

2.2 Feature: View News Feed

Scenario: View News Feed

Given that I am a Listener

And I am following at least one user

When I click "Home" to see the News Feed

Then I see the recent activity of the users I follow

Scenario: Blank News Feed

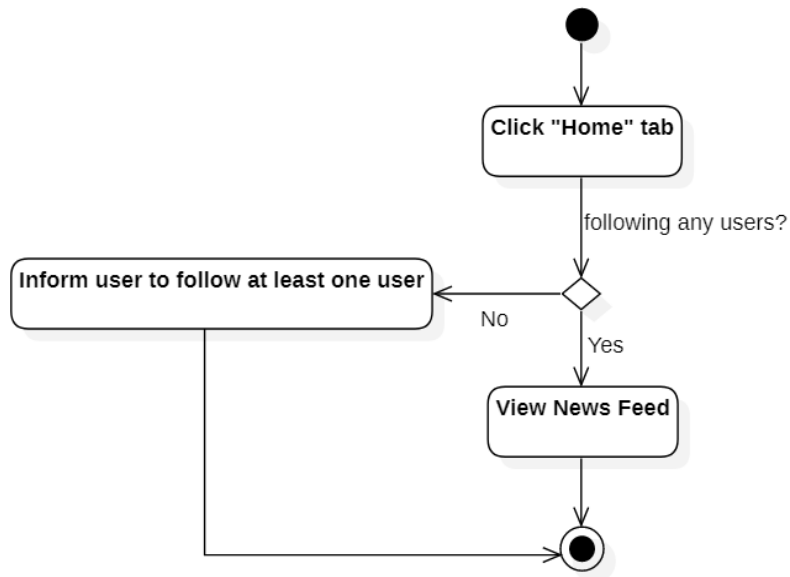
Given that I am a Listener

And I am not following other users

When I click "Home" to see the News Feed

Then News Feed is blank

And I am informed that I am not following any users



Σχήμα 2: View News Feed

2.3 Feature: Search Music or Profiles

Scenario: Search Music or Profiles and display results

Given that I am a User

And I am in "Home"

When I search Music or Profiles entering text in the Search Bar

And there are results that match with my search

Then I see the results

Scenario: Search non-existent Music or Profiles

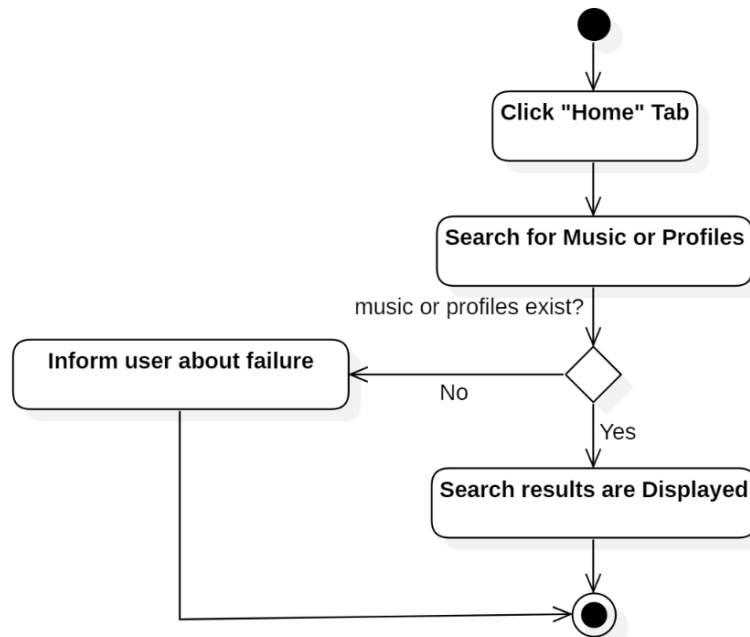
Given that I am a User

And I am in "Home"

When I search Music or Profiles entering text in the Search Bar

But there aren't results that match with my search

Then I am informed that there are no results for my search



Σχήμα 3: Search Music or Profiles

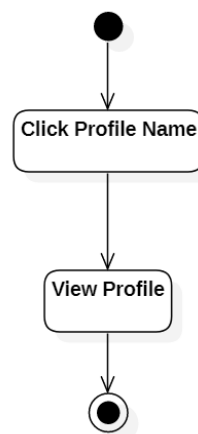
2.4 Feature: View Profile

Scenario: Display Profile

Given that I am a User

When I click to a Profile name

Then I view this Profile



Σχήμα 4: View Profile

2.5 Feature: Follow / Unfollow User

Scenario: Follow/Unfollow User

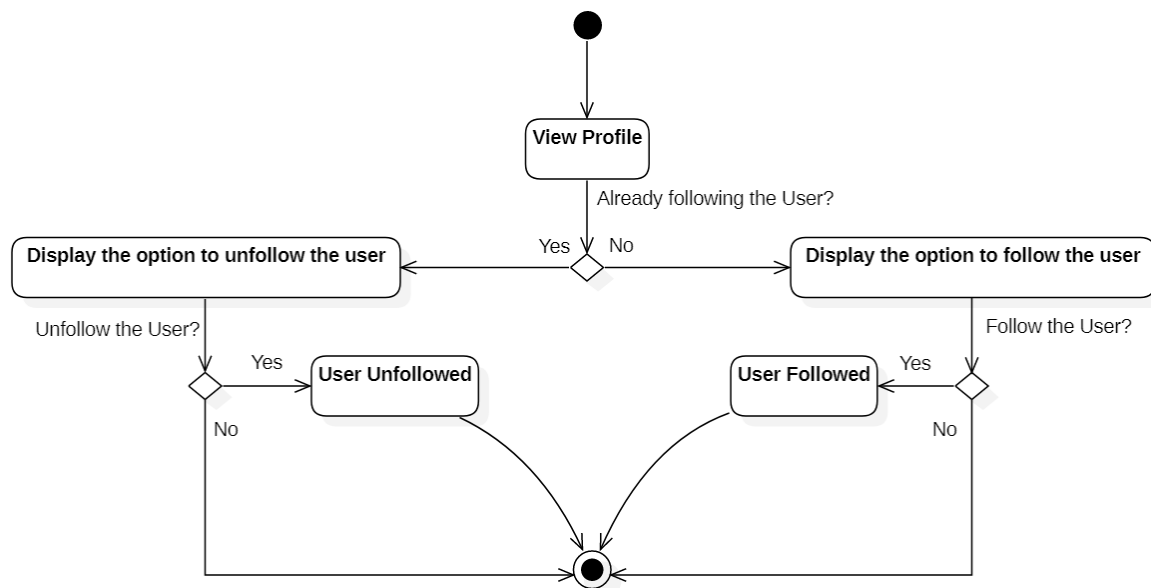
Given that I am a Listener

When I view a Profile

Then I have the options to "Follow" / "Unfollow" this specific user

When I click "Follow" / "Unfollow" this User

Then I am informed that I successfully followed or unfollowed this User



Σχήμα 5: Follow / Unfollow User

2.6 Feature: Update profile info

Scenario: Update profile info

Given that I am a user

When I click to update my profile info

And I provide valid info

Then my profile info is updated

And I am informed that I have successfully updated my profile info

Scenario: Invalid info (email, password, profile photo)

Given that I am a user

When I click to update my profile info

But I provide invalid info (wrong email format, weak password, wrong type for profile photo)

Then I am informed that the info I provided is invalid

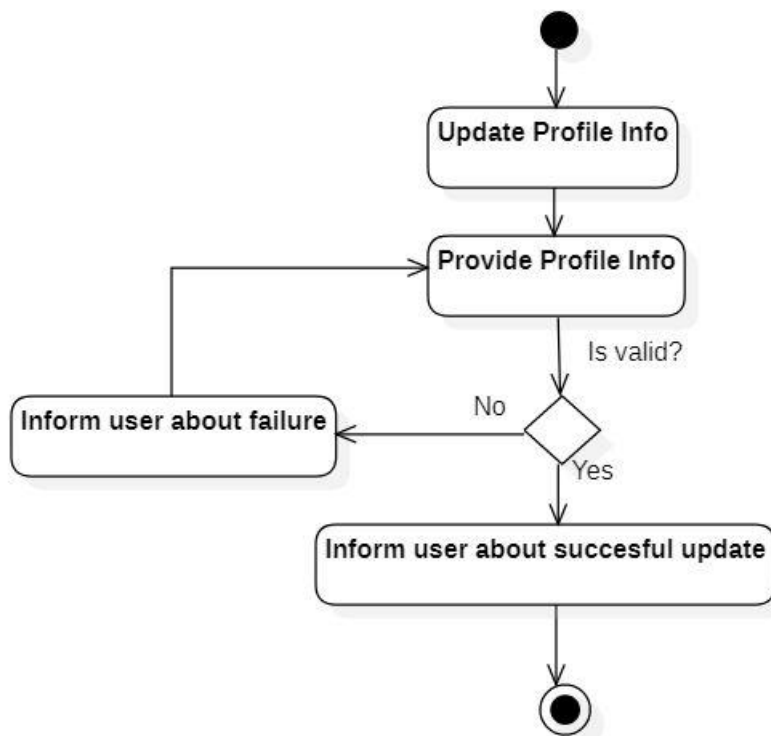
Scenario: Invalid username (already exists, invalid characters)

Given that I am a user

When I click to change my username

But I provide invalid username (already exists, invalid characters)

Then I am informed that the username I provided is invalid



Σχήμα 6: Update profile info

2.7 Feature: Create Public Playlist

Scenario: Create Public Playlist

Given that I am a User

When I click "Create New Playlist"

Then I am prompted to enter a Playlist name and a description

When I enter a valid Playlist name and a description

Then a Playlist is created

And I am informed that the Playlist was successfully created

Scenario: Provide empty Playlist name

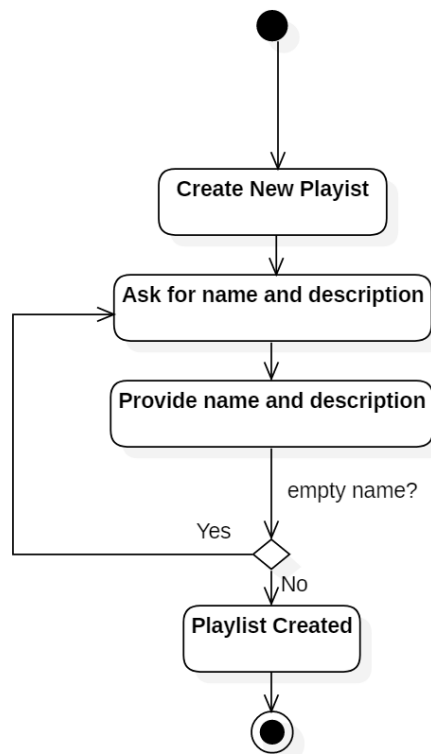
Given that I am a User

When I click "Create New Playlist"

Then I am prompted to enter a Playlist name and description

But I enter an empty Playlist name

Then I am informed that I must provide a Playlist name containing at least one character



Σχήμα 7: Create Public Playlist

2.8 Feature: Make Playlist Private

Scenario: Make a public playlist private

Given that I am a Listener

When I click to make a Playlist Private

And this Playlist is Public

Then this Playlist becomes Private

And I am informed that the Playlist is now Private

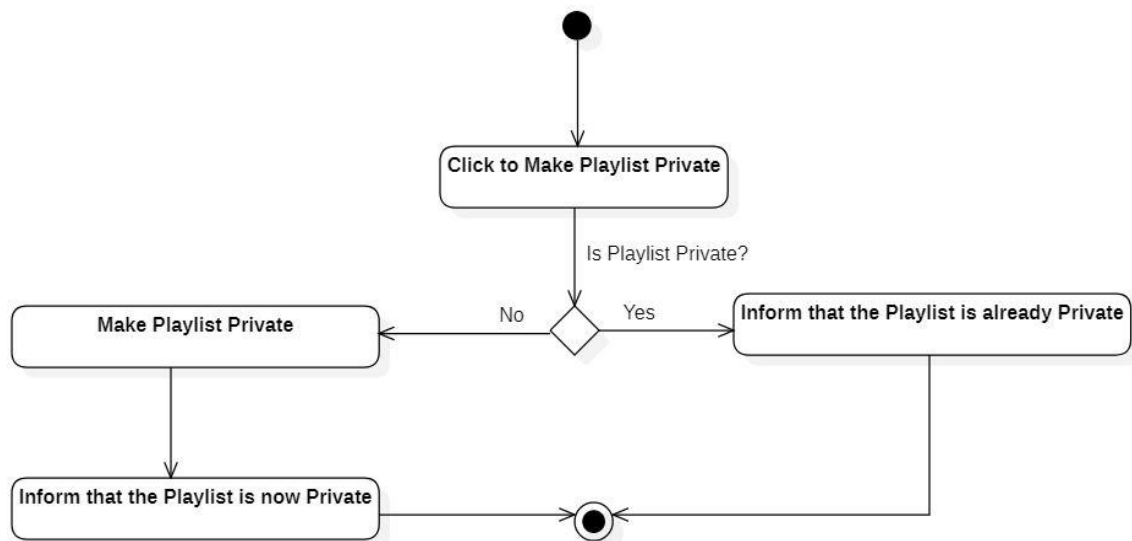
Scenario: Make an already private playlist private

Given that I am a Listener

When I click to make a Playlist Private

And this Playlist is Private

Then I am informed that this Playlist is already Private



Σχήμα 8: Make Playlist Private

2.9 Feature: Edit Playlist

Scenario: Edit name/ description of the Playlist

Given that I am a User

When I click the “Edit Playlist” option on a Playlist on my profile

Then I am prompted to change the name or the description of this Playlist

When I enter a valid Playlist name and a description

Then this Playlist is modified

And I am informed that the Playlist was successfully modified

Scenario: Add song in the Playlist

Given that I am a User

And I view a song

When I click the “Add to Playlist” option

Then I should see my Playlists

When I choose a Playlist

Then the song is added to this Playlist

And this Playlist is modified

Scenario: Delete song of the Playlist

Given that I am a User

And I view a Playlist

When I click the “Delete song” option

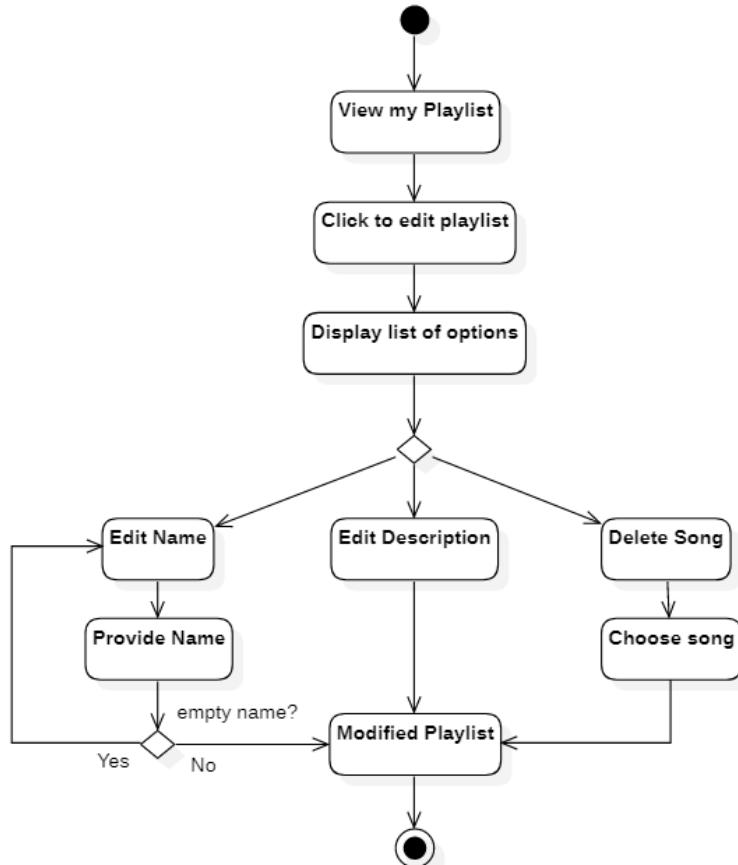
Then the song is deleted from this Playlist

And this Playlist is modified

Scenario: Provide empty Playlist name

Given that I am a User

When I click the “Edit Playlist” option on a Playlist on my profile
Then I am prompted to change the name or the description of this Playlist
But I enter an empty Playlist name
Then I am informed that I must provide a Playlist name containing at least one character



Σχήμα 9: Edit Playlist



Σχήμα 10: Add song

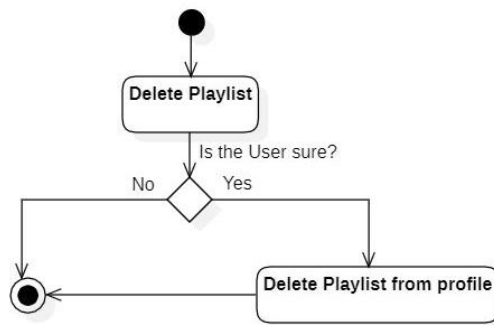
2.10 Feature: Delete Playlist

Scenario: Delete Playlist

Given that I am a User
When I click the “Delete Playlist” option on a Playlist on my profile
Then I am asked if I want to Delete this Playlist
When I choose “Yes”
Then this Playlist is deleted
And I am informed that this Playlist was successfully deleted

Scenario: Cancel Delete Process

Given that I am a User
When I click the “Delete Playlist” option on a Playlist on my profile
Then I am asked if I want to Delete this Playlist
When I choose “No”
Then I am redirected to my Profile



Σχήμα 11: Delete Playlist

2.11 Feature: Control Music Player

Scenario: Change Repeat Option

Given that I am a User

When I click to change the Repeat Option

Then a menu appears

And I have the options to activate “Repeat Track” / “Don’t Repeat”

When I click one of the options

Then the Play Queue is rearranged

Scenario: Change Shuffle Option

Given that I am a User

And I have the options to activate “Shuffle” / “Don’t Shuffle

When I click one of the options

Then the Play Queue is rearranged

Scenario: Change Play / Pause Option

Given that I am a User

When I click to change the Play / Pause Option

Then music plays / pauses respectively

Scenario: Next/Previous Song

Given that I am a User

When I click to listen the Next / Previous Song

Then I listen the Next / Previous Song

Scenario: Manage slider

Given that I am a User

When I click a specific moment on the slider

Then music continues playing from this moment

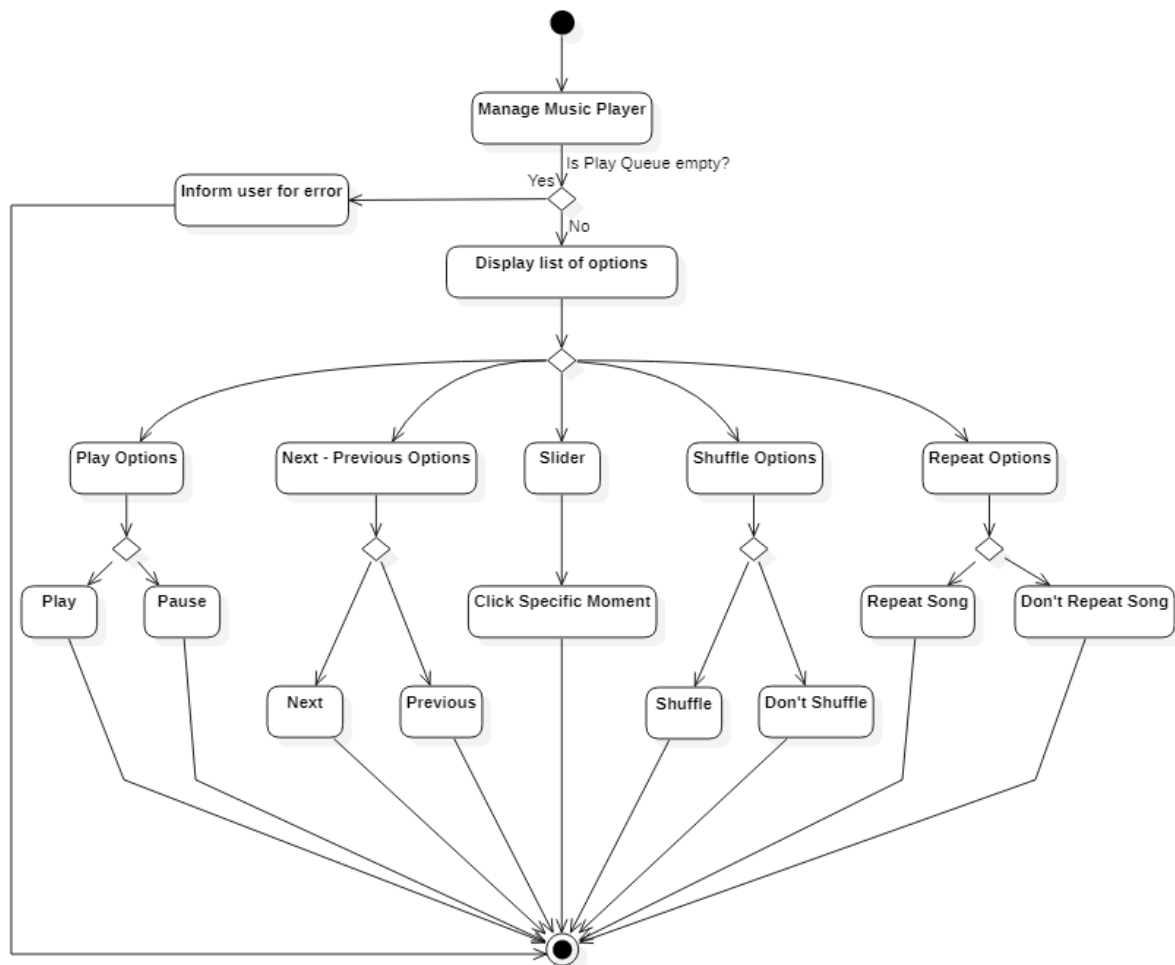
Scenario: Play Queue is empty

Given that I am a User

And the Play Queue is empty

When I want to control Music Player

Then I am informed that the Play Queue is empty



Σχήμα 12: Control Music Player

2.12 Feature: Manage Play Queue

Scenario: Rearrange Play Queue

Given that I am a User

And the Play Queue is not empty

When I click to change the Play Queue

Then I should see a list of options

And I choose to rearrange the Play Queue

Scenario: Edit Play Queue

Given that I am a User

And the Play Queue is not empty

When I click to change the Play Queue

Then I should see a list of options

And I choose to edit the Play Queue

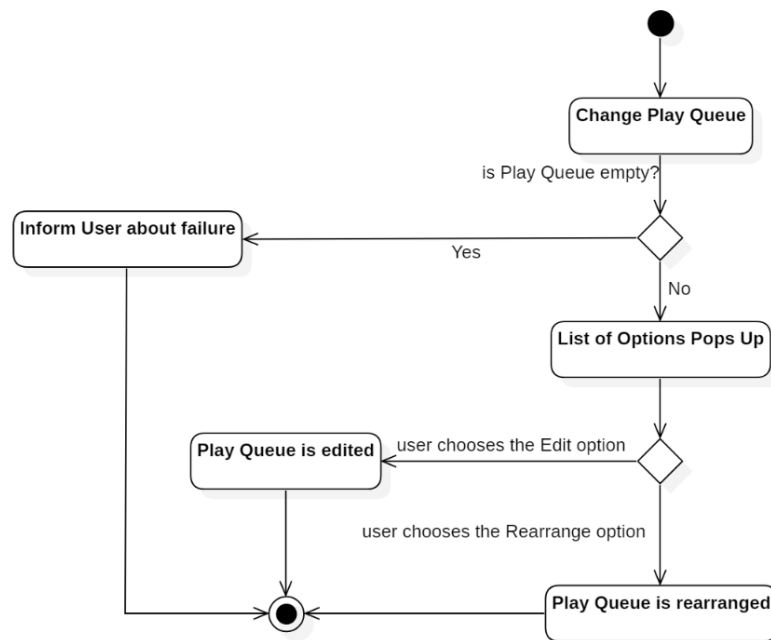
Scenario: Play Queue is empty

Given that I am a User

When I click to change the Play Queue

And the Play Queue is empty

Then I am informed that the Play Queue is empty



Σχήμα 13: Manage Play Queue

2.13 Feature: Upload Music

Scenario: Upload Music

Given that I am an Artist

When I click to upload music

And I provide the correct file type

Then the music is uploaded on my profile

And I am informed that music is successfully uploaded

Scenario: Wrong File Type

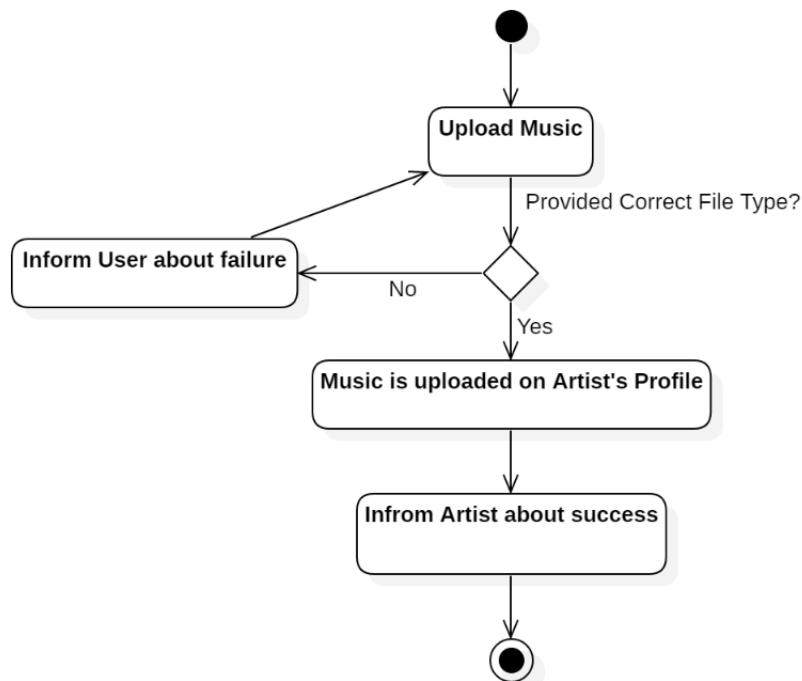
Given that I am an Artist

When I click to upload music

And I provide the wrong file type

Then I am informed that I provided wrong file type

And I am redirected to "upload music" again



Σχήμα 14: Upload Music

2.14 Feature: View Audience Stats

Scenario: View Stats

Given that I am an Artist

When I click to view Audience Stats

Then I view statistics about my audience

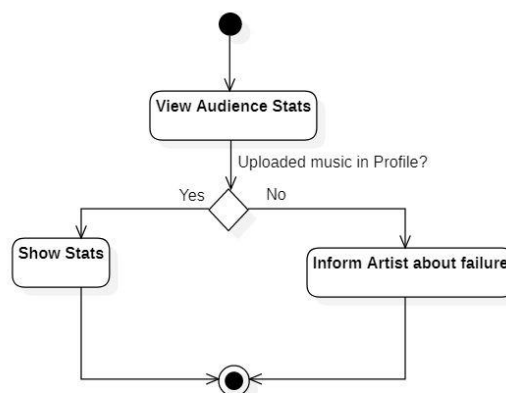
Scenario: No uploaded music

Given that I am an Artist

And I haven't uploaded music yet

When I click to view Audience Stats

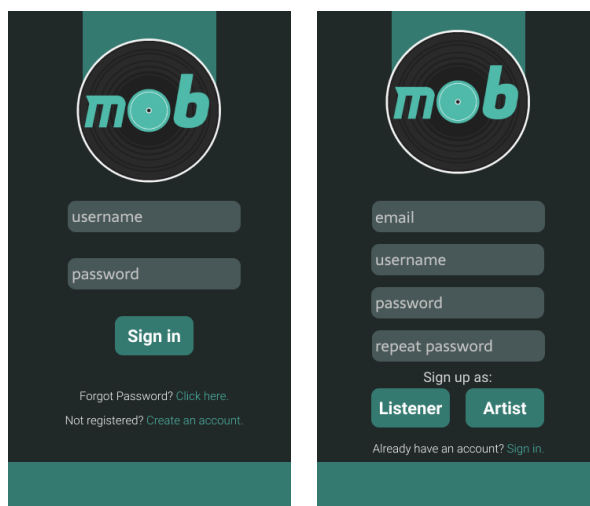
Then I am informed that I haven't uploaded any music yet



Σχήμα 15: View Audience Stats

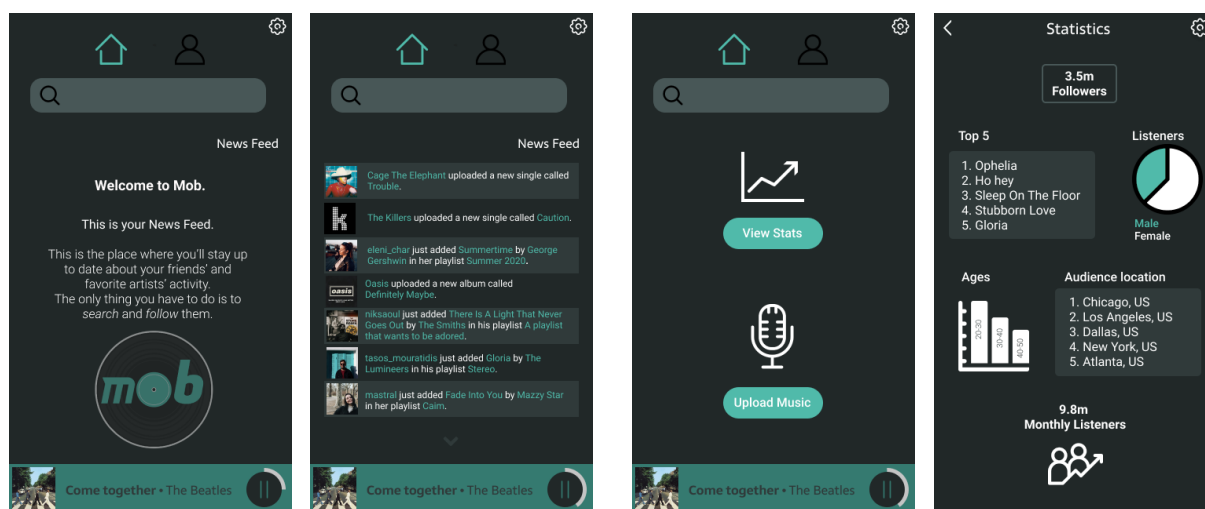
3 Επιδεικτικά γραφικά παράθυρα διεπαφής

3.1 Σελίδα σύνδεσης και εγγραφής της εφαρμογής



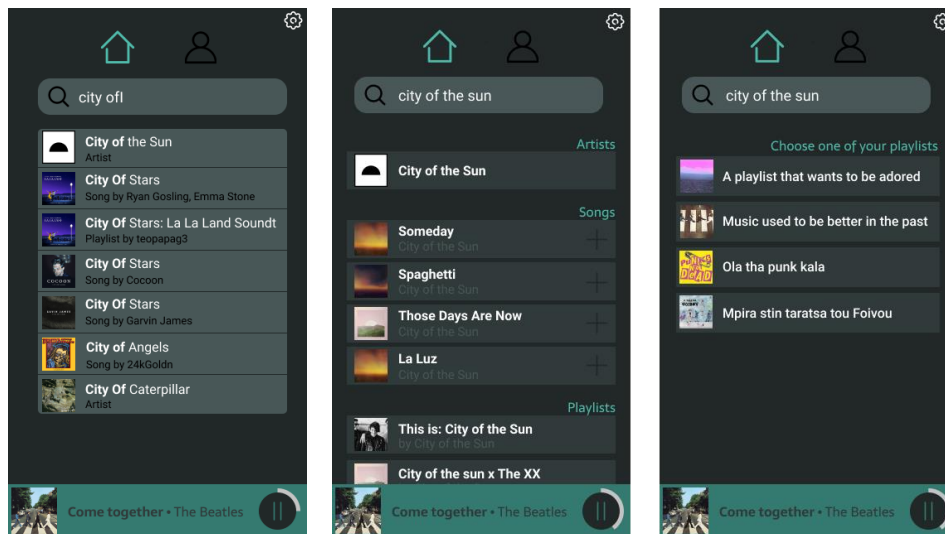
Εικόνα 3.1: Η σελίδα στην οποία εγγράφονται ή συνδέονται οι χρήστες

3.2 Η αρχική σελίδα της εφαρμογής



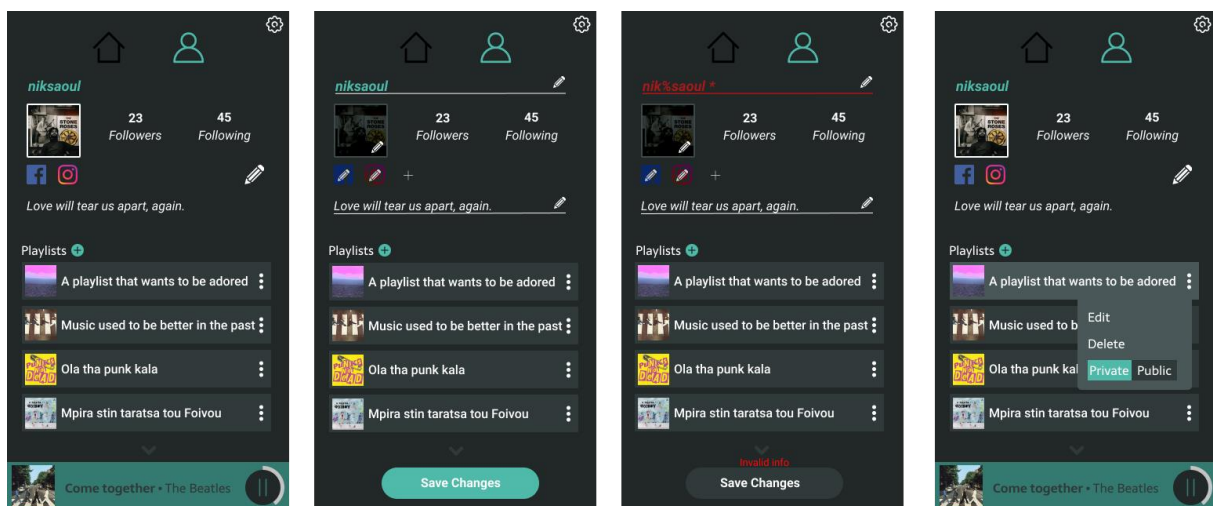
Εικόνα 3.2: Η αρχική σελίδα της εφαρμογής όταν ο Ακροατής δεν ακολουθεί κανέναν άλλο χρήστη και όταν ακολουθεί. Η αρχική σελίδα του Καλλιτέχνη και η στατιστική προβολή της απήχησής του

3.3 Σελίδα αναζήτησης και εμφάνισης αποτελεσμάτων

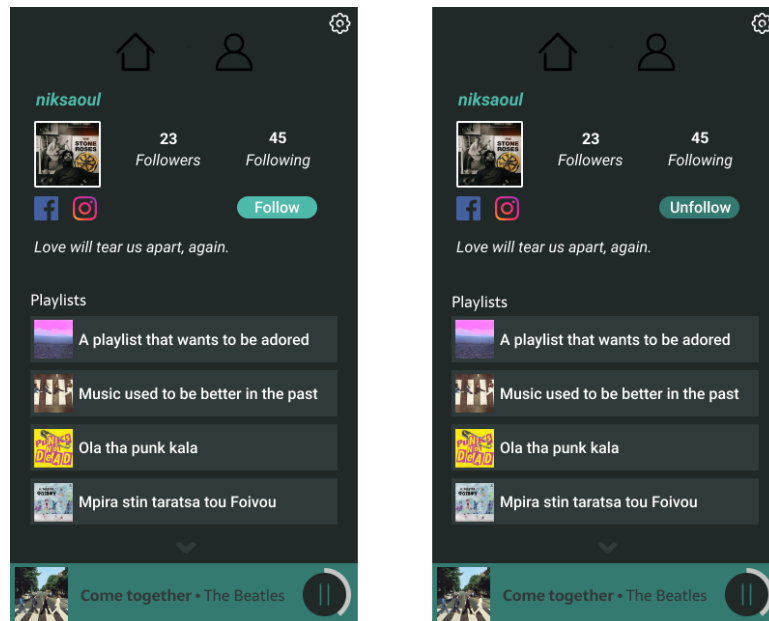


Εικόνα 3.3: Σελίδα αναζήτησης για city of και αποτελέσματα αναζήτησης για city of the sun. Επιλογή Playlist πατώντας το «+» για προσθήκη τραγουδιού.

3.4 Σελίδα για το προφίλ του Ακροατή

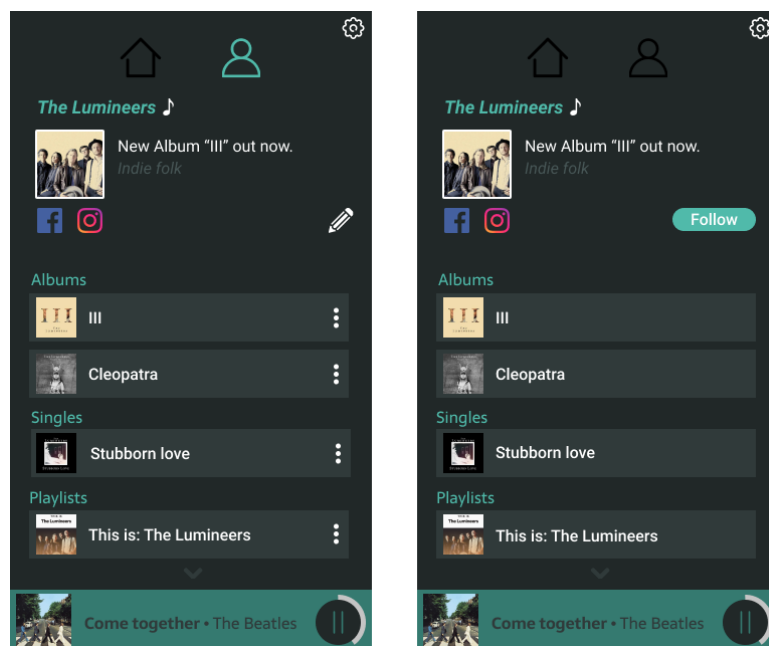


Εικόνα 3.4: Το προφίλ του Ακροατή όπως το βλέπει ο ίδιος, όταν το τροποποιεί, όταν κάνει κάποια μη αποδεκτή τροποποίηση και όταν επρόκειτο να επεξεργαστεί μία από τις λίστες του



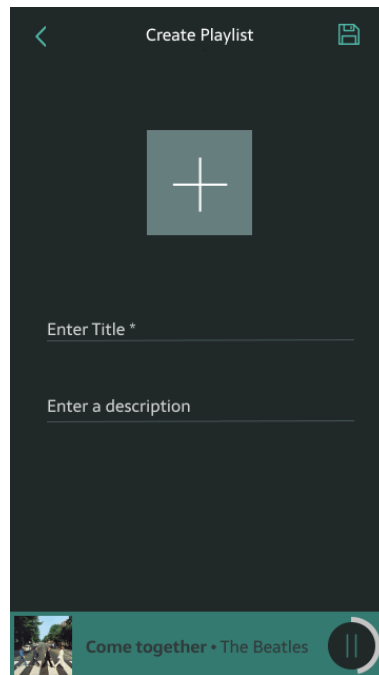
Εικόνα 3.5: Το προφίλ του Ακροατή όπως το βλέπει κάποιος άλλος χρήστης που δεν τον ακολουθεί και έχει τη δυνατότητα να το ακολουθήσει ή που τον ακολουθεί και επιθυμεί να μην τον ακολουθεί πια

3.5 Σελίδα για το προφίλ του Καλλιτέχνη



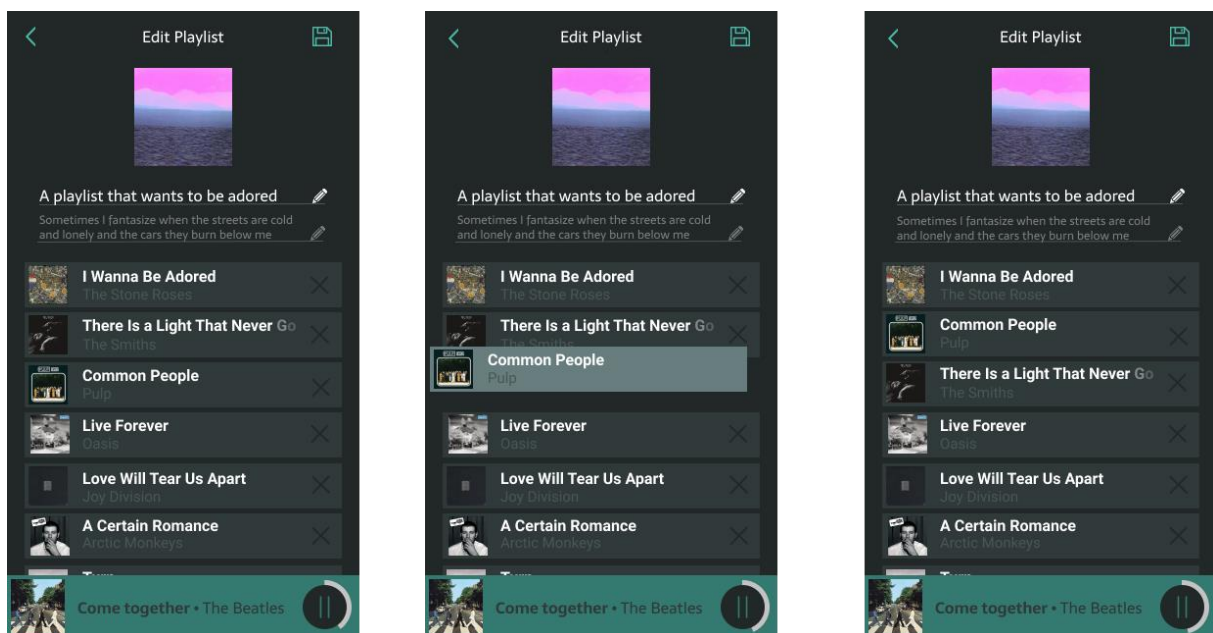
Εικόνα 3.6: Το προφίλ του Καλλιτέχνη όπως το βλέπει ο ίδιος και όπως το βλέπει ένας άλλος χρήστης

3.6 Σελίδα για τη δημιουργία μιας Playlist



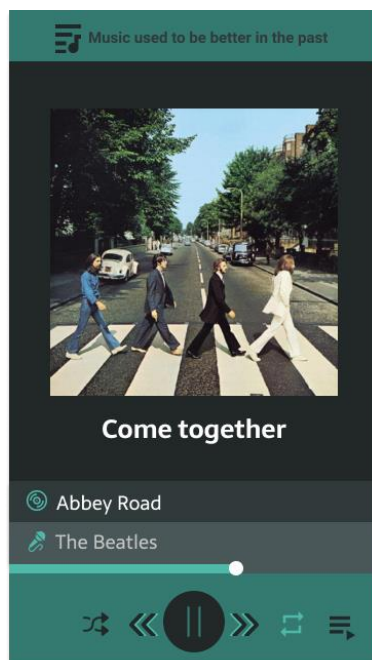
Εικόνα 3.7: Δημιουργία μίας Playlist

3.7 Σελίδα για την τροποποίηση μιας Playlist



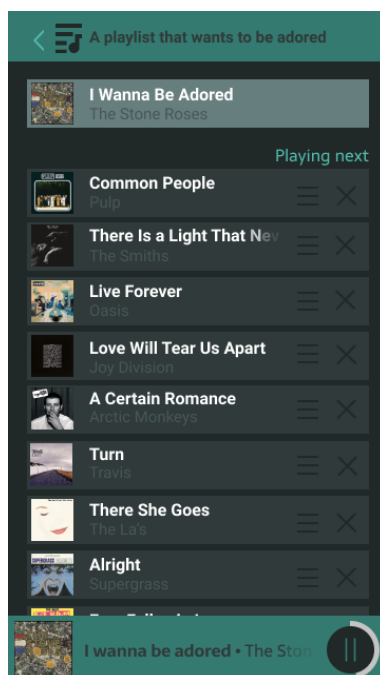
Εικόνα 3.8: Αλλαγή σειράς τραγουδιών στην playlist “A playlist that wants to be adored”

3.8 Σελίδα επεξεργασίας Music Player



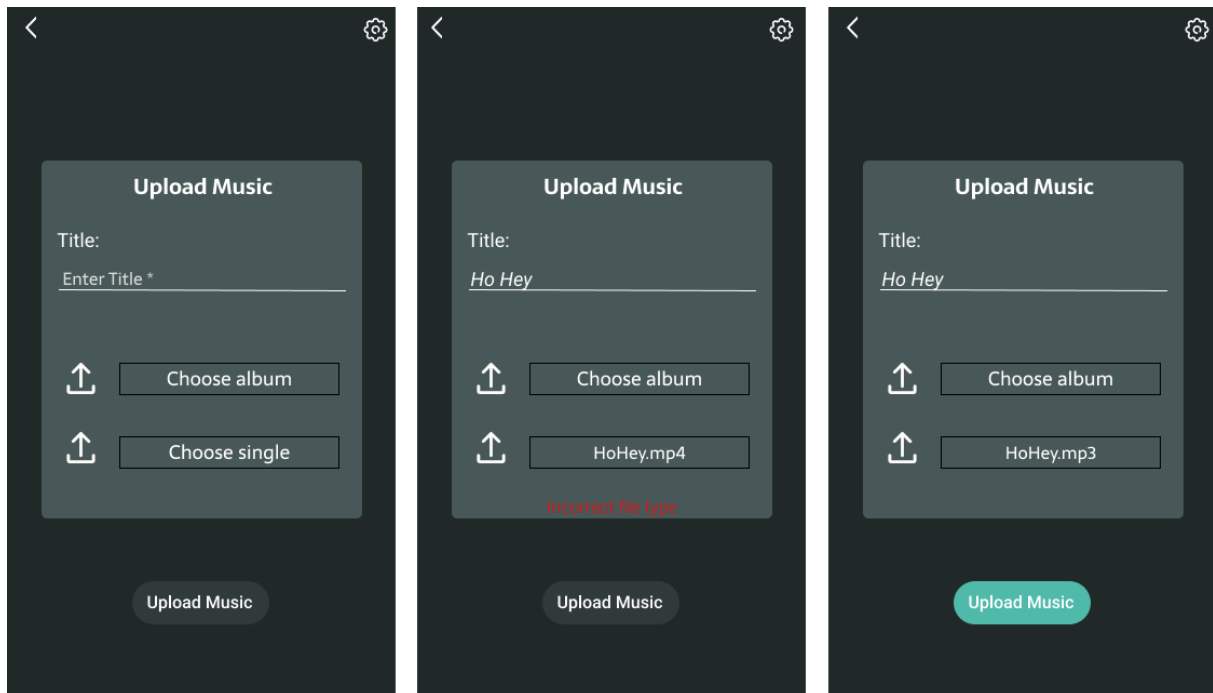
Εικόνα 3.9: Διαχείριση του Music Player, δηλαδή πρόσβαση στις επιλογές παύση/συνέχεια αναπαραγωγής, επόμενο/προηγούμενο τραγούδι τυχαία ή μη αναπαραγωγή, επανάληψη τραγουδιού, διαχείριση του slider και μετάβαση στην ουρά αναπαραγωγής.

3.9 Σελίδα προβολής Play Queue



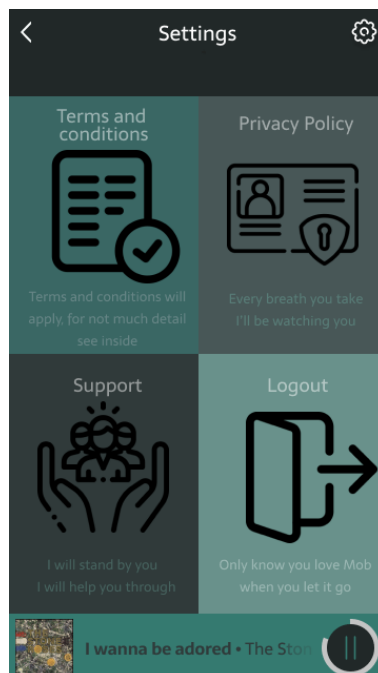
Εικόνα 3.10: Προβολής της σειράς αναπαραγωγής των τραγουδιών με δυνατότητα αναδιάταξης αφαιρέσεις τους από την ουρά

3.10 Σελίδα δημοσίευσης/μεταφόρτωσης μουσικής



Εικόνα 3.11: Διαδικασία μεταφόρτωσης άλμπουμ ή single από τον Καλλιτέχνη, η περίπτωση λάθους τύπου αρχείου και η σωστού

3.11 Σελίδα τροποποίησης των ρυθμίσεων



Εικόνα 3.12: Πρόσβαση στους όρους και προϋποθέσεις, στην πολιτική απορρήτου, στην υποστήριξη και στη δυνατότητα αποσύνδεσης

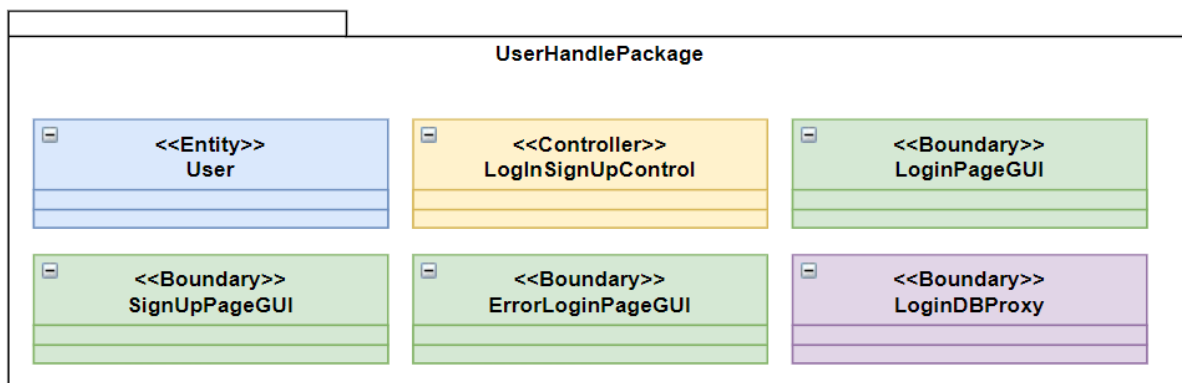
4 Στατική Μοντελοποίηση

Σημείωση: Στα πακέτα που ακολουθούν όλοι οι Constructors έχουν μπει με κενό όρισμα χάριν συντομίας.

4.1 Πακέτα υψηλής προτεραιότητας

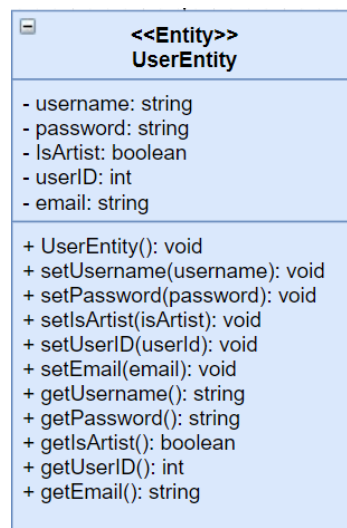
4.1.1 UserHandle Package

Σκοπός αυτού του πακέτου είναι να διαχειριστεί τις εγγραφές, τη σύνδεση και την αποσύνδεση του χρήστη. Περιλαμβάνει τις οθόνες σύνδεσης και εγγραφής, καθώς και την οθόνη που βλέπει ο χρήστης σε περίπτωση σφάλματος. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Entity Classes

- **UserEntity**



Αυτή είναι κλάση οντότητας του χρήστη και περιέχει στοιχεία απαραίτητα για την εγγραφή του και την μετέπειτα ταυτοποίησή του (κατά το Login). Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- **username:** Το όνομα που επιλέγει ο χρήστης κατά τη διάρκεια του SignUp. Πρέπει να είναι μοναδικό.
- **password:** Ο κωδικός πρόσβασης που επιλέγει ο χρήστης κατά τη διάρκεια του SignUp και είναι απαραίτητος για την σύνδεσή του.
- **isArtist:** Μεταβλητή που κάνει τον διαχωρισμό ανάμεσα σε Listener / Artist για δεδομένο χρήστη. Έχει τιμή true αν ο χρήστης εγγράφεται ως Artist, false αν εγγράφεται ως Listener.



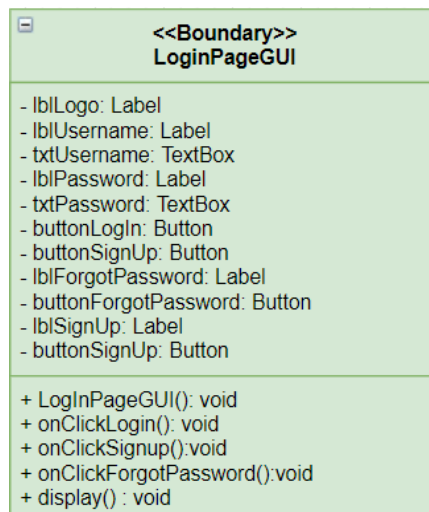
- userID: Ένας μοναδικός και χαρακτηριστικός αριθμός για κάθε χρήστη
- email: Η διεύθυνση e-mail με την οποία εγγράφεται ο χρήστης

Μέθοδοι της κλάσης

- UserEntity(): Constructor του UserEntity
- setUsername(username): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο όνομα χρήστη.
- setPassword(password): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο password του χρήστη.
- setIsArtist(isArtist): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στη μεταβλητή που καθορίζει το εάν ο χρήστης είναι Listener ή Artist.
- setUserID(userId): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο user ID του χρήστη.
- setEmail(email): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στην ηλεκτρονική διεύθυνση του χρήστη.
- getUsername(): Η συνάρτηση αυτή επιστρέφει το όνομα χρήστη.
- getPassword(): Η συνάρτηση αυτή επιστρέφει το password του χρήστη.
- getIsArtist(): Η συνάρτηση αυτή επιστρέφει true εάν ο χρήστης είναι Artist, false εάν είναι Listener.
- getUserID(): Η συνάρτηση αυτή επιστρέφει το user ID του χρήστη.
- getEmail(): Η συνάρτηση αυτή επιστρέφει τη διεύθυνση email του χρήστη.

Boundary Classes

- **LoginPageGUI**



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας σύνδεσης του χρήστη στην εφαρμογή. Υπάρχει ανακατεύθυνση στο SignUp page, σε περίπτωση που ο χρήστης δεν έχει εγγραφεί ακόμα.

Χαρακτηριστικά της κλάσης

- lblLogo: Το Logo του Mob
- lblUsername: Ένδειξη πάνω στο πεδίο συμπλήρωσης username
- txtUsername: Πεδίο συμπλήρωσης κειμένου username
- lblPassword: Ένδειξη πάνω στο πεδίο συμπλήρωσης password

- txtPassword: Πεδίο συμπλήρωσης κειμένου για το password
- buttonLogin: Κουμπί για σύνδεση στην εφαρμογή, σύμφωνα με τα στοιχεία που συμπληρώθηκαν προηγουμένως.
- lblForgotPassword: Κείμενο που κατευθύνει τον χρήστη να πατήσει το αντίστοιχο button σε περίπτωση που ξέχασε τον κωδικό του.
- buttonForgotPassword: Κουμπί που οδηγεί τον χρήστη σε μια διαδικασία υπενθύμισης του password.
- lblSignUp: Κείμενο που κατευθύνει τον χρήστη να πατήσει το αντίστοιχο button σε περίπτωση που δεν έχει εγγραφεί στο Mob.
- buttonSignUp: Κουμπί που ανακατευθύνει τον χρήστη στη σελίδα εγγραφής του Mob

Μέθοδοι της κλάσης

- LoginPageGUI(): Constructor της διεπαφής LoginPageGUI
- onClickLogin(): Καλεί τη συνάρτηση Login() του ελεγκτή, η οποία θα ελέγξει τα στοιχεία εισόδου και θα ανακατευθύνει τον χρήστη.
- onClickSignup(): Καλεί τη συνάρτηση displaySignUpPage() του ελεγκτή για να εμφανιστεί η σελίδα του Sign Up.
- onClickForgotPassword(): Καλεί τη συνάρτηση remindUserPassword() του ελεγκτή, δίνοντας ως όρισμα το email του χρήστη, ώστε να του υπενθυμίσει το password.
- display(): Αυτή η συνάρτηση καλείται από την displayLoginPage() του ελεγκτή και είναι υπεύθυνη για τον σχεδιασμό της διεπαφής σύνδεσης του χρήστη.

• SignUpPageGUI

<<Boundary>> SignUpPageGUI	
- lblLogo: Label	
- lblUsername: Label	
- txtUsername: TextBox	
- lblPassword: Label	
- txtPassword: TextBox	
- lblRepeatPassword: Label	
- txtRepeatPassword: TextBox	
- lblEmail: Label	
- txtEmail: TextBox	
- lblLogin: Label	
- buttonLogin: Button	
- lblSignUpAs: Label	
- buttonSignUpAsListener: Button	
- buttonSignUpAsArtist: Button	
+ SignUpPageGUI(): void	
+ onClickLogin(): void	
+ onClickSignUpAsArtist(): void	
+ onClickSignUpAsListener(): void	
+ display(): void	

Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας εγγραφής του χρήστη στην εφαρμογή. Υπάρχει ανακατεύθυνση στο Login page, σε περίπτωση που ο χρήστης έχει ήδη εγγραφεί στο Mob.

Χαρακτηριστικά της κλάσης

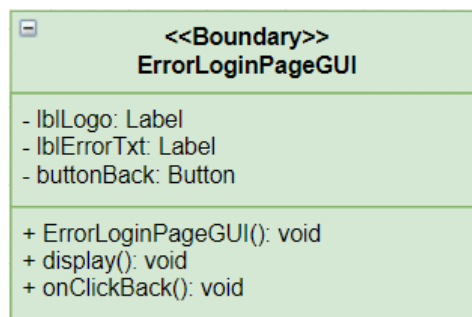
- lblLogo: Το Logo του Mob
- lblUsername: Ένδειξη πάνω στο πεδίο συμπλήρωσης username
- txtUsername: Πεδίο συμπλήρωσης κειμένου username
- lblPassword: Ένδειξη πάνω στο πεδίο συμπλήρωσης password

- txtPassword: Πεδίο συμπλήρωσης κειμένου για το password
- lblRepeatPassword: Ένδειξη πάνω στο πεδίο επανάληψης password
- txtRepeatPassword: Πεδίο συμπλήρωσης κειμένου για επανάληψη του password
- lblEmail: Ένδειξη πάνω στο πεδίο συμπλήρωσης email
- txtEmail: Πεδίο συμπλήρωσης κειμένου για εισαγωγή της ηλεκτρονικής διεύθυνσης του χρήστη
- lblLogIn: Κείμενο που κατευθύνει τον χρήστη να πατήσει το αντίστοιχο button σε περίπτωση που έχει ήδη εγγραφεί στο Mob.
- buttonLogIn: Κουμπί που ανακατευθύνει τον χρήστη στη σελίδα σύνδεσης του Mob.
- lblSignUpAs: Κείμενο που προτρέπει τον χρήστη να εγγραφεί ως Artist ή Listener.
- buttonSignUpAsListener: Κουμπί για εγγραφή στην εφαρμογή ως Listener, σύμφωνα με τα στοιχεία που συμπληρώθηκαν προηγουμένως.
- buttonSignUpAsArtist: Κουμπί για εγγραφή στην εφαρμογή ως Artist, σύμφωνα με τα στοιχεία που συμπληρώθηκαν προηγουμένως.

Μέθοδοι της κλάσης

- SignUpPageGUI(): Constructor της διεπαφής SignUpPageGUI
- onClickSignUpAsArtist(): Καλεί τη συνάρτηση SignUp() του ελεγκτή, θέτοντας την isArtist true.
- onClickSignUpAsListener(): Καλεί τη συνάρτηση SignUp() του ελεγκτή θέτοντας την isArtist false.
- onClickLogin(): Καλεί τη συνάρτηση displayLoginPage() του ελεγκτή για να εμφανιστεί η σελίδα του Login.
- display(): Αυτή η συνάρτηση καλείται από την displaySignUpPage() του ελεγκτή και είναι υπεύθυνη για τον σχεδιασμό της διεπαφής εγγραφής του χρήστη στο Mob.

• ErrorLoginPageGUI



Η κλάση αυτή υλοποιεί τη διεπαφή που πληροφορεί το χρήστη σε περίπτωση σφάλματος στη σύνδεση.

Χαρακτηριστικά της κλάσης

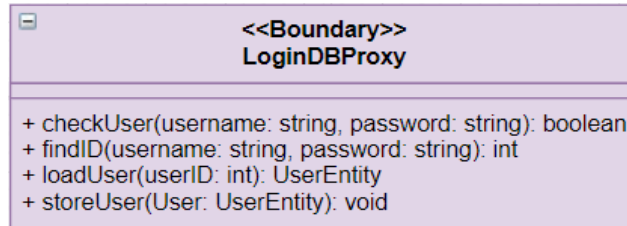
- lblLogo: Το Logo του Mob
- lblErrorTxt: Μήνυμα προς τον χρήστη για σφάλμα κατά τη σύνδεσή του, τον προτρέπει να γυρίσει πίσω και να προσπαθήσει να ξανασυνδεθεί
- buttonBack: Κουμπί που γυρνάει τον χρήστη πίσω στη σελίδα σύνδεσης

Μέθοδοι της κλάσης

- ErrorLoginPageGUI(): Constructor της διεπαφής ErrorLoginPageGUI()

- `display()`: Αυτή η συνάρτηση καλείται από την `displayErrorLoginPage()` του ελεγκτή και είναι υπεύθυνη για τον σχεδιασμό της διεπαφής λάθους κατά την σύνδεση του χρήστη στο Mob.
- `onClickBack()`: Καλεί τη συνάρτηση `back()` του ελεγκτή, που επιστρέφει τον χρήστη στην οθόνη σύνδεσης στο Mob.

• LoginDBProxy



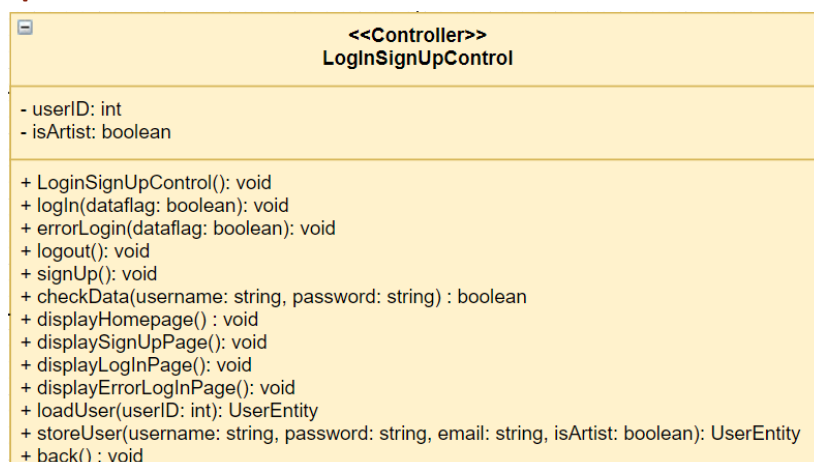
Η κλάση αυτή είναι υπεύθυνη για την αποθήκευση των δεδομένων χρήστη κατά την εγγραφή του στο Mob, αλλά και για την ταυτοποίηση των χρηστών κατά τη διαδικασία του Login.

Μέθοδοι της κλάσης

- `checkUser(username, password)`: Συνάρτηση που ελέγχει αν ο συνδυασμός username - password είναι σωστός.
- `findID(username, password)`: Με βάση τον συνδυασμό username - password, αναζητείται ο αντίστοιχος χρήστης και επιστρέφεται το μοναδικό ID του.
- `loadUser(userID)`: Με τη βοήθεια αυτής της μεθόδου, φορτώνεται ένα User Entity με όλα τα χαρακτηριστικά του, με βάση ένα ID που παρέχεται από το controller.
- `storeUser(User)`: Κατά τη διαδικασία του SignUp, ο controller δημιουργεί ένα UserEntity, το οποίο αποθηκεύεται στη βάση μέσω του DBProxy, ώστε να χρησιμοποιηθεί σε μελλοντικό Login του χρήστη.

Control Classes

• LoginSignUpControl



Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για τον έλεγχο της εισόδου χρήστη και την εγγραφή νέων χρηστών στην εφαρμογή.

Ελέγχει επίσης την ανακατεύθυνση και προβολή των κατάλληλων διεπαφών στον χρήστη.

Χαρακτηριστικά της κλάσης

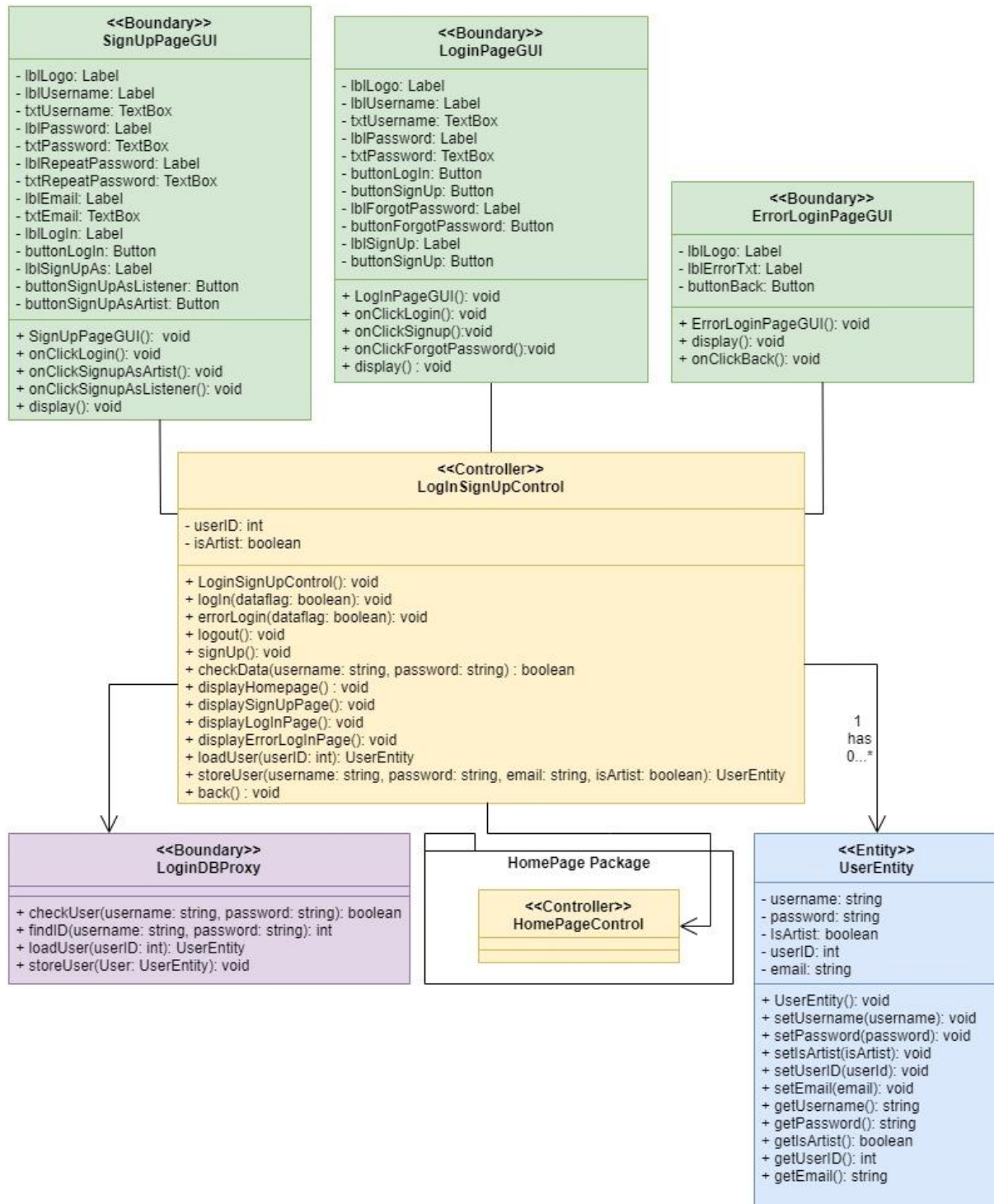
- **userID**: Ένας μοναδικός και χαρακτηριστικός αριθμός για κάθε χρήστη
- **isArtist**: Μεταβλητή που κάνει τον διαχωρισμό ανάμεσα σε Listener / Artist για δεδομένο χρήστη. Έχει τιμή true αν ο χρήστης εγγράφεται ως Artist, false αν εγγράφεται ως Listener.

Μέθοδοι της κλάσης

- **LoginSignUpControl()**: Constructor του ελεγκτή LoginSignUpControl
- **login(dataflag)**: Η μέθοδος φροντίζει να συνδέσει τον χρήστη στο σύστημα εάν η μεταβλητή dataflag έχει τιμή true. Καλεί στη συνέχεια τη συνάρτηση displayHomePage() που κατευθύνει τον χρήστη στην αρχική του σελίδα.
- **errorLogin(dataflag)**: Η μέθοδος φροντίζει να πληροφορήσει τον χρήστη πως συνέβη σφάλμα στη σύνδεση, εάν η μεταβλητή dataflag έχει τιμή false. Καλεί στη συνέχεια τη συνάρτηση displayErrorLoginPage().
- **logout()**: Η μέθοδος αυτή αποσυνδέει τον χρήστη από το σύστημα.
- **signUp()**: Η μέθοδος αυτή φροντίζει ώστε να δημιουργηθεί ένας νέος χρήστης στο σύστημα, καλώντας την storeUser().
- **checkData(username, password)**: Η μέθοδος αυτή ελέγχει εάν ο συνδυασμός username - password που πληκτρολογήθηκε από το χρήστη είναι σωστός. Εάν ναι, καλείται η login(), αν όχι καλείται η errorLogin().
- **displayHomePage()**: Αυτή η συνάρτηση καλεί την display() συνάρτηση του HomePage Controller, εμφανίζοντας την αρχική του χρήστη που συνδέθηκε επιτυχημένα.
- **displaySignUpPage()**: Αυτή η συνάρτηση καλεί την display() του SignUpPageGUI.
- **displayLoginPage()**: Αυτή η συνάρτηση καλεί τη display() του LoginPageGUI.
- **displayErrorLoginPage()**: Αυτή η συνάρτηση καλεί την display() του ErrorLoginPageGUI.
- **remindUserPassword(email)**: Αυτή η συνάρτηση αναλαμβάνει να στείλει στον χρήστη που ξέχασε το password του οδηγίες για την ανάκτησή του.
- **loadUser(userID)**: “Προσφέροντας” ένα userID, αυτή η συνάρτηση, καλώντας την αντίστοιχη συνάρτηση του proxy, μας επιστρέφει ένα UserEntity με όλα τα στοιχεία του συγκεκριμένου χρήστη.
- **storeUser(username, password, email, isArtist)**: Η μέθοδος αυτή αποθηκεύει έναν νέο χρήστη στη βάση, με τα στοιχεία που εισήχθησαν στο SignUpPageGUI, «μιλώντας» με το LoginDBProxy.
- **back()**: Αυτή η συνάρτηση που καλείται μέσα από το ErrorLoginPageGUI, επιτρέπει στον χρήστη να επιστρέψει στη σελίδα του Log In ώστε να εισάγει ξανά τα στοιχεία του.

Διάγραμμα κλάσεων

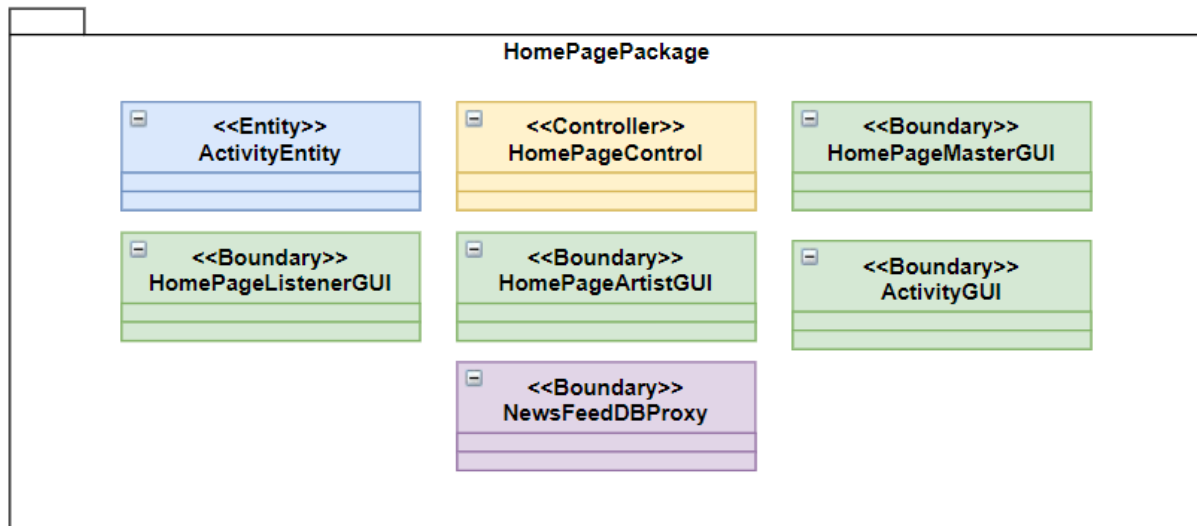
Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.1: Διάγραμμα Κλάσεων Πακέτου UserHandle

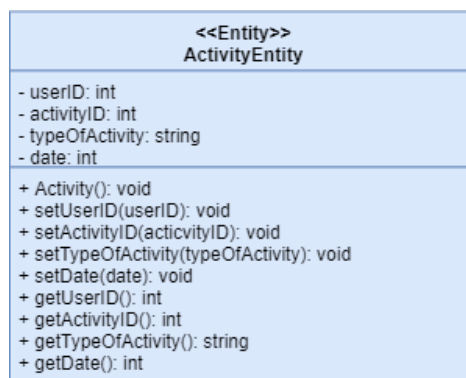
4.1.2 HomePage Package

Το συγκεκριμένο πακέτο μοντελοποιεί την αρχική σελίδα και την λειτουργικότητα της. Όταν ο χρήστης είναι Ακροατής(Listener) στην αρχική σελίδα βλέπει στο News Feed την πρόσφατη δραστηριότητα όσων ακολουθεί. Στην περίπτωση που ο χρήστης είναι Καλλιτέχνης(Artist) στην αρχική σελίδα έχει τις επιλογές να ανεβάσει τη μουσική του και να δει τα στατιστικά του κοινού του. Τέλος, και οι δύο έχουν πρόσβαση στην μπάρα αναζήτησης που θα αναλυθεί σε επόμενο πακέτο.



Entity Classes

• ActivityEntity



Αυτή είναι η κλάση οντότητας της δραστηριότητας των χρηστών που εμφανίζεται στο News Feed και περιέχει χαρακτηριστικά όπως το είδος της κι η ημερομηνία της. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- `userID`: Ο κωδικός του χρήστη που εκτελεί την δραστηριότητα
- `activityID`: Ο κωδικός της δραστηριότητας
- `typeOfActivity`: Ο τύπος της δραστηριότητας, όπως το ανέβασμα άλμπουμ/τραγουδιού από έναν Καλλιτέχνη ή η ενημέρωση Playlist από κάποιον χρήστη
- `date`: Η ημερομηνία της δραστηριότητας

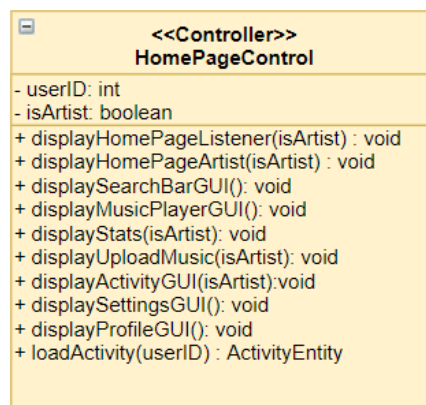
Μέθοδοι της κλάσης

- `Activity()`: Constructor του `ActivityEntity`

- `setUserID(userID)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `userID` του χρήστη.
- `setActivityID(activityID)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `activityID` της δραστηριότητας.
- `setTypeOfActivity(typeOfActivity)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `typeOfActivity` της δραστηριότητας.
- `setDate(date)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `date` της δραστηριότητας.
- `getUserID()`: Η συνάρτηση αυτή επιστρέφει το `userID` του χρήστη.
- `getActivityID()`: Η συνάρτηση αυτή επιστρέφει το `activityID` της δραστηριότητας.
- `getTypeOfActivity()`: Η συνάρτηση αυτή επιστρέφει τον τύπο της δραστηριότητας.
- `getDate()`: Η συνάρτηση αυτή επιστρέφει το `date` της δραστηριότητας.

Control Classes

• HomepageControl



Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για τον έλεγχο της αρχικής οθόνης και εμφανίζει τις κατάλληλες οθόνες ανάλογα με τον αν ο χρήστης είναι Ακροατής ή Καλλιτέχνης. Επίσης είναι υπεύθυνη για την σύνδεση και την ανακατεύθυνση των κατάλληλων διεπαφών στον χρήστη.

Χαρακτηριστικά της κλάσης

- `userID`: ο κωδικός του χρήστη που εκτελεί την δραστηριότητα
- `isArtist`: Μεταβλητή που κάνει τον διαχωρισμό ανάμεσα σε Listener / Artist για δεδομένο χρήστη.

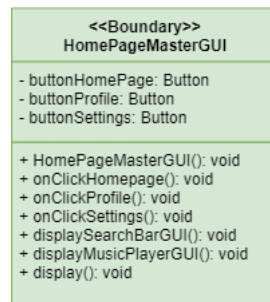
Μέθοδοι της κλάσης

- `displayHomePageListener(isArtist)`: Αυτή η συνάρτηση καλεί την `display()` του `HomePageListenerGUI`, αν η μεταβλητή `isArtist` είναι `false`.
- `displayHomePageArtist(isArtist)`: Αυτή η συνάρτηση καλεί την `display()` του `HomePageArtistGUI`, αν η μεταβλητή `isArtist` είναι `true`.
- `displaySearchBarGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `SearchBarGUI`.
- `displayMusicPlayerGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `MusicPlayerGUI`.
- `displayStats(isArtist)`: Αυτή η συνάρτηση καλεί την `display()` του `SignUpPageGUI`, αν η μεταβλητή `isArtist` είναι `true` και ο Καλλιτέχνης έχει πατήσει το `buttonViewStats`.

- `displayUploadMusic(isArtist)`: Αυτή η συνάρτηση καλεί την `display()` του `UploadMusicGUI`, αν η μεταβλητή `isArtist` είναι `true` και ο Καλλιτέχνης έχει πατήσει το `buttonUploadMusic`.
- `displayActivityGUI(isArtist)`: Αυτή η συνάρτηση καλεί την `display()` του `ActivityGUI`, αν η μεταβλητή `isArtist` είναι `false`.
- `displaySettingsGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `SettingsGUI`, όταν ο χρήστης πατήσει το `buttonSettings`.
- `displayProfileGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `ProfileGUI`, όταν ο χρήστης πατήσει το `buttonProfile`.
- `loadActivity(userID)`: Καλείται και εμφανίζεται μια λίστα από `activities` χρηστών που ακολουθεί ο χρήστης για να μπουν στο `newsfeed` του. Παίρνει ως όρισμα το ID του χρήστη στον οποίο θέλουμε να προβάλουμε τα `activities`.

Boundary Classes

• HomePageMasterGUI



Η κλάση αυτή υλοποιεί τη διεπαφή του μέρους της αρχικής οθόνης που είναι κοινό και στις δύο κατηγορίες χρηστών.

Χαρακτηριστικά της κλάσης

- `buttonHomePage`: Κουμπί για μετάβαση στην Αρχική Σελίδα
- `buttonProfile`: Κουμπί για μετάβαση στο προφίλ του χρήστη
- `buttonSettings`: Κουμπί για μετάβαση στις ρυθμίσεις

Μέθοδοι της κλάσης

- `HomePageMasterGUI()`: Constructor της διεπαφής `HomePageMasterGUI`
- `onClickHomepage()`: Καλεί την συνάρτηση `display()` του `HomePageMasterGUI`.
- `onClickProfile()`: Καλεί την συνάρτηση `displayProfileGUI()` του ελεγκτή.
- `onClickSettings()`: Καλεί την συνάρτηση `displaySettingsGUI()` του ελεγκτή.
- `displaySearchBarGUI()`: Καλεί την συνάρτηση `displaySearchBarGUI()` του ελεγκτή.
- `displayMusicPlayerGUI()`: Καλεί την συνάρτηση `displayMusicPlayerGUI()` του ελεγκτή.
- `display()`: Η συνάρτηση που καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● HomePageListenerGUI

<<Boundary>> HomePageListenerGUI	
- lblNewsfeed: Label	
+ HomePageListenerGUI(): void	
+ displayActivityGUI(): void	
+ display(): void	

Η κλάση αυτή κληρονομεί την HomePageMasterGUI κι επιπλέον υλοποιεί τη διεπαφή που είναι προσβάσιμη μόνο από τον Ακροατή, δηλαδή το News Feed.

Χαρακτηριστικά της κλάσης

- lblNewsfeed: Ένδειξη “News Feed” στην αρχική σελίδα του Ακροατή.

Μέθοδοι της κλάσης

- HomePageListenerGUI(): Constructor της διεπαφής HomePageListenerGUI
- displayActivityGUI(): Καλεί την συνάρτηση displayActivityGUI() του ελεγκτή.
- display(): Η συνάρτηση που καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● HomePageArtistGUI

<<Boundary>> HomePageArtistGUI	
- buttonViewStats: Button	
- buttonUploadMusic: Button	
- imgStats: Img	
- imgMusic: Img	
+ HomePageArtistGUI(): void	
+ onClickViewStats(): void	
+ onClickUploadMusic(): void	
+ display(): void	

Η κλάση αυτή κληρονομεί την HomePageMasterGUI κι επιπλέον υλοποιεί τη διεπαφή που είναι προσβάσιμη μόνο από τον Καλλιτέχνη, δηλαδή τα κουμπιά για την ανακατεύθυνση στη σελίδα των στατιστικών και του ανεβάσματος της μουσικής.

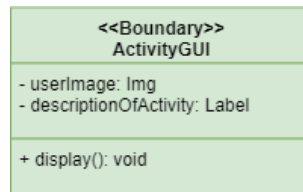
Χαρακτηριστικά της κλάσης

- buttonViewStats: Κουμπί για μετάβαση στη σελίδα των στατιστικών.
- buttonUploadMusic: Κουμπί για μετάβαση στη σελίδα ανεβάσματος μουσικής.
- imgStats: Γραφικό για τα στατιστικά
- imgMusic: Γραφικό για ανέβασμα μουσικής στην εφαρμογή

Μέθοδοι της κλάσης

- HomePageArtistGUI(): Constructor της διεπαφής HomePageArtistGUI
- onClickViewStats(): Καλεί την συνάρτηση displayStats() του ελεγκτή.
- onClickUploadMusic(): Καλεί την συνάρτηση displayUploadMusic() του ελεγκτή.
- display(): Η συνάρτηση που καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● ActivityGUI



Η κλάση αυτή υλοποιεί τη διεπαφή των πληροφοριών της δραστηριότητας ενός χρήστη που ακολουθεί ο Ακροατής σε ένα πλαίσιο, έτσι ώστε να δημιουργηθεί η λίστα των δραστηριοτήτων στο News Feed.

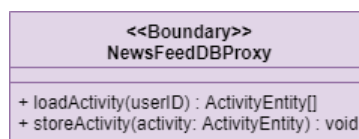
Χαρακτηριστικά της κλάσης

- imageUrl: Η εικόνα του χρήστη που εκτελεί την δραστηριότητα
- descriptionOfActivity: Η περιγραφή της δραστηριότητας του χρήστη, που περιλαμβάνει το όνομα του καθώς και τον τύπο της δραστηριότητας.

Μέθοδοι της κλάσης

- display(): Η συνάρτηση η οποία καλείται κι εμφανίζει στο News Feed την εικόνα προφίλ του χρήστη που ακολουθεί ο χρήστης, μαζί με το όνομα και την περιγραφή της δραστηριότητας του. Στην περίπτωση δραστηριοτήτων από πολλούς χρήστες, εμφανίζει τα παραπάνω σε μορφή λίστας.

● NewsFeedDBProxy



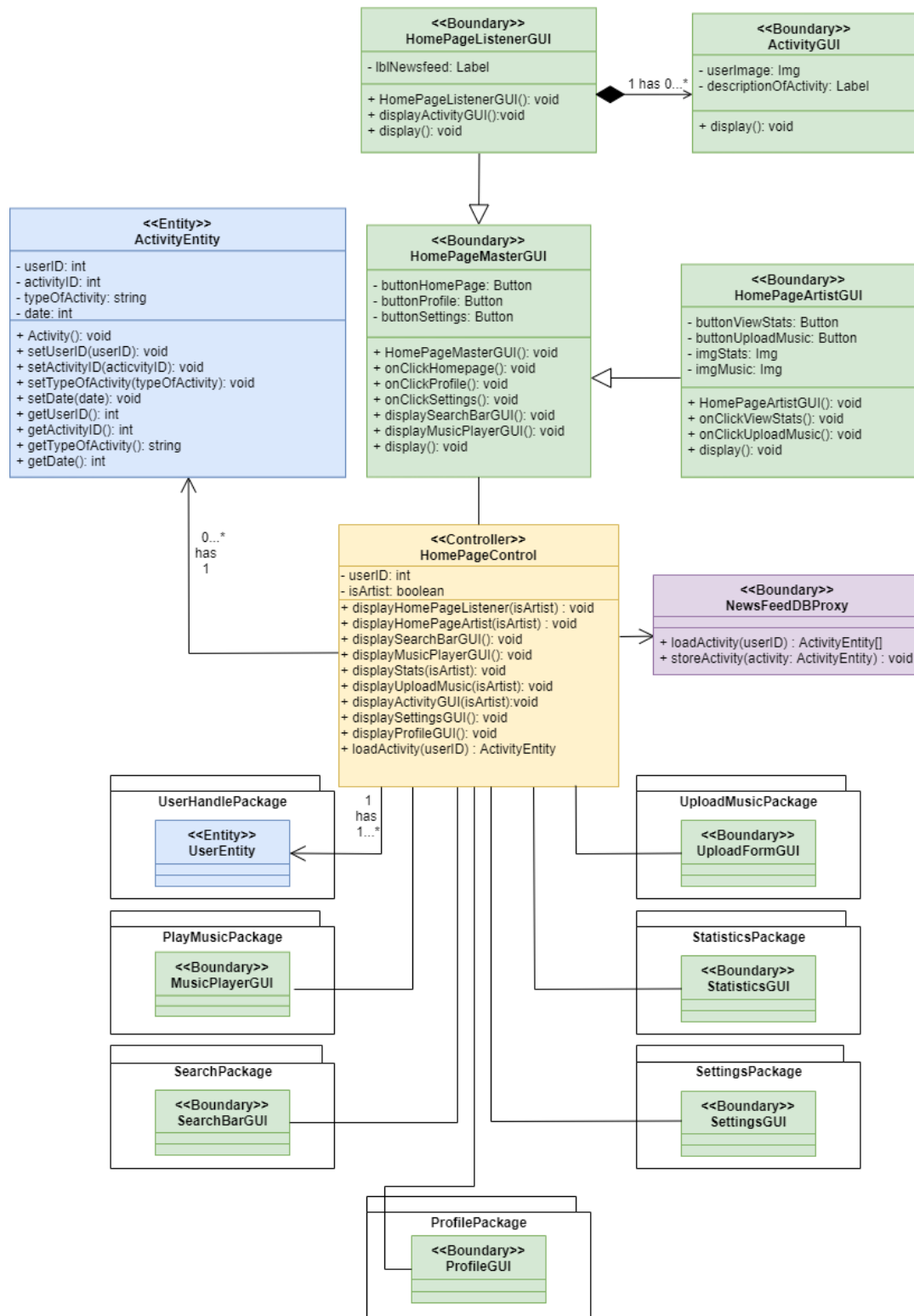
Η κλάση αυτή είναι υπεύθυνη για την αποθήκευση της δραστηριότητας των χρηστών.

Μέθοδοι της κλάσης

- loadActivity(userID): Αυτή η μέθοδος επιστρέφει στον controller μια λίστα από activities χρηστών που ακολουθεί ο χρήστης για να μπουν στο newsfeed του. Παίρνει ως όρισμα το ID του χρήστη στον οποίο θέλουμε να προβάλουμε τα activities.
- storeActivity(activity: ActivityEntity) : Αυτή είναι η συνάρτηση που αποθηκεύει στη βάση όλα τα Activities του κάθε χρήστη.

Διάγραμμα κλάσεων

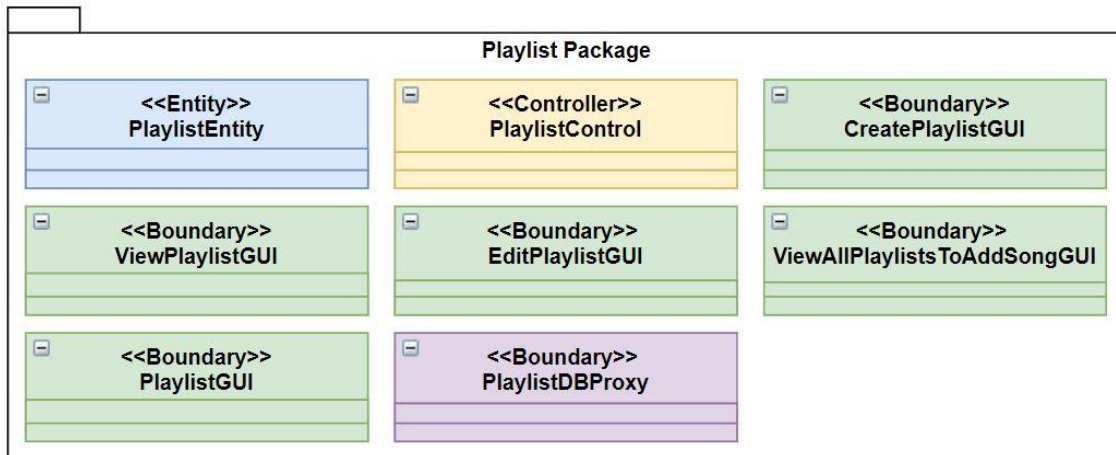
Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.2: Διάγραμμα κλάσεων του Πακέτου HomePage

4.1.3 Playlist Package

Σκοπός αυτού του πακέτου είναι να διαχειριστεί τις Playlists. Περιλαμβάνει την δημιουργία, επεξεργασία και την προβολή μιας Playlist, όπως επίσης και την οθόνη που περιλαμβάνει μια λίστα από τις Playlists του χρήστη κατά την ενέργεια προσθήκης ενός τραγουδιού σε μία από αυτές. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Entity Classes

• PlaylistEntity



Αυτή είναι κλάση οντότητας της Playlist και περιέχει στοιχεία για τον κάτοχο της, το περιεχόμενο και την ιδιωτικότητα της. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- **playlistID:** Ο κωδικός της Playlist, ο οποίος είναι μοναδικός και χαρακτηριστικός της κάθε Playlist
- **title:** Ο τίτλος της Playlist
- **userID:** Ο κωδικός του χρήστη που έχει δημιουργήσει την Playlist

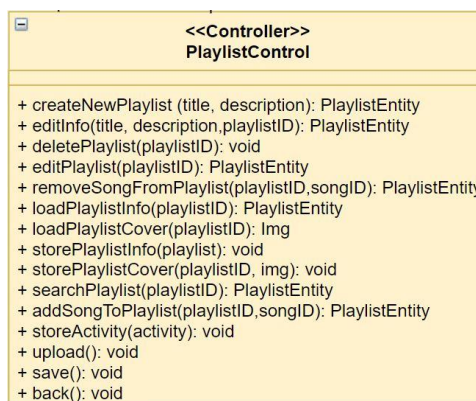
- **description:** Η περιγραφή της Playlist
- **songs:** Πίνακας με το σύνολο των τραγουδιών που έχουν προστεθεί στη Playlist
- **isPublic:** Ένδειξη σχετικά με την ιδιωτικότητα της Playlist, έχει τιμή true εάν η Playlist είναι δημόσια, false εάν είναι ιδιωτική.
- **isEmpty:** Ένδειξη σχετικά με το περιεχόμενο της Playlist, έχει τιμή true εάν η Playlist είναι άδεια, false εάν έχει ένα τουλάχιστον τραγούδι.

Μέθοδοι της κλάσης

- **PlaylistEntity():** Constructor που αρχικοποιεί όλες τις μεταβλητές της κλάσης PlaylistEntity
- **setPlaylistID(playlistID):** Η συνάρτηση αυτή θέτει τιμή στο playlistID της Playlist.
- **setTitle(title):** Η συνάρτηση αυτή θέτει τον τίτλο της Playlist.
- **setUserID(userID):** Η συνάρτηση αυτή θέτει τιμή στον κωδικό χρήστη της Playlist.
- **setDescription(description):** Η συνάρτηση αυτή θέτει την περιγραφή της Playlist.
- **setSongs(songs):** Η συνάρτηση αυτή θέτει τα τραγούδια της Playlist.
- **setPublic(public):** Η συνάρτηση αυτή θέτει ένδειξη δημόσιας ή ιδιωτικής Playlist.
- **setEmpty(empty):** Η συνάρτηση αυτή θέτει τιμή άδειας ή όχι Playlist.
- **getPlaylistID():** Η συνάρτηση αυτή επιστρέφει τον κωδικό της Playlist.
- **getTitle():** Η συνάρτηση αυτή επιστρέφει τον τίτλο της Playlist.
- **getUserID():** Η συνάρτηση αυτή επιστρέφει τον κωδικό του χρήστη της Playlist.
- **getDescription():** Η συνάρτηση αυτή επιστρέφει την περιγραφή της Playlist.
- **getSongs():** Η συνάρτηση αυτή επιστρέφει το σύνολο των τραγουδιών της Playlist.
- **getPublic():** Η συνάρτηση αυτή επιστρέφει αν η Playlist είναι ιδιωτική ή δημόσια.
- **getEmpty():** Η συνάρτηση αυτή επιστρέφει αν η Playlist είναι άδεια ή όχι.

Control Classes

- **PlaylistControl**



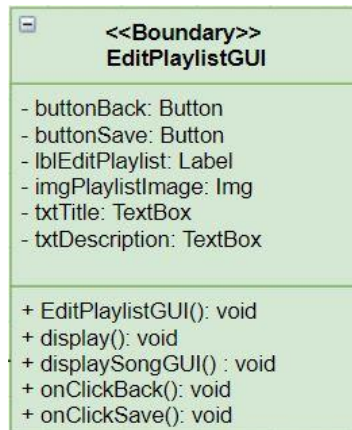
Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για τον έλεγχο όλων των ενεργειών που σχετίζονται με την δημιουργία, επεξεργασία και προβολή μιας Playlist. Ελέγχει επίσης την ανακατεύθυνση και προβολή των κατάλληλων διεπαφών στον χρήστη.

Μέθοδοι της κλάσης

- **createNewPlaylist (title, description):** Η συνάρτηση που καλείται κατά την δημιουργία μιας νέας Playlist και παίρνει σαν ορίσματα τα πεδία που έχει συμπληρώσει ο χρήστης και δημιουργεί την Playlist.

- `editInfo(title, description, playlistID)`: Η συνάρτηση που καλείται όταν ο χρήστης επεξεργάζεται τον τίτλο ή την περιγραφή μιας Playlist και επιστρέφει την ενημερωμένη Playlist.
- `deletePlaylist(playlistID)`: Η συνάρτηση που καλείται ώστε να διαγραφεί μια Playlist και παίρνει σαν όρισμα το `playlistID` της Playlist προς διαγραφή.
- `editPlaylist(playlistID)`: Η συνάρτηση που καλείται όταν ο χρήστης πατήσει για επεξεργασία μιας Playlist και παίρνοντας ως όρισμα το `playlistID` της Playlist επιστρέφει την ενημερωμένη Playlist.
- `removeSongFromPlaylist(playlistID, songID)`: Η συνάρτηση που καλείται όταν ο χρήστης κατά την επεξεργασία της Playlist επιλέξει να διαγράψει ένα τραγούδι από αυτήν. Παίρνει ως όρισμα το `playlistID` της Playlist και το `songID` του τραγουδιού και επιστρέφει την ενημερωμένη Playlist.
- `loadPlaylistInfo(playlistID)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση κλήσης μιας Playlist στο proxy αντικείμενο `PlaylistDBProxy` με σκοπό την επιστροφή μιας Playlist και των πληροφοριών της.
- `loadPlaylistCover(playlistID)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση φόρτωσης της εικόνας μιας Playlist στο proxy αντικείμενο `PlaylistDBProxy` με σκοπό την επιστροφή της.
- `storePlaylistInfo(playlist)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης των πληροφοριών μιας Playlist στο proxy αντικείμενο `PlaylistDBProxy`.
- `storePlaylistCover(playlistID, img)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης της εικόνας μιας Playlist στο proxy αντικείμενο `PlaylistDBProxy`.
- `searchPlaylist(playlistID)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αναζήτησης μιας Playlist στο proxy αντικείμενο `PlaylistDBProxy` με σκοπό της επιστροφής της Playlist που αναζητά ο χρήστης.
- `addSongToPlaylist(playlistID, songID)`: Πρόκειται για την συνάρτηση που καλείται όταν έχει επιλεγεί ένα τραγούδι για προσθήκη σε μια Playlist.
- `storeActivity(activity)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης της δραστηριότητας του χρήστη που αφορά μια Playlist στο proxy αντικείμενο `PlaylistDBProxy`.
- `upload()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει Upload, στην συγκεκριμένη περίπτωση για την εικόνα της Playlist, και περιλαμβάνει την ενημέρωση της και την αποθήκευση στη βάση με την `storePlaylistCover()`.
- `save()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει Save και περιλαμβάνει την ενημέρωση της Playlist μέσω της `editInfo()` και την αποθήκευση των αλλαγών στη βάση μέσω των `storePlaylistInfo()` και `storePlaylistCover()`.
- `back()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει Back από τα `EditPlaylistGUI`, `CreatePlaylistGUI`, `ViewPlaylistGUI` και περιλαμβάνει την επιστροφή στην προηγούμενη σελίδα, δηλαδή την σελίδα προφίλ του.

• EditPlaylistGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας επεξεργασίας της Playlist του χρήστη.

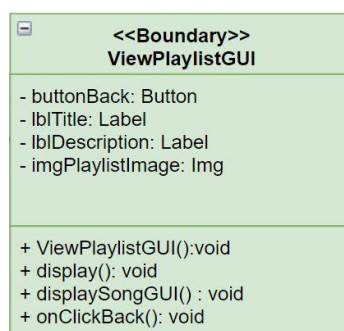
Χαρακτηριστικά της κλάσης

- buttonBack: Κουμπί για επιστροφή στην προηγούμενη σελίδα
- buttonSave: Κουμπί για αποθήκευση των αλλαγών
- lblEditPlaylist: Ονομασία της σελίδας
- imgPlaylistImage: Εικόνα της Playlist
- txtTitle: Πεδίο συμπλήρωσης που αφορά τον τίτλο
- txtDescription: Πεδίο συμπλήρωσης που αφορά την περιγραφή

Μέθοδοι της κλάσης

- EditPlaylistGUI(): Constructor της διεπαφήςEditPlaylistGUI
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- displaySongGUI(): Η συνάρτηση αυτή καλείται ώστε να εμφανιστούν τα τραγούδια που περιέχονται στην Playlist.
- onClickBack(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και καλεί την back() του ελεγκτή.
- onClickSave(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Save και καλεί την save() του ελεγκτή.

• ViewPlaylistGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας στην οποία βρίσκεται ο χρήστης όταν έχει επιλέξει να προβληθεί μια Playlist και να αποκτήσει πρόσβαση στο περιεχόμενο της, δηλαδή τίτλο, περιγραφή και μια λίστα τραγουδιών.

Χαρακτηριστικά της κλάσης

- buttonBack: Κουμπί για επιστροφή στην προηγούμενη σελίδα
- buttonSave: Κουμπί για αποθήκευση των αλλαγών
- lblTitle: Ονομασία της Playlist που εμφανίζεται στη σελίδα
- lblDescription: Η περιγραφή που έχει δοθεί για την Playlist
- imgPlaylistImage: Εικόνα της Playlist

Μέθοδοι της κλάσης

- ViewPlaylistGUI(): Constructor της διεπαφής ViewPlaylistGUI
- display() : Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- displaySongGUI(): Η συνάρτηση αυτή καλείται ώστε να εμφανιστούν τα τραγούδια που περιέχονται στην Playlist.
- onClickBack(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και καλεί την back() του ελεγκτή.

• ViewAllPlaylistsToAddSongGUI

<<Boundary>> ViewAllPlaylistsToAddSongGUI	
-	lblTitle: Label
+	ViewAllPlaylistsToAddSong(): void
+	display(): void
+	displayPlaylistGUI(): void

Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας στην οποία βρίσκεται ο χρήστης όταν έχει επιλέξει ένα τραγούδι να προστεθεί σε μια από τις Playlists του. Περιέχει όλες τις Playlists στις οποίες μπορεί να προσθέσει το τραγούδι.

Χαρακτηριστικά της κλάσης

- lblTitle: Ονομασία της σελίδας

Μέθοδοι της κλάσης

- ViewAllPlaylistsToAddSong(): Constructor της διεπαφής ViewAllPlaylistsToAddSong
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- displayPlaylistGUI: Η συνάρτηση αυτή καλείται ώστε να εμφανιστούν οι Playlist προς επιλογή με τη χρήση του PlaylistGUI.

● PlaylistGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της Playlist.

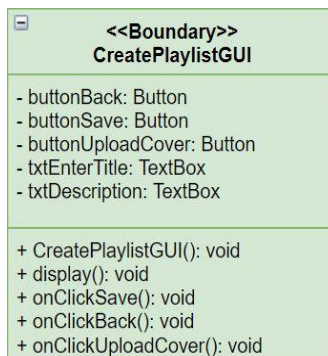
Χαρακτηριστικά της κλάσης

- lblTitle: Ονομασία Playlist που εμφανίζονται στη σελίδα

Μέθοδοι της κλάσης

- PlaylistGUI(): Constructor της διεπαφής PlaylistGUI
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- onClickTitle(): Η συνάρτηση αυτή καλείται όταν ο χρήστης επιλέγει μια Playlist και επιφέρει την εισαγωγή του επιλεγμένου τραγουδιού στη συγκεκριμένη Playlist, καλώντας την addSongToPlaylist() του Controller.

● CreatePlaylistGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας στην οποία ο χρήστης δημιουργεί μια καινούρια Playlist δίνοντας όλα τα απαραίτητα στοιχεία, δηλαδή τίτλο, περιγραφή και cover.

Χαρακτηριστικά της κλάσης

- buttonBack: Κουμπί για την επιστροφή στην προηγούμενη σελίδα
- buttonSave: Κουμπί για την αποθήκευση των αλλαγών
- buttonUploadCover: Κουμπί για το Upload εικόνας της Playlist
- txtEnterTitle: Πεδίο συμπλήρωσης που αφορά τον τίτλο της Playlist
- txtDescription: Πεδίο συμπλήρωσης που αφορά την περιγραφή της Playlist

Μέθοδοι της κλάσης

- CreatePlaylistGUI(): Constructor της διεπαφής CreatePlaylistGUI.
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- onClickBack(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και καλεί την back() του ελεγκτή.

- `onClickSave()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Save και καλεί την `save()` του ελεγκτή.
- `onClickUploadCover()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει πάνω στο σημείο αλλαγής της εικόνας της Playlist και καλεί την συνάρτηση `upload()` του controller.

• PlaylistDBProxy



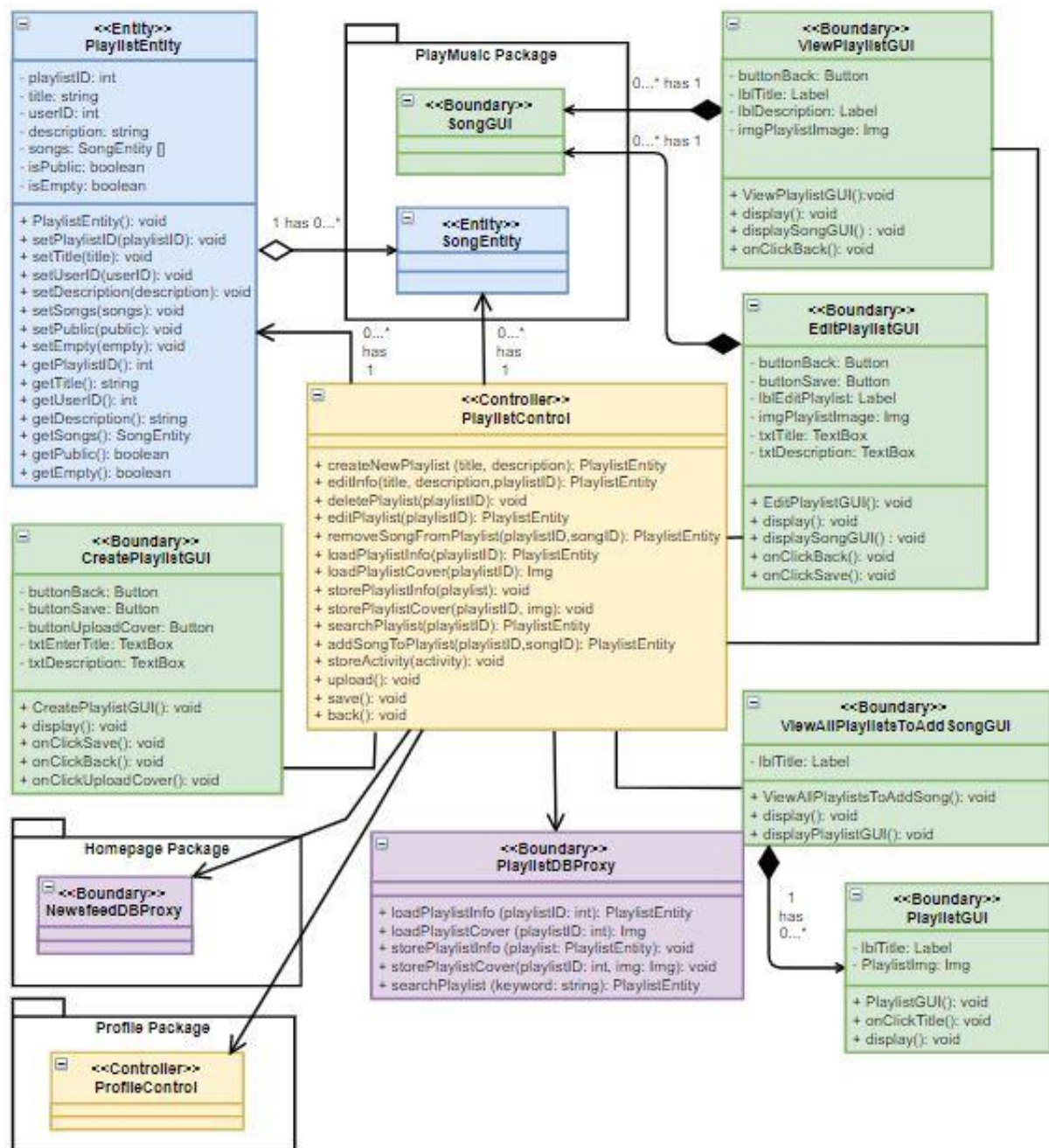
Η κλάση αυτή είναι υπεύθυνη για την αποθήκευση και τη φόρτωση των δεδομένων των Playlist και της δραστηριότητας του χρήστη που σχετίζονται με αυτές.

Μέθοδοι της κλάσης

- `loadPlaylistInfo(playlistID)`: Η μέθοδος αυτή επιστρέφει μια Playlist με τις πληροφορίες της από την βάση, δεδομένου του `playlistID` της.
- `loadPlaylistCover(playlistID)`: Η μέθοδος αυτή επιστρέφει την εικόνα μιας Playlist από την βάση, δεδομένου του `playlistID` της.
- `storePlaylistInfo(playlist)`: Η μέθοδος αυτή αποθηκεύει τις πληροφορίες μιας Playlist στη βάση δεδομένων.
- `storePlaylistCover(playlistID, image)`: Η μέθοδος αυτή αποθηκεύει την εικόνα μιας Playlist στη βάση δεδομένων.
- `searchPlaylist(playlistID)`: Η μέθοδος αυτή αναζητά και επιστρέφει μια Playlist από την βάση, δεδομένης μιας φράσης/λέξης-κλειδί.

Διάγραμμα κλάσεων

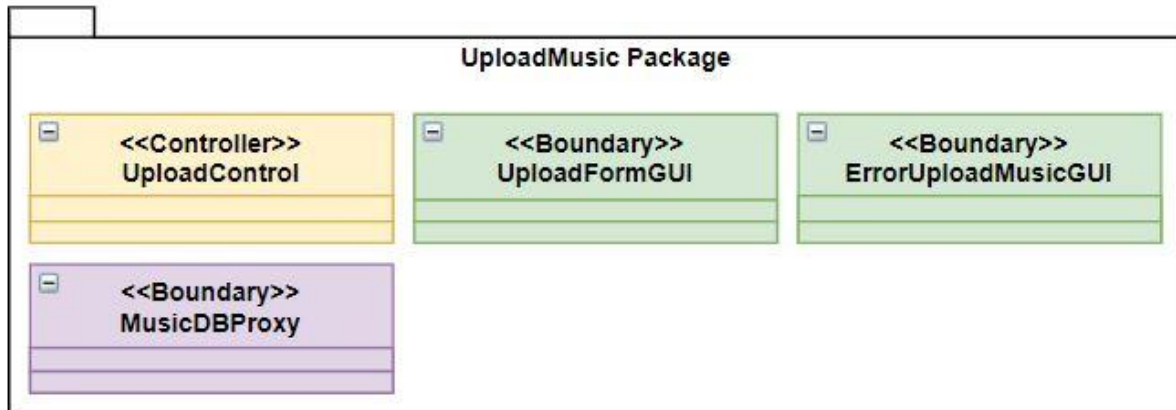
Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.3: Διάγραμμα κλάσεων του Πακέτου Playlist

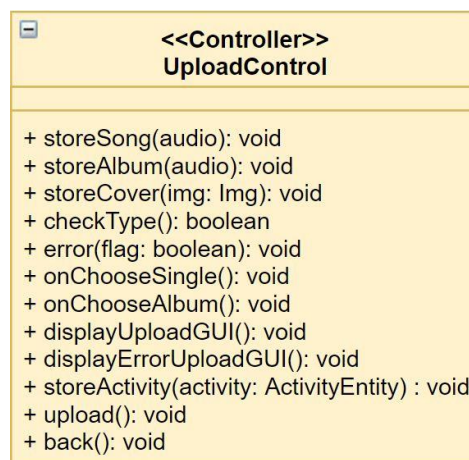
4.1.4 UploadMusic Package

Σκοπός αυτού του πακέτου είναι να διαχειριστεί το «ανέβασμα» αρχείων του Artist, δηλαδή την μεταφόρτωση και δημοσίευση ενός τραγουδιού ή ενός Album. Περιλαμβάνει την οθόνη στην οποία ο Artist μπορεί να ‘ανεβάσει’ αρχεία επιλεγμένα από τον προσωπικό αποθηκευτικό του χώρο, αλλά και την οθόνη σε περίπτωση επιλογής εσφαλμένου τύπου αρχείου. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Control Classes

- **UploadControl**



Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για τον έλεγχο της μεταφόρτωσης και δημοσίευσης νέων τραγουδιών ή Album από Artists. Ελέγχει επίσης την ανακατεύθυνση και προβολή των κατάλληλων διεπαφών στον χρήστη.

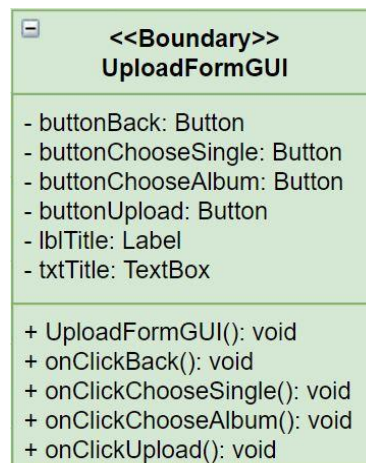
Μέθοδοι της κλάσης

- storeSong(audio): Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης του αρχείου ήχου για ένα τραγούδι στο proxy αντικείμενο MusicDBProxy.
- storeAlbum(audio): Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης του αρχείου ήχου για ένα Album στο proxy αντικείμενο MusicDBProxy.
- storeCover(img): Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης εικόνας για ένα τραγούδι ή Album στο proxy αντικείμενο MusicDBProxy.

- `checkType()`: Η συνάρτηση αυτή καλείται ώστε να επαληθευτεί ο τύπος αρχείου που ανέβασε ο Artist.
- `error(flag)`: Η συνάρτηση αυτή καλείται ώστε να ενημερώσει τον Artist για λάθος τύπο αρχείου.
- `onChooseSingle()`: Η συνάρτηση αυτή καλείται όταν ο Artist επιλέγει το 'ανέβασμα' τραγουδιού από τον προσωπικό αποθηκευτικό του χώρο.
- `onChooseAlbum()`: Η συνάρτηση αυτή καλείται όταν ο Artist επιλέγει το 'ανέβασμα' Album από τον προσωπικό αποθηκευτικό του χώρο.
- `displayUploadGUI()`: Η συνάρτηση αυτή καλείται για την εμφάνιση της σελίδας.
- `displayErrorUploadGUI()`: Η συνάρτηση αυτή καλείται για την εμφάνιση της σελίδας όταν υπάρχει σφάλμα στη μεταφόρτωση.
- `storeActivity(activity)`: Η συνάρτηση αυτή αποθηκεύει στη βάση τα Activities του Artist σχετικά με το «ανέβασμα» τραγουδιών ή Album.
- `upload()`: Η συνάρτηση αυτή καλείται μέσα από το UploadFormGUI, ώστε να γίνει η μεταφόρτωση και δημοσίευση του τραγουδιού/Album που έχει επιλέξει ο Artist με τις συναρτήσεις `storeSong()`, `storeAlbum()`, `storeCover()` αναλόγως την ενέργεια στην οποία έχει προβεί ο Artist.
- `back()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει Back μέσα από το UploadFormGUI και περιλαμβάνει την επιστροφή του στην προηγούμενη σελίδα, δηλαδή το HomePage του.

Boundary Classes

• UploadFormGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας μεταφόρτωσης και δημοσίευσης αρχείων από τον Artist.

Χαρακτηριστικά της κλάσης

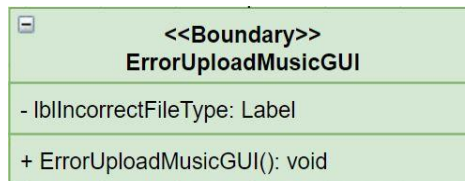
- `buttonBack`: Κουμπί για την επιστροφή στην προηγούμενη σελίδα
- `buttonChooseSingle`: Κουμπί για την επιλογή ενός τραγουδιού από τον προσωπικό αποθηκευτικό χώρο του Artist
- `buttonChooseAlbum`: Κουμπί για την επιλογή ενός Album από τον προσωπικό αποθηκευτικό χώρο του Artist
- `buttonUpload`: Κουμπί για το Upload του επιλεγμένου τραγουδιού/Album.

- lblTitle: Ονομασία πεδίου για τον τίτλο
- txtTitle: Πεδίο συμπλήρωσης που αφορά τον τίτλο του αρχείου προς 'ανέβασμα'

Μέθοδοι της κλάσης

- UploadFormGUI(): Constructor της διεπαφής UploadFormGUI
- onClickBack(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και περιλαμβάνει την κλήση της συνάρτησης back() του controller για επιστροφή στο HomePage.
- onClickChooseSingle(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί ChooseSingle και περιλαμβάνει την κλήση της συνάρτησης onChooseSingle() του controller.
- onClickChooseAlbum(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί ChooseAlbum και περιλαμβάνει την κλήση της συνάρτησης onChooseAlbum() του controller.
- onClickUpload(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί για το Upload του επιλεγμένου τραγουδιού/Album και περιλαμβάνει την κλήση της συνάρτησης upload() του controller.

• ErrorUploadMusicGUI



Η κλάση αυτή υλοποιεί τη διεπαφή που πληροφορεί το χρήστη σε περίπτωση σφάλματος στον τύπο αρχείου. Κληρονομεί την UploadMusicGUI.

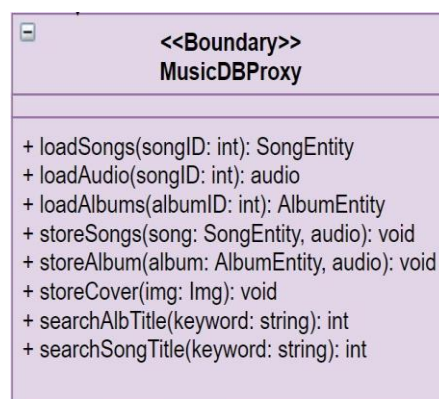
Χαρακτηριστικά της κλάσης

- lblIncorrectFileType: Πρόκειται για μήνυμα ενημέρωσης κατά την μεταφόρτωση εσφαλμένου τύπου αρχείου.

Μέθοδοι της κλάσης

- ErrorUploadMusicGUI(): Constructor της διεπαφής ErrorUploadMusicGUI

• MusicDBProxy



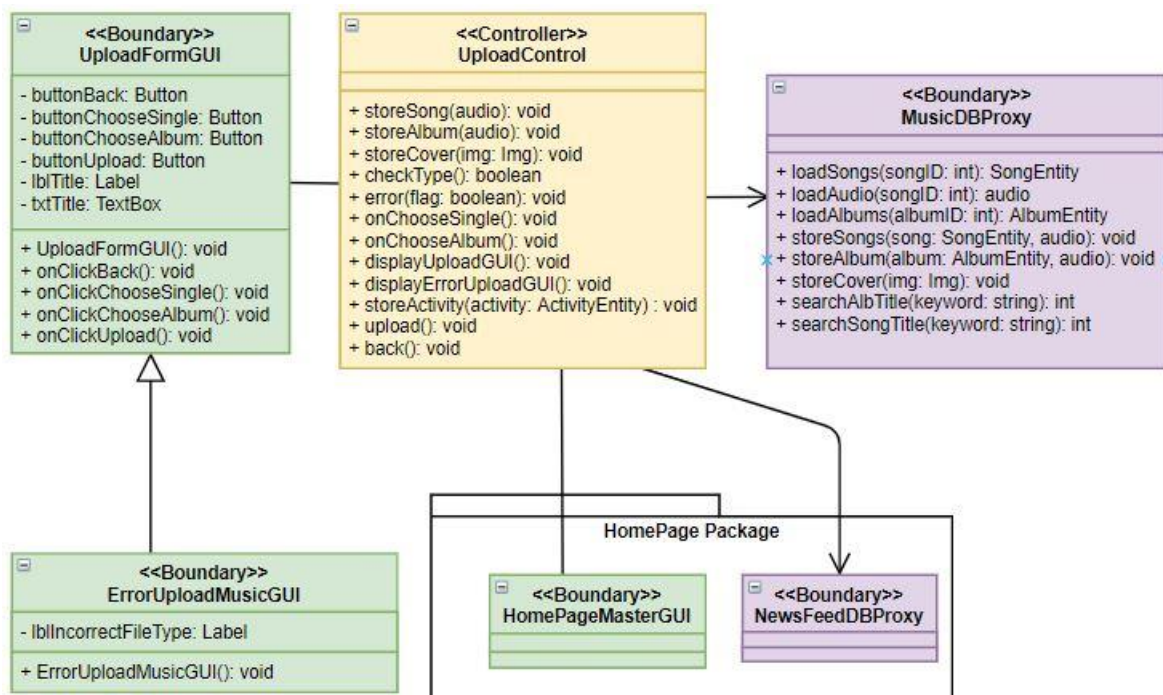
Η κλάση αυτή είναι υπεύθυνη για την αποθήκευση των δεδομένων όλων των τραγουδιών, Album, cover και την αναζήτηση και εύρεση τους.

Μέθοδοι της κλάσης

- loadSongs(songID): Η μέθοδος αυτή επιστρέφει ένα τραγούδι από την βάση, δεδομένου του songID του.
- loadAudio(songID): Η μέθοδος αυτή επιστρέφει ένα αρχείο ήχου από την βάση, δεδομένου του songID του.
- loadAlbums(albumID): Η μέθοδος αυτή επιστρέφει ένα Album από την βάση, δεδομένου του albumID του.
- storeSongs(song, audio): Η μέθοδος που αποθηκεύει τις πληροφορίες ενός τραγουδιού και το αρχείο ήχου που του αντιστοιχεί στη βάση δεδομένων.
- storeAlbum(album, audio): Η μέθοδος που αποθηκεύει τις πληροφορίες ενός Album και το αρχείο ήχου που του αντιστοιχεί στη βάση δεδομένων.
- storeCover(img): Η μέθοδος που αποθηκεύει την εικόνα ενός τραγουδιού ή Album στη βάση δεδομένων.
- searchAlbTitle(keyword): Η μέθοδος αυτή αναζητά ένα Album και επιστρέφει τον albumID του, δεδομένης μιας φράσης/λέξης-κλειδί.
- searchSongTitle(keyword): Η μέθοδος αυτή αναζητά ένα τραγούδι και επιστρέφει τον songID του, δεδομένης μιας φράσης/λέξης-κλειδί.

Διάγραμμα κλάσεων

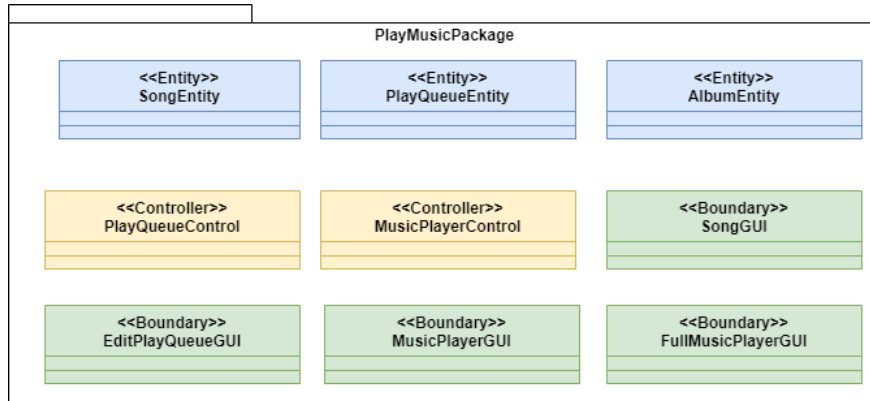
Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.4: Διάγραμμα κλάσεων του Πακέτου UploadMusic

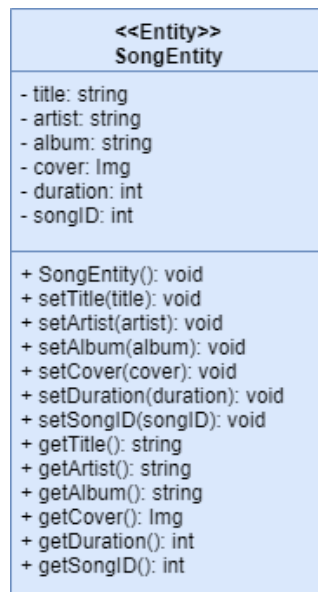
4.1.5 PlayMusic Package

Σκοπός αυτού του πακέτου είναι να διαχειριστεί την αναπαραγωγή μουσικής, δηλαδή την επεξεργασία της PlayQueue. Περιλαμβάνει την διεπαφή στην οποία ο χρήστης μπορεί να πατήσει play για να ξεκινήσει η αναπαραγωγή ενός τραγουδιού. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Entity Classes

- **SongEntity**



Αυτή είναι κλάση οντότητας του τραγουδιού και περιέχει στοιχεία απαραίτητα για την αποθήκευση, εύρεση και αναπαραγωγής του. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- title: Ο τίτλος του τραγουδιού
- artist: Το όνομα του Καλλιτέχνη του τραγουδιού
- album: Ένα άλμπουμ μέσα στο οποίο ανήκει ένα τραγούδι
- cover: Μια εικόνα που αντιστοιχεί σε ένα συγκεκριμένο τραγουδιού
- duration: Η χρονική διάρκεια του τραγουδιού
- songID: Ο κωδικός του τραγουδιού, που είναι μοναδικός και χαρακτηριστικός για κάθε Song

Μέθοδοι της κλάσης

- SongEntity(): Constructor του SongEntity
- setTitle(title): Η συνάρτηση που θέτει τον τίτλο του τραγουδιού.

- `setArtist(artist)`: Η συνάρτηση που θέτει το όρισμα που δέχεται το `Artist` του τραγουδιού.
- `setAlbum(album)`: Η συνάρτηση που θέτει το όρισμα που δέχεται το `Album` του τραγουδιού.
- `setCover(cover)`: Η συνάρτηση που θέτει το όρισμα που δέχεται το `Cover` του τραγουδιού.
- `setDuration(duration)`: Η συνάρτηση που θέτει το όρισμα που δέχεται το `Duration` του τραγουδιού.
- `setSongID(songID)`: Η συνάρτηση που θέτει τιμή στον κωδικό του τραγουδιού.
- `getTitle()`: Η συνάρτηση που επιστρέφει τον τίτλο του τραγουδιού.
- `getArtist()`: Η συνάρτηση που επιστρέφει τον Καλλιτέχνη του τραγουδιού.
- `getAlbum()`: Η συνάρτηση που επιστρέφει το άλμπουμ του τραγουδιού.
- `getCover()`: Η συνάρτηση που επιστρέφει την εικόνα του τραγουδιού.
- `getDuration()`: Η συνάρτηση που επιστρέφει την χρονική διάρκεια του τραγουδιού.
- `getSongID()`: Η συνάρτηση που επιστρέφει τον κωδικό του τραγουδιού.

• PlayQueueEntity

<<Entity>> PlayQueueEntity
- isEmpty: boolean - isShuffle: boolean - isRepeat: boolean - songs: SongEntity[]
+ PlayQueueEntity(): void + setIsEmpty(empty):void + setSongs(songs):void + setIsShuffle(shuffle):void + setIsRepeat(repeat):void + getIsEmpty(): boolean + getSongs(): SongEntity[] + getIsShuffle(): boolean + getIsRepeat(): boolean

Αυτή είναι κλάση οντότητας της `PlayQueue` και περιέχει στοιχεία απαραίτητα για την επεξεργασία της και την αναπαραγωγή της. Η κλάση `PlayQueueEntity` περιγράφει τη σειρά των τραγουδιών στην ουρά αναπαραγωγής. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- `isEmpty`: Ένδειξη για το αν η ουρά αναπαραγωγής είναι κενή ή όχι
- `isShuffle`: Ένδειξη για τη σειρά αναπαραγωγής των τραγουδιών σχετικά με την διάταξή τους
- `isRepeat`: Ένδειξη για τη σειρά αναπαραγωγής των τραγουδιών σχετικά με την επανάληψή τους
- `songs`: Πίνακας με το σύνολο των τραγουδιών που έχουν προστεθεί στη Playlist

Μέθοδοι της κλάσης

- `PlayQueueEntity()`: Constructor του `PlayQueueEntity`
- `setIsEmpty(empty)`: Η συνάρτηση που θέτει τιμή άδειας ή όχι Playlist.
- `setIsShuffle(shuffle)`: Η συνάρτηση που θέτει αν η `PlayQueue` είναι τυχαία ταξινομημένη ή όχι.
- `setIsRepeat(repeat)`: Η συνάρτηση που θέτει αν θα επαναληφθεί ή όχι η `PlayQueue`.
- `setSongs(songs)`: Η συνάρτηση που θέτει τα τραγούδια της Playlist.
- `getIsEmpty()`: Η συνάρτηση που επιστρέφει αν η Playlist είναι άδεια ή όχι.

- `getIsShuffle()`: Η συνάρτηση που επιστρέφει αν η Playlist είναι ταξινομημένη ή όχι.
- `getIsRepeat()`: Η συνάρτηση που επιστρέφει αν η Playlist θα αναπαραχθεί με επανάληψη ή όχι.
- `getSongs()`: Η συνάρτηση που επιστρέφει το σύνολο των τραγουδιών της Playlist.

- **AlbumEntity**

<<Entity>> AlbumEntity
- title: string - artist: string - album: string - cover: Image - albumID: int - songs: SongEntity[] - albums: AlbumEntity[]
+ AlbumEntity(): void + setTitle(title): void + setArtist(artist): void + setAlbum(album): void + setCover(cover): void + setAlbumID(albumID): void + setSong(song): void + getTitle(): string + getArtist(): string + getAlbum(): string + getCover(): Image + getAlbumID(): int + getSong(): SongEntity[]

Αυτή είναι κλάση οντότητας του άλμπουμ και περιέχει στοιχεία απαραίτητα για την αποθήκευση, εύρεση και επεξεργασία του. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- title: ο τίτλος του άλμπουμ.
- artist: το όνομα του Καλλιτέχνη του άλμπουμ.
- cover: μια εικόνα που αντιστοιχεί σε ένα συγκεκριμένο άλμπουμ.
- albumID: ο κωδικός του άλμπουμ, που είναι μοναδικός και χαρακτηριστικός για κάθε άλμπουμ.
- songs: πίνακας με τα τραγούδια ενός συγκεκριμένου άλμπουμ.
- albums: πίνακας με το σύνολο των άλμπουμ που έχουν προστεθεί στο άλμπουμ.

Μέθοδοι της κλάσης

- `AlbumEntity()`: Constructor του AlbumEntity
- `setTitle(title)`: Η συνάρτηση που θέτει τον τίτλο του άλμπουμ.
- `setArtist(artist)`: Η συνάρτηση που θέτει το όρισμα που δέχεται το Artist του άλμπουμ.
- `setCover(cover)`: Η συνάρτηση που θέτει το όρισμα που δέχεται το Cover του άλμπουμ.
- `setAlbumID(albumID)`: Η συνάρτηση που θέτει τιμή στον κωδικό του άλμπουμ.
- `setSongs(song)`: Η συνάρτηση που θέτει τα τραγούδια του άλμπουμ.
- `setAlbums(album)`: Η συνάρτηση που θέτει τα επιμέρους άλμπουμ των άλμπουμ.
- `getTitle()`: Η συνάρτηση που επιστρέφει τον τίτλο του άλμπουμ.
- `getCover()`: Η συνάρτηση που επιστρέφει την εικόνα του άλμπουμ.
- `getAlbumID()`: Η συνάρτηση που επιστρέφει τον κωδικό του άλμπουμ.
- `getSongs()`: Η συνάρτηση που επιστρέφει στον κωδικό του τραγουδιού ενός άλμπουμ.

Control Classes

• PlayQueueControl

<<Controller>> PlayQueueControl
+ changeSongOrder(playQueue): PlayQueueEntity + removeSong(playQueue): void

Αυτή η κλάση υλοποιεί τις απαραίτητες συναρτήσεις για την επεξεργασία μιας PlayQueue.

Μέθοδοι της κλάσης

- **changeSongOrder(playQueue):** Η συνάρτηση καλείται όταν ο χρήστης κατά την επεξεργασία της PlayQueue επιλέξει να αλλάξει την σειρά αναπαραγωγής των τραγουδιών. Παίρνει ως όρισμα την PlayQueue και επιστρέφει την ενημερωμένη.
- **removeSong(playQueue):** Η συνάρτηση καλείται όταν ο χρήστης κατά την επεξεργασία της PlayQueue επιλέξει να διαγράψει ένα τραγούδι από αυτήν. Επιστρέφει την ενημερωμένη PlayQueue.

• MusicPlayerControl

<<Controller>> MusicPlayerControl
- currentSecond: Int
+ play(song): void + pause(song): void + shuffle(playQueue): PlayQueueEntity + repeat(playQueue): PlayQueueEntity + playNext(): void + playPrevious(): void + addSongToPlaylist(songID): void + changeMoment(currentSecond): void + displayFullMusicPlayer(): void + displayMusicPlayerGUI(): void + loadSongs(SongID): SongEntity + loadAudio(SongID): audio + loadAlbums(AlbumID): AlbumEntity[]

Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για την προβολή, αναπαραγωγή και επεξεργασία μιας PlayQueue. Ελέγχει επίσης την ανακατεύθυνση των κατάλληλων διεπαφών στον χρήστη.

Χαρακτηριστικά της κλάσης

- **currentSecond:** συγκεκριμένη χρονική στιγμή, στην οποία θέλει ο χρήστης να ακούσει το τραγούδι.

Μέθοδοι της κλάσης

- **play(song):** Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το buttonPlay και ξεκινάει την αναπαραγωγή του τραγουδιού της PlayQueue.
- **pause(song):** Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το buttonPlay και σταματάει την αναπαραγωγή του τραγουδιού της PlayQueue.
- **shuffle(playQueue):** Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το buttonShuffle και έτσι ταξινομείται τυχαία η PlayQueue.
- **repeat(playQueue):** Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το buttonRepeat και επαναλαμβάνεται η αναπαραγωγή του συγκεκριμένου τραγουδιού.

- `playNext()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το `buttonNext` και αναπαράγεται το επόμενο τραγουδιού.
- `playPrevious()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το `buttonPrevious` και αναπαράγεται το προηγούμενο τραγουδιού.
- `addSongToPlaylist(songID)`: Πρόκειται για την συνάρτηση που καλείται όταν έχει επιλεγεί ένα τραγούδι για προσθήκη σε μια Playlist.
- `changeMoment(currentSecond)`: Πρόκειται για την συνάρτηση που καλείται όταν μετακινηθεί ο slider, παίρνει σαν μεταβλητή ένα συγκεκριμένο δευτερόλεπτο και συνεχίζει από εκεί την αναπαραγωγή του τραγουδιού.
- `displayFullMusicPlayerGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `MusicPlayerGUI`, όταν ο χρήστης πατήσει πάνω στον τίτλο του τραγουδιού.
- `displayMusicPlayerGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `MusicPlayerGUI`.
- `loadSongs(SongID)`: Αυτή η συνάρτηση, επικοινωνώντας με το `MusicDBProxy` επιστρέφει τα `SongEntity`, με έναν συγκεκριμένο κωδικό.
- `loadAudio(SongID)`: Αυτή η συνάρτηση, επικοινωνώντας με το `MusicDBProxy` επιστρέφει ένα αρχείο ήχου από την βάση, δεδομένου του `songID` του.
- `loadAlbums(AlbumID)`: Αυτή η συνάρτηση, επικοινωνώντας με το `ProfileDBProxy` επιστρέφει τα `AlbumEntity`, με έναν συγκεκριμένο κωδικό.

Boundary Classes

• SongGUI

<<Boundary>> SongGUI
- lblTitle: Label - lblArtist: Label - imgSong: Img
+ SongGUI(): void + onClickTitle(): void + display(): void

Η κλάση αυτή υλοποιεί τη διεπαφή εμφάνισης του τραγουδιού στο `MusicPlayerGUI`.

Χαρακτηριστικά της κλάσης

- `lblTitle`: Τίτλος ενός τραγουδιού
- `lblArtist`: Ένδειξη του ονόματος του Καλλιτέχνη ενός τραγουδιού
- `imgSong`: Εικόνα που αντιστοιχεί στο συγκεκριμένο τραγούδι

Μέθοδοι της κλάσης

- `SongGUI()`: Constructor διεπαφής `SongGUI`
- `onClickTitle()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει πάνω στον τίτλο του τραγουδιού που αναπαράγεται και μεταβαίνει στο `FullMusicPlayerGUI`.
- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

• EditPlayQueueGUI

<<Boundary>> EditPlayQueueGUI	
- lblTitle: Label	
- buttonBack: Button	
- lblPlayingNow: Label	
- lblPlayingNext: Label	
- buttonMoveSong: Button	
+ EditPlayQueueGUI(): void	
+ onClickBack(): void	
+ onMove(): void	
+ display(): void	

Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας επεξεργασίας της PlayQueue του χρήστη.

Χαρακτηριστικά της κλάσης

- lblTitle: Τίτλος της σελίδας
- buttonBack: Κουμπί για την επιστροφή στην προηγούμενη σελίδα
- lblPlayingNow: Ένδειξη του τραγουδιού που αναπαράγεται/παίζεται αυτή τη στιγμή
- lblPlayingNext: Τίτλος του επόμενου τραγουδιού της PlayQueue
- buttonMoveSong: Κουμπί που επιτρέπει την μετακίνηση ενός τραγουδιού, αλλάζοντας την σειρά αναπαραγωγής των τραγουδιών στην PlayQueue.

Μέθοδοι της κλάσης

- EditPlayQueueGUI(): Constructor διεπαφής EditPlayQueueGUI
- onClickBack(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει στο buttonBack.
- onMove(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει στο buttonMoveSong και προσπαθήσει να μετακινήσει ένα τραγούδι.
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

• MusicPlayerGUI

<<Boundary>> MusicPlayerGUI	
- buttonPlay: Button	
- slider: Slider	
+ MusicPlayerGUI(): void	
+ onClickPlay(): void	
+ onClickSlider(): void	
+ display(): void	

Η κλάση αυτή υλοποιεί τη διεπαφή εμφάνισης του MusicPlayer.

Χαρακτηριστικά της κλάσης

- buttonPlay: κουμπί για να πραγματοποιήσει ο χρήστης παύση/αναπαραγωγή ενός τραγουδιού
- slider: είναι ένα widget, με το οποίο μπορεί ο χρήστης να επιλέξει την επιθυμητή χρονική στιγμή αναπαραγωγής ενός τραγουδιού

Μέθοδοι της κλάσης

- MusicPlayerGUI(): Constructor διεπαφής MusicPlayerGUI
- onClickPlay(): Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το buttonPlay, για να αναπαραχθεί/παύσει ένα τραγούδι.
- onClickSlider(): Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει πάνω σε κάποιο σημείο του slider για να μεταφερθεί στο επιθυμητό σημείο αναπαραγωγής του τραγουδιού.
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● FullMusicPlayerGUI

<<Boundary>> FullMusicPlayerGUI	
- lblPlaylist: Label	
- lblArtist: Label	
- lblAlbum: Label	
- lblSong: Label	
- imgCover: Img	
- buttonPlay: Button	
- buttonShuffle: Button	
- buttonRepeat: Button	
- buttonNext: Button	
- buttonPrevious: Button	
- buttonAddToPlaylist: Button	
- buttonPlayQueue: Button	
- slider: Slider	
+ FullMusicPalyerGUI(): void	
+ onClickPlay(): void	
+ onClickShuffle(): void	
+ onClickRepeat(): void	
+ onClickNext(): void	
+ onClickPrevious(): void	
+ onClickAddToPlaylist(): void	
+ onClickPlayQueue(): void	
+ onClickSlider(): void	
+ displayEditPlayQueue(): void	
+ display(): void	

Η κλάση αυτή υλοποιεί τη διεπαφή εμφάνισης του FullMusicPlayer, με δυνατότητα επεξεργασίας της PlayQueue.

Χαρακτηριστικά της κλάσης

- lblPlaylist: Τίτλος της Playlist
- lblArtist: Όνομα Καλλιτέχνη του τραγουδιού
- lblAlbum: Όνομα του άλμπουμ
- lblSong: Όνομα του τραγουδιού
- imgCover: Εικόνα του τραγουδιού
- buttonPlay: Κουμπί για να πραγματοποιήσει ο χρήστης παύση/αναπαραγωγή ενός τραγουδιού
- buttonShuffle: Κουμπί για να επιλέξει ο χρήστης την τυχαία ή μη ταξινόμηση της PlayQueue
- buttonRepeat: Κουμπί για να επιλέξει ο χρήστης την επανάληψη ή όχι κάποιου τραγουδιού
- buttonNext: Κουμπί για να επιλέξει ο χρήστης την αναπαραγωγή του επόμενου τραγουδιού.
- buttonPrevious: Κουμπί για να επιλέξει ο χρήστης την αναπαραγωγή του προηγούμενου τραγουδιού.
- buttonAddToPlaylist: Κουμπί για να προσθέσει ο χρήστης ένα τραγούδι σε μία Playlist.
- buttonPlayQueue: Κουμπί για να μεταβεί ο χρήστης στην σελίδα της PlayQueue
- slider: είναι ένα widget, με το οποίο μπορεί ο χρήστης να επιλέξει την επιθυμητή χρονική στιγμή αναπαραγωγής ενός τραγουδιού.

Μέθοδοι της κλάσης

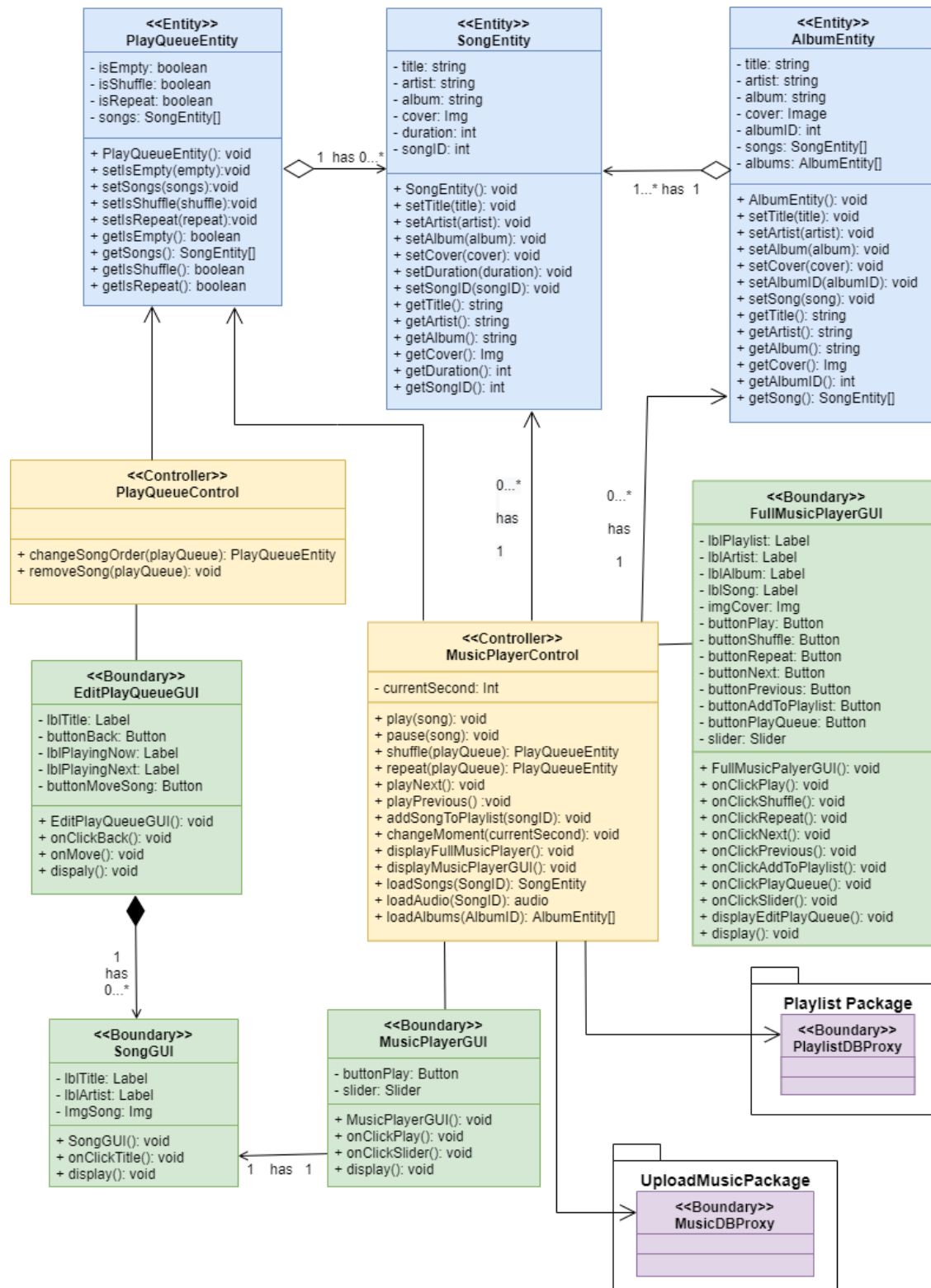
- FullMusicPlayerGUI(): Constructor της FullMusicPlayerGUI
- onClickPlay(): Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το Play, για να αναπαραχθεί/παύσει ένα τραγούδι.
- onClickShuffle(): Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το Shuffle.
- onClickRepeat(): Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το Repeat.
- onClickNext(): Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το Next.



- `onClickPrevious()`: Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το Previous.
- `onClickAddToPlaylist()`: Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το AddToPlaylist.
- `onClickPlayQueue()`: Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει το PlayQueue.
- `onClickSlider()`: Η συνάρτηση αυτή ενεργοποιείται όταν ο χρήστης πατήσει πάνω σε κάποιο σημείο του slider για να μεταφερθεί στο επιθυμητό σημείο αναπαραγωγής του τραγουδιού.
- `displayEditPlayerGUI()`: Η συνάρτηση αυτή καλείται για την εμφάνιση της σελίδας EditPlayer-GUI
- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

Διάγραμμα κλάσεων

Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.

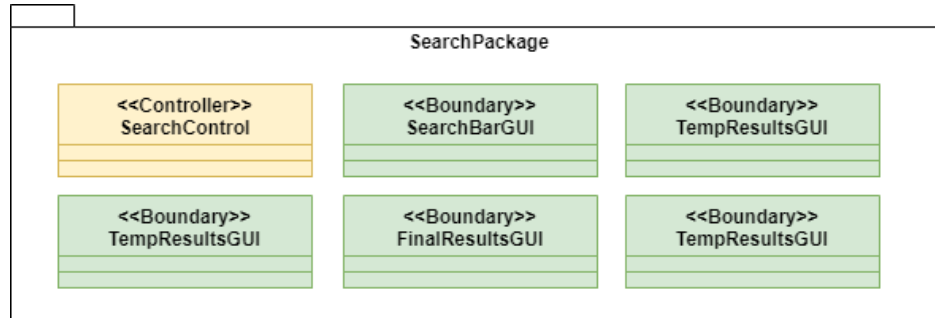


Εικόνα 4.5: Διάγραμμα κλάσεων του Πακέτου PlayMusic

4.2 Πακέτα μεσαίας προτεραιότητας

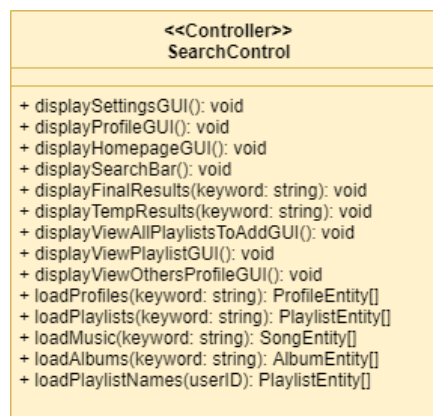
4.2.1 Search Package

Το συγκεκριμένο πακέτο υλοποιεί την αναζήτηση στην εφαρμογή. Επικοινωνεί με την βάση δεδομένων του συστήματος και εμφανίζει στον χρήστη αποτελέσματα σχετικά με την φράση που πληκτρολογεί.



Control Classes

- **SearchControl**



Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για τον έλεγχο της αναζήτησης, προβάλλοντας στον χρήστη τις οντότητες της βάσης δεδομένων που είναι σχετικές με την αναζήτησή του. Επίσης είναι υπεύθυνη για την ανακατεύθυνση του χρήστη στις υπόλοιπες διεπαφές της εφαρμογής ανάλογα με τις ενέργειές του.

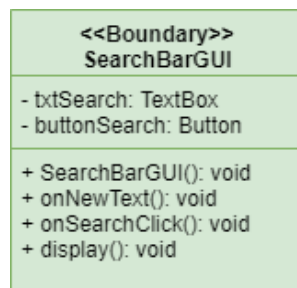
Μέθοδοι της κλάσης

- displaySettingsGUI(): Αυτή η συνάρτηση καλεί την display() του SettingsGUI, όταν ο χρήστης πατήσει το Settings.
- displayProfileGUI(): Αυτή η συνάρτηση καλεί την display() του ProfileGUI, όταν ο χρήστης πατήσει το Profile.
- displayHomePageGUI(): Αυτή η συνάρτηση καλεί την display() του HomePageMasterGUI, όταν ο χρήστης πατήσει το Home.
- displaySearchBar(): Αυτή η συνάρτηση καλεί την display() του SearchBarGUI.
- displayFinalResults(): Αυτή η συνάρτηση καλεί την display() του FinalResultsGUI.
- displayTempResults(): Αυτή η συνάρτηση καλεί την display() του TempResultsGUI.
- displayViewAllPlaylistsToAddGUI(): Αυτή η συνάρτηση καλεί την display() του ViewAllPlaylistsToAddSongGUI.
- displayViewPlaylistGUI(): Αυτή η συνάρτηση καλεί την display() του ViewPlaylistGUI.

- `displayViewOthersProfileGUI()`: Αυτή η συνάρτηση καλεί την `display()` του `displayViewOthersProfileGUI`.
- `loadProfile(keyword)`: Αυτή η συνάρτηση, επικοινωνώντας με το `ProfileDBProxy` επιστρέφει τα `ProfileEntity` που περιέχουν στο όνομα τους το όρισμα `keyword`.
- `loadPlaylists(keyword)`: Αυτή η συνάρτηση, επικοινωνώντας με το `PlaylistDBProxy` επιστρέφει τα `PlaylistEntity` που περιέχουν στο όνομα τους το όρισμα `keyword`.
- `loadSongs(keyword)`: Αυτή η συνάρτηση, επικοινωνώντας με το `MusicDBProxy` επιστρέφει τα `SongEntity` που περιέχουν στο όνομα τους το όρισμα `keyword`.
- `loadAlbums(keyword)`: Αυτή η συνάρτηση, επικοινωνώντας με το `ProfileDBProxy` επιστρέφει τα `AlbumEntity` που περιέχουν στο όνομα τους το όρισμα `keyword`.
- `loadPlaylistNames(userID)`: Αυτή η συνάρτηση, επικοινωνώντας με το `PlaylistDBProxy` επιστρέφει τα `PlaylistEntity` που περιέχουν του χρήστη που έχει το `userID` του ορίσματος.

Boundary Classes

- **SearchBarGUI**



Η κλάση αυτή υλοποιεί την διεπαφή της μπάρας αναζήτησης.

Χαρακτηριστικά της κλάσης

- `txtSearch`: Πεδίο συμπλήρωσης αναζήτησης
- `buttonSearch`: Κουμπί αναζήτησης

Μέθοδοι της κλάσης

- `SearchBarGUI()`: Constructor της διεπαφής `SearchBarGUI`
- `onNewText()`: Αυτή η συνάρτηση καλεί την `displayTempResults()` του `Controller` μόλις ο χρήστης πληκτρολογήσει στο `SearchBar`.
- `onSearchClick()`: Καλεί την συνάρτηση `displayFinalResults()` του ελεγκτή.
- `display()`: Η συνάρτηση που καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● TempResultsGUI

<<Boundary>> TempResultsGUI
- buttonSettings: Button - buttonHomepage: Button - buttonProfile: Button - buttonSearch: Button
+ TempResultsGUI(): void + onClickSettings(): void + onClickHomePage(): void + onClickProfile(): void + onSearchClick(): void + display(): void

Η κλάση αυτή υλοποιεί την διεπαφή των προτεινόμενων αποτελεσμάτων της αναζήτησης του χρήστη πριν αυτός πατήσει το κουμπί της αναζήτησης.

Χαρακτηριστικά της κλάσης

- buttonHomePage: Κουμπί για μετάβαση στην Αρχική Σελίδα
- buttonProfile: Κουμπί για μετάβαση στο προφίλ του χρήστη
- buttonSettings: Κουμπί για μετάβαση στις ρυθμίσεις
- buttonSearch: Κουμπί αναζήτησης

Μέθοδοι της κλάσης

- TempResultsGUI(): Constructor της διεπαφής TempResultsGUI
- onClickHomepage(): Καλεί την συνάρτηση display() του HomePageMasterGUI.
- onClickProfile(): Καλεί την συνάρτηση displayProfileGUI() του ελεγκτή.
- onClickSettings(): Καλεί την συνάρτηση displaySettingsGUI() του ελεγκτή.
- onSearchClick(): Καλεί την συνάρτηση displayFinalResults() του ελεγκτή.
- display(): Η συνάρτηση καλείται κι εμφανίζει αποτελέσματα στον χρήστη σχετικά με την φράση που πληκτρολογεί, πριν πατήσει το κουμπί της αναζήτησης . Συγκεκριμένα εμφανίζει τα αποτελέσματα σε λίστα όπως αυτά εμφανίζονται στα αντίστοιχα επιμέρους GUIs (ProfileSearchGUI, AlbumGUI, ViewPlaylistGUI, SongGUI)

● FinalResultsGUI

<<Boundary>> FinalResultsGUI
- buttonSettings: Button - buttonHomepage: Button - buttonProfile: Button - lblProfiles: Label - lblSongs: Label - lblAlbums: Label - lblPlaylists: Label - buttonAddSongInPlaylist: Button
+ FinalResultsGUI(): void + onClickSettings(): void + onClickHomePage(): void + onClickProfile(): void + onClickAddSongInPlaylist(): void + onClickPlaylist(): void + onClickProfile(): void + display(): void

Η κλάση αυτή υλοποιεί την διεπαφή των τελικών αποτελεσμάτων της αναζήτησης του χρήστη αφού αυτός πατήσει το κουμπί της αναζήτησης.

Χαρακτηριστικά της κλάσης

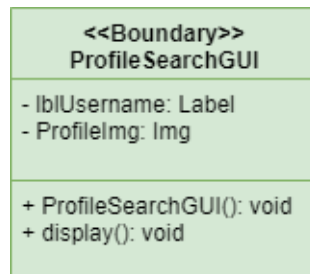
- buttonHomePage: Κουμπί για μετάβαση στην Αρχική Σελίδα
- buttonProfile: Κουμπί για μετάβαση στο προφίλ του χρήστη

- buttonSettings: Κουμπί για μετάβαση στις ρυθμίσεις
- lblProfiles: Ένδειξη ότι τα παρακάτω αποτελέσματα είναι Profile.
- lblSongs: Ένδειξη ότι τα παρακάτω αποτελέσματα είναι Song.
- lblAlbums: Ένδειξη ότι τα παρακάτω αποτελέσματα είναι Album
- lblPlaylists: Ένδειξη ότι τα παρακάτω αποτελέσματα είναι Playlist
- buttonAddSongInPlaylist: Κουμπί για την προσθήκη τραγουδιού σε Playlist

Μέθοδοι της κλάσης

- FinalResultsGUI(): Constructor της διεπαφής FinalResultsGUI
- onClickHomepage(): Καλεί την συνάρτηση display() του HomePageMasterGUI.
- onClickProfile(): Καλεί την συνάρτηση displayProfileGUI() του ελεγκτή.
- onClickSettings(): Καλεί την συνάρτηση displaySettingsGUI() του ελεγκτή.
- onClickAddSongInPlaylist(): Καλεί την συνάρτηση displayViewAllPlaylistsToAddGUI() του ελεγκτή.
- onClickPlaylist(): Καλεί την συνάρτηση displayViewPlaylistGUI() του ελεγκτή.
- onClickProfile(): Καλεί την συνάρτηση displayViewOthersProfileGUI() του ελεγκτή.
- display(): Η συνάρτηση καλείται κι εμφανίζει τα τελικά αποτελέσματα στον χρήστη. Συγκεκριμένα εμφανίζει σε μορφή λίστας τις τέσσερις κατηγορίες αποτελεσμάτων, δηλαδή Profiles, Songs, Albums και Playlist με την αντίστοιχη ετικέτα, καθώς και τα αποτελέσματα ανά κατηγορία όπως αυτά εμφανίζονται στα αντίστοιχα επιμέρους GUIs (ProfileSearchGUI, AlbumGUI, ViewPlaylistGUI, SongGUI).

• ProfileSearchGUI



Η κλάση αυτή υλοποιεί τη διεπαφή του Profile όπως εμφανίζεται στα αποτελέσματα της αναζήτησης.

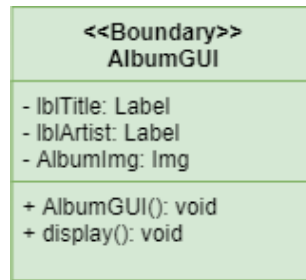
Χαρακτηριστικά της κλάσης

- lblUsername: Το όνομα του χρήστη που αναζητείται
- ProfileImg: Η εικόνα προφίλ του χρήστη που αναζητείται

Μέθοδοι της κλάσης

- ProfileSearchGUI(): Constructor της διεπαφής ProfileSearchGUI
- display(): Η συνάρτηση καλείται κι εμφανίζει στα αποτελέσματα της αναζήτησης την εικόνα προφίλ του χρήστη καθώς και το όνομά του. Στην περίπτωση που υπάρχουν πολλά Profiles για τη συγκεκριμένη αναζήτηση, η συνάρτηση εμφανίζει τα παραπάνω σε μορφή λίστας.

- **AlbumGUI**



Η κλάση αυτή υλοποιεί τη διεπαφή του Album όπως εμφανίζεται στα αποτελέσματα της αναζήτησης.
Χαρακτηριστικά της κλάσης

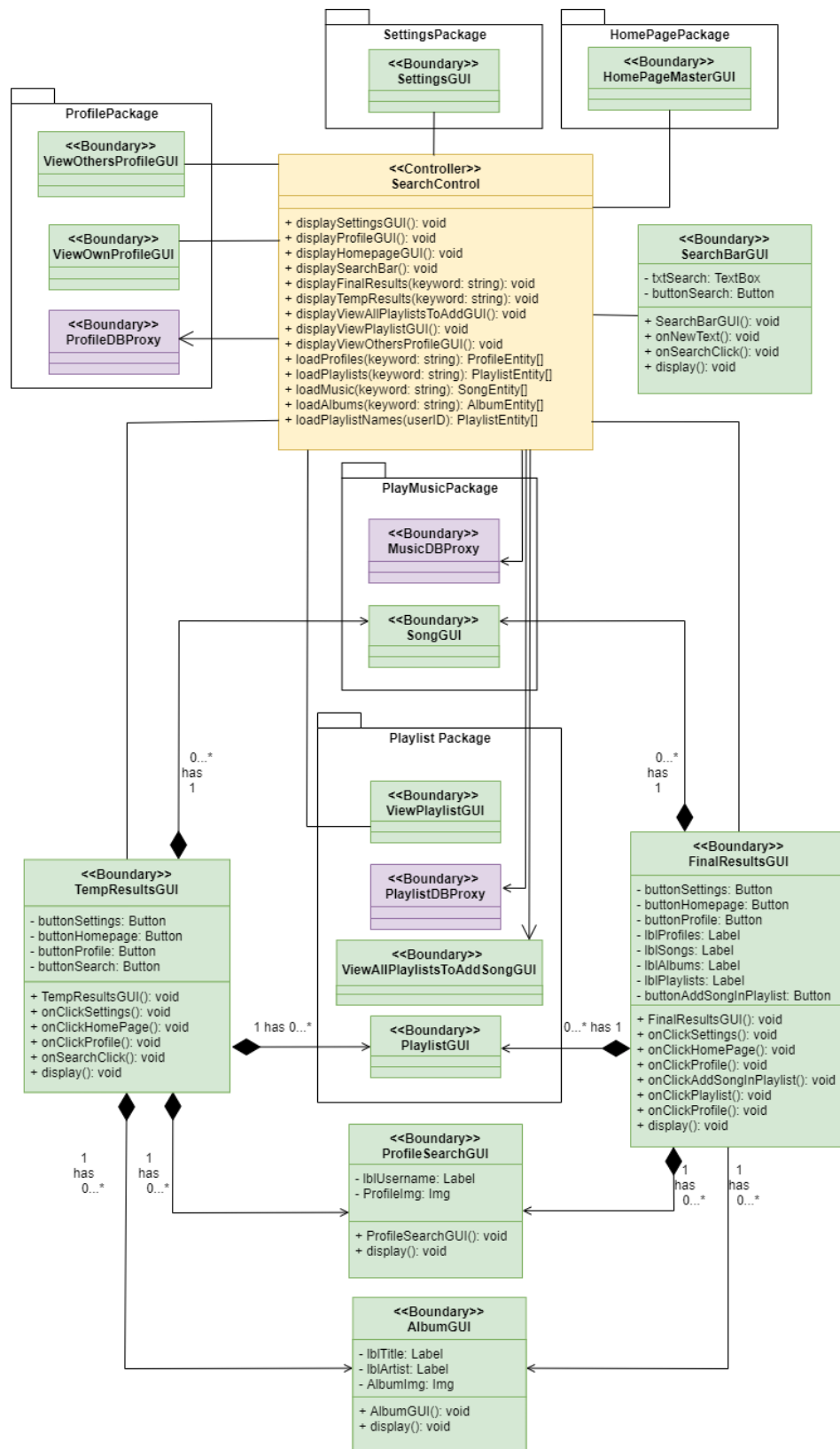
- lblTitle: Ο τίτλος του Album που αναζητείται
- lblArtist: Ο Artist του Album που αναζητείται
- albumImg: Το εξώφυλλο του Album που αναζητείται

Μέθοδοι της κλάσης

- AlbumGUI(): Constructor της διεπαφής AlbumGUI
- display(): Η συνάρτηση καλείται κι εμφανίζει στα αποτελέσματα της αναζήτησης την εικόνα εξωφύλλου του άλμπουμ καθώς και το όνομά του. Στην περίπτωση που υπάρχουν πολλά άλμπουμ για τη συγκεκριμένη αναζήτηση, η συνάρτηση εμφανίζει τα παραπάνω σε μορφή λίστας.

Διάγραμμα κλάσεων

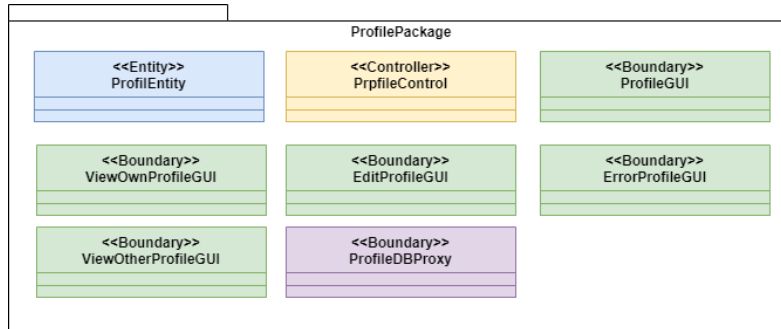
Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.6: Διάγραμμα Κλάσεων Πακέτου Search

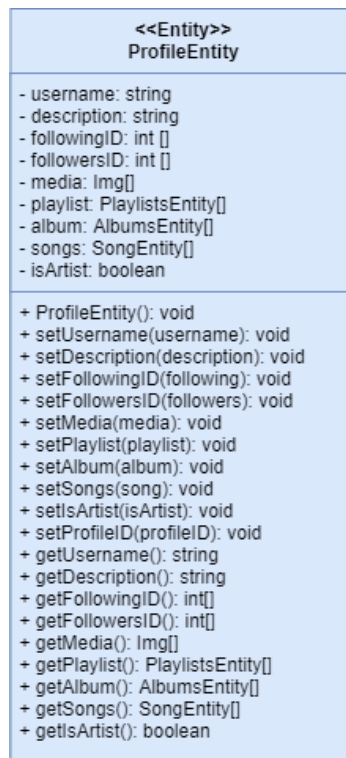
4.2.2 Profile Package

Σκοπός αυτού του πακέτου είναι να παρουσιάζει κάποιες πληροφορίες και τις μουσικές προτιμήσεις του χρήστη και να δώσει τη δυνατότητα στους χρήστες να ακολουθήσουν άτομα με κοινές μουσικές προτιμήσεις. Παίρνει από τη βάση τα αποθηκευμένα στοιχεία και τα παρουσιάζει στο προφίλ του καθενός. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Entity Classes

- **ProfileEntity**



Αυτή είναι κλάση οντότητας του προφίλ του χρήστη. Περιέχει τις απαραίτητες πληροφορίες για αυτόν και δίνει τη δυνατότητα επεξεργασίας τους. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- **username:** Το όνομα που επιλέγει ο χρήστης να εμφανίζεται στο προφίλ του
- **description:** Η περιγραφή στο προφίλ του χρήστη
- **followingID:** Πίνακας με το σύνολο των ατόμων που ακολουθεί ο συγκεκριμένος χρήστης
- **followersID:** Πίνακας με το σύνολο των ατόμων που ακολουθούν τον συγκεκριμένο χρήστη
- **media:** Πίνακας με το σύνολο των συνδεδεμένων προφίλ σε άλλα social media του χρήστη
- **playlist:** Πίνακας με το σύνολο των Playlist στο προφίλ ενός χρήστη.

- album: Πίνακας με το σύνολο των τραγουδιών που έχουν προστεθεί στα άλμπουμ ενός Καλλιτέχνη.
- songs: Πίνακας με το σύνολο των τραγουδιών που έχουν προστεθεί στις Playlist του χρήστη
- isArtist: Μεταβλητή που κάνει τον διαχωρισμό ανάμεσα σε Listener / Artist για δεδομένο χρήστη. Έχει τιμή true αν ο χρήστης εγγράφεται ως Artist και false αν εγγράφεται ως Listener.

Μέθοδοι της κλάσης

- ProfileEntity(): Constructor του ProfileEntity
- setUsername(username): Η συνάρτηση αυτή θέτει το όνομα στο προφίλ του χρήστη.
- setDescription(description): Η συνάρτηση αυτή θέτει την περιγραφή στο προφίλ του χρήστη.
- setFollowingID(following): Η συνάρτηση αυτή θέτει τον αριθμό των follower του χρήστη.
- setFollowersID (followers): Η συνάρτηση αυτή θέτει τον αριθμό των following του χρήστη.
- setMedia(media): Η συνάρτηση αυτή θέτει την περιγραφή στο προφίλ του χρήστη.
- setPlaylist(playlist): Η συνάρτηση αυτή θέτει τις Playlist στο προφίλ του χρήστη.
- setAlbum(album): Η συνάρτηση αυτή θέτει τα Album στο προφίλ του Καλλιτέχνη
- setSongs(song): Η συνάρτηση που θέτει τα τραγούδια στις Playlists του χρήστη.
- setIsArtist(isArtist): Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στη μεταβλητή που καθορίζει το εάν ο χρήστης είναι Listener ή Artist
- getUsername(): Η συνάρτηση αυτή επιστρέφει το όνομα χρήστη
- getDescription(): Η συνάρτηση αυτή επιστρέφει την περιγραφή του χρήστη
- getFollowersID(): Η συνάρτηση αυτή επιστρέφει τους ακόλουθους του χρήστη
- getFollowingID (): Η συνάρτηση αυτή επιστρέφει τους άλλους χρήστες που ακολουθούν τον χρήστη
- getMedia(): Η συνάρτηση που επιστρέφει τα υπόλοιπα συνδεδεμένα με τον χρήστη social media.
- getPlaylist(): Η συνάρτηση που επιστρέφει το σύνολο των Playlist στο προφίλ του χρήστη.
- getAlbum(): Η συνάρτηση που επιστρέφει το σύνολο των άλμπουμ στο προφίλ του Καλλιτέχνη.
- getSongs(): Η συνάρτηση που επιστρέφει το σύνολο των τραγουδιών στις Playlists του χρήστη.
- getIsArtist(): Η συνάρτηση αυτή επιστρέφει true εάν ο χρήστης είναι Artist και false εάν είναι Listener.

Control Classes

• ProfileControl

<<Controller> ProfileControl	
- userID: int	
+ updateProfileInfo(userID): ProfileEntity	
+ follow(userID): void	
+ unfollow(userID): void	
+ changePrivacy(playlistID): PlaylistEntity	
+ deletePlaylist(playlistID): PlaylistControl	
+ showPlaylist(): void	
+ showEditPlaylistOption(): void	
+ displayEditPlaylist(): void	
+ displayViewPlaylist(): void	
+ displayCreatePlaylist(): void	
+ updateFollowers(followersID: ProfileEntity): void	
+ loadProfileInfo(userID): ProfileEntity	
+ storeProfileInfo(userID): ProfileEntity	
+ storeStatistics(activity: ActivityEntity): void	
+ amIFollowing(userID): boolean	
+ isProfileMine(userID): boolean	
+ displayViewOthersProfile(): void	
+ displayProfile(): void	
+ displayEditProfile(): void	
+ displayHomePageMaster(): void	
+ displayErrorProfile(): void	
+ displaySettings(): void	
+ search(keyword: string): ProfileEntity	

Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για την προβολή και επεξεργασία ενός προφίλ, εμφανίζει τις κατάλληλες οθόνες ανάλογα με τον αν ο χρήστης βλέπει το δικό του προφίλ ή κάποιου άλλου. Επίσης είναι υπεύθυνη για την σύνδεση και την ανακατεύθυνση των κατάλληλων διεπαφών στον χρήστη.

Χαρακτηριστικά της κλάσης

- userID: Ένας μοναδικός και χαρακτηριστικός αριθμός για κάθε χρήστη.

Μέθοδοι της κλάσης

- updateProfileInfo(userID): Η συνάρτηση αυτή καλείται ώστε να γίνει η ενημέρωση των στοιχείων του χρήστη.
- follow(userID): Συνάρτηση που επιτρέπει στον χρήστη να ακολουθήσει κάποιον χρήστη που δεν ακολουθεί ήδη.
- unfollow(userID): Συνάρτηση που επιτρέπει στον χρήστη να πάψει να ακολουθεί κάποιον χρήστη που ακολουθεί ήδη.
- changePrivacy(playlistID): Συνάρτηση που αλλάζει την ιδιωτικότητα της Playlist, καλώντας τις getPublic() και setPublic() του Playlist Entity.
- deletePlaylist(playlistID): Καλεί την συνάρτηση deletePlaylist() του ελεγκτή PlaylistControl.
- showPlaylist(): Η συνάρτηση αυτή καλεί την display() του PlaylistGUI.
- showEditPlaylistOption(): Η συνάρτηση αυτή καλείται ώστε να γίνει προβολή των επιλογών επεξεργασίας μιας Playlist.
- displayEditPlaylist(): Η συνάρτηση αυτή καλεί την display() του EditPlaylistGUI.
- displayViewPlaylist(): Η συνάρτηση αυτή καλεί την display() του ViewPlaylistGUI.
- displayCreatePlaylist(): Η συνάρτηση αυτή καλεί την display() του CreatePlaylistGUI.
- updateFollowers(followersID): Η συνάρτηση αυτή καλείται ώστε να γίνει η ενημέρωση των ακολούθων του χρήστη.
- loadProfileInfo(userID): Αυτή η συνάρτηση, επικοινωνώντας με το ProfileDBProxy επιστρέφει ProfileEntity, με όρισμα έναν συγκεκριμένο κωδικό.

- `storeProfileInfo(userID)`: Πρόκειται για τη συνάρτηση που καλεί την αντίστοιχη συνάρτηση αποθήκευσης των πληροφοριών του χρήστη στο proxy αντικείμενο `ProfileDBProxy`.
- `storeStatistics(activity)`: Αυτή η συνάρτηση καλεί την συνάρτηση `storeActivity()` αποθήκευσης των πληροφοριών του χρήστη στο proxy αντικείμενο `StatisticsBProxy`.
- `amIFollowing(userID)`: Η συνάρτηση επιστρέφει αν κάποιος χρήστης ακολουθεί κάποιον άλλο χρήστη ή όχι.
- `isProfileMine(userID)`: Η συνάρτηση αυτή επιστρέφει αν το προφίλ αυτό ανήκει στον ίδιο χρήστη ή σε άλλον.
- `displayViewOthersProfile()`: Αυτή η συνάρτηση καλεί την `display()` του `ViewOthersProfileGUI`, όταν ο κάποιος χρήστης μπει στο προφίλ κάποιου άλλου χρήστη.
- `displayProfile()`: Αυτή η συνάρτηση καλεί την `display()` του `ProfileGUI`, όταν ο χρήστης πατήσει το `Profile`.
- `displayEditProfile()`: Η συνάρτηση αυτή καλείται για την εμφάνιση της σελίδας όταν ο χρήστης επιθυμεί να επεξεργαστεί τα στοιχεία του προφίλ του.
- `displayHomePageMaster()`: Αυτή η συνάρτηση καλεί την `display()` του `HomePageMasterGUI`, όταν ο χρήστης πατήσει το `Home`.
- `displayErrorProfile()`: Η συνάρτηση αυτή καλείται για την εμφάνιση της σελίδας όταν υπάρχει κάποιος απαγορευμένος χαρακτήρας στα στοιχεία που τροποποιήθηκαν.
- `displaySettings()`: Αυτή η συνάρτηση καλεί την `display()` του `SettingsGUI`, όταν ο χρήστης πατήσει το `Settings`.
- `search(keyword)`: Η συνάρτηση αυτή, επικοινωνώντας με το `SearchDBProxy` επιστρέφει `ProfileEntity`, με όρισμα μία συγκεκριμένη λέξη κλειδί.

Boundary Classes

ViewOtherProfileGUI

<<Boundary>> ViewOthersProfileGUI	
-	lblUsername: Label
-	buttonFollowUnfollow: Button
+	ViewOthersProfileGUI(): void
+	onClickFollowUnfollow(amIFollowing): void
+	displayFollowUnfollow(amIFollowing): void
+	displayViewPlaylist(): void
+	displayMediaPlayer(): void
+	display(): void

Η κλάση αυτή κληρονομεί την `ProfileGUI` και υλοποιεί τη διεπαφή προβολής του προφίλ ενός χρήστη όπως το βλέπει κάποιος άλλος.

Χαρακτηριστικά της κλάσης

- `lblUsername`: Ένδειξη του ονόματος του χρήστη
- `buttonFollowUnfollow`: Κουμπί με το οποίο ο χρήστης έχει να επιλογή να ακολουθεί ή όχι κάποιον άλλο χρήστη.

Μέθοδοι της κλάσης

- `ViewOthersProfileGUI()`: Constructor της διεπαφής `ViewOthersProfileGUI`
- `onClickFollowUnfollow(amIFollowing)`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το `buttonFollowUnfollow` και καλεί την `follow()` του ελεγκτή όταν το `amIFollowing` είναι `false`, και την `unfollow()` όταν είναι `true`.
- `displayFollowUnfollowButton(amIFollowing)`: Τροποποιεί την εμφάνιση του `buttonFollowUnfollow` ανάλογα με το αν ο χρήστης ακολουθεί ή όχι τον άλλο χρήστη.

- `displayViewPlaylist()`: Η συνάρτηση αυτή καλείται για την εμφάνιση όλων των Playlist του χρήστη.
- `displayMusicPlayer()`: Καλεί την συνάρτηση `displayMusicPlayerGUI()` του ελεγκτή
- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα δηλαδή όταν η `isProfileMine()` του ελεγκτή επιστρέφει `false`.

● ProfileGUI

<<Boundary>> ProfileGUI	
-	<imgprofile: img<br=""></imgprofile:> lblFollowing: Label lblFollowers: Label imgMedia: Img lblPlaylists: Label buttonHomePage: Button buttonProfile: Button buttonSettings: Button
+	ProfileGUI(): void onClickHomepage(): void onClickProfile(): void onClickSettings(): void isProfileMine(): boolean displayViewPlaylist(): void displayAllPlaylists(): void display(): void

Η κλάση αυτή υλοποιεί τη διεπαφή του μέρους της προβολής προφίλ που είναι κοινό και για τον ίδιο το χρήστη αλλά και για άλλους χρήστες.

Χαρακτηριστικά της κλάσης

- `imgProfile`: Η εικόνα προφίλ του χρήστη
- `lblFollowing`: Ένδειξη όσων ακολουθεί ο χρήστης
- `lblFollowers`: Ένδειξη των ακολούθων του χρήστη
- `imgMedia`: Εικόνα σχετική με άλλα social media
- `lblPlaylists`: Τίτλος "Playlists" στο προφίλ του χρήστη
- `buttonHomepage`: Κουμπί για μετάβαση στην Αρχική Σελίδα
- `buttonProfile`: Κουμπί για μετάβαση στο προφίλ του χρήστη
- `buttonSettings`: Κουμπί για μετάβαση στις ρυθμίσεις

Μέθοδοι της κλάσης

- `ProfileGUI()`: Constructor της διεπαφής ProfileGUI
- `onClickHomepage()`: Καλεί την συνάρτηση `displayHomePageMaster()` του ελεγκτή.
- `onClickProfile()`: Καλεί την συνάρτηση `displayProfile()` του ελεγκτή.
- `onClickSettings()`: Καλεί την συνάρτηση `displaySettings()` του ελεγκτή.
- `isProfileMine()`: Η συνάρτηση αυτή επιστρέφει αν το προφίλ αυτό έχει ίδιο `userID` με τον χρήστη ή διαφορετικό, ώστε να καθοριστεί το γραφικό περιβάλλον.
- `displayViewPlaylist()`: Η συνάρτηση αυτή καλείται για την εμφάνιση όλων των τραγουδιών μιας Playlist.
- `displayAllPlaylists()`: Η συνάρτηση αυτή καλεί τη `showPlaylist()` του ελεγκτή και εμφανίζει σε μορφή λίστας τις Playlists του χρήστη.
- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● ViewOwnProfileGUI

<<Boundary>> ViewOwnProfileGUI	
- lblUsername: Label	
- buttonPen: Button	
- buttonDots: Button	
- lblDescription: Label	
- imgMedia: Img[]	
- buttonEditPlaylist: Button	
- buttonDeletePlaylist: Button	
- buttonPublicPrivatePlaylist: Button	
- buttonAddPlaylist: Button	
+ ViewOwnProfileGUI(): void	
+ onClickPen(): void	
+ onClickDots(): void	
+ onClickSaveChanges(): void	
+ onClickEditPlaylist(): void	
+ onClickDeletePlaylist(): void	
+ onClickPublicPrivatePlaylist(): void	
+ onClickAddPlaylist(): void	
+ displayEditProfile(): void	
+ displayMusicPlayer(): void	
+ display(): void	

Η κλάση αυτή κληρονομεί την ProfileGUI και υλοποιεί τη διεπαφή προβολής του προφίλ όπως το βλέπει ο ίδιος ο χρήστης.

Χαρακτηριστικά της κλάσης

- lblUsername: Ένδειξη του ονόματος του χρήστη
- buttonPen: Κουμπί για την επεξεργασία στα στοιχεία του προφίλ του χρήστη
- buttonDots: Κουμπί για την επεξεργασία κάθε Playlist στο προφίλ του χρήστη
- lblDescription: Ένδειξη της περιγραφής του χρήστη
- imgMedia: Εικόνα σχετική με άλλα social media
- buttonEditPlaylist: Κουμπί για την επεξεργασία μιας Playlist
- buttonDeletePlaylist: Κουμπί για την διαγραφή μιας Playlist
- buttonPublicPrivatePlaylist: Κουμπί που καθιστά μια Playlist ιδιωτική ή δημόσια.
- buttonAddPlaylist: Κουμπί για να προσθέσει ο χρήστης άλλα social media στο προφίλ του

Μέθοδοι της κλάσης

- ViewOwnProfileGUI(): Constructor της διεπαφής ViewOwnProfileGUI
- onClickPen(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το μολυβάκι και καλεί την displayEditProfile().
- onClickDots(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει τις τρεις τελείες δίπλα στις Playlists του.
- onClickSaveChanges(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το SaveChanges.
- onClickEditPlaylist(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει την επιλογή Edit στις τρεις τελείες και καλεί την displayEditPlaylist() του ελεγκτή.
- onClickDeletePlaylist(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει την επιλογή Delete στις τρεις τελείες και καλεί την deletePlaylist() του ελεγκτή.
- onClickPublicPrivatePlaylist(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει την επιλογή Private/Public στις τρεις τελείες και καλεί την changePrivacy() του ελεγκτή.
- onClickAddPlaylist(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το "+" δίπλα από τον τίτλο Playlist καλώντας την displayCreatePlaylist() του ελεγκτή.
- onClickEditProfile(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το μολυβάκι καλώντας την displayEditProfile() του ελεγκτή.

- `displayMediaPlayer()`: Καλεί την συνάρτηση `displayMediaPlayerGUI()` του ελεγκτή.
- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

● EditProfileGUI

<<Boundary>> EditProfileGUI	
-	txtUsername: TextBox
-	buttonUploadCover: Button
-	imgMedia: Img[]
-	txtDescription: TextBox
-	buttonAddMedia: Button
-	buttonSaveChanges: Button
+	EditProfileGUI(): void
+	onClickUploadCover(): void
+	onClickAddMedia(): void
+	onClickSaveChanges(): void
+	displayErrorProfile(): void
+	display(): void

Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδα επεξεργασίας του προφίλ του χρήστη και κληρονομεί την `ViewOwnProfileGUI`.

Χαρακτηριστικά της κλάσης

- `txtUsername`: Πεδίο συμπλήρωσης που αφορά το όνομα του χρήστη
- `buttonUploadCover`: Κουμπί για το Upload εικόνας στο προφίλ του
- `imgMedia`: Εικόνα σχετική με άλλα social media
- `txtDescription`: Πεδίο συμπλήρωσης που αφορά την περιγραφή
- `buttonAddMedia`: Κουμπί για να προσθέσει ο χρήστης άλλα social media στο προφίλ του
- `buttonSaveChanges`: Κουμπί για να αποθηκεύσει ο χρήστης τις αλλαγές στο προφίλ του

Μέθοδοι της κλάσης

- `EditProfileGUI()`: Constructor της διεπαφής `EditProfileGUI`
- `onClickUploadCover()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει πάνω στο σημείο αλλαγής της εικόνας προφίλ του και θα καλεί την `updateProfileInfo()` του ελεγκτή.
- `onClickAddMedia()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει πάνω στο `imgMedia` και καλεί την `updateProfileInfo()` του ελεγκτή και μετά το `displayViewOwnProfile()`.
- `onClickSaveChanges()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το `buttonSaveChanges`, καλεί την `updateProfileInfo()` του ελεγκτή και μετά το `displayViewOwnProfile()`.
- `displayErrorProfile()`: Η συνάρτηση αυτή καλείται για την εμφάνιση της σελίδας όταν υπάρχει κάποιος απαγορευμένος χαρακτήρας στα στοιχεία που τροποποιήθηκαν.
- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

• ErrorProfileGUI

<<Boundary>> ErrorProfileGUI
- lblInvalid: Label
+ ErrorProfileGUI(): void + display(): void

Η κλάση αυτή υλοποιεί τη διεπαφή που πληροφορεί το χρήστη σε περίπτωση σφάλματος επεξεργασίας των πληροφοριών του προφίλ του.

Χαρακτηριστικά της κλάσης

- lblInvalid: ένδειξη που εμφανίζεται όταν ο χρήστης προσπαθεί να γράψει κάποιο απαγορευμένο χαρακτήρα κατά την επεξεργασία των στοιχείων του προφίλ του

Μέθοδοι της κλάσης

- ErrorProfileGUI(): Constructor της διεπαφής ErrorProfileGUI
- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.

• ProfileDBProxy

<<Boundary>> ProfileDBProxy
+ loadProfileInfo(userID): ProfileEntity + storeProfileInfo(ProfileEntity): void + search(keyword:string): ProfileEntity

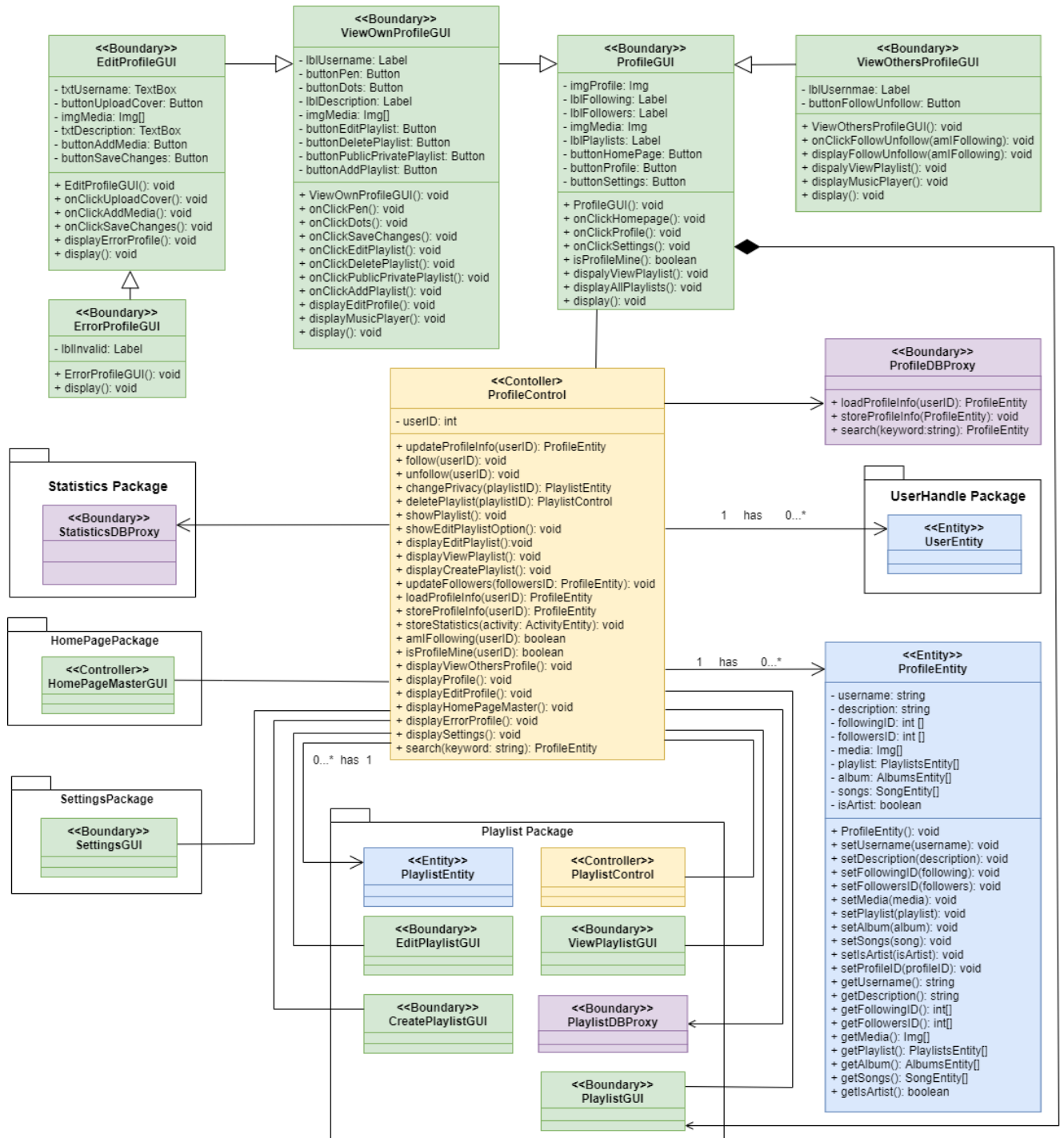
Η κλάση αυτή είναι υπεύθυνη για την αποθήκευση των πληροφοριών στο προφίλ ενός χρήστη.

Μέθοδοι της κλάσης

- loadProfileInfo(userID): Η μέθοδος αυτή επιστρέφει μια οντότητα (ProfileEntity) με τις πληροφορίες της από την βάση, δεδομένου του userID.
- storeProfileInfo(ProfileEntity): Η μέθοδος αυτή αποθηκεύει τις πληροφορίες μιας οντότητας στη βάση δεδομένων.
- search(keyword:string): Η μέθοδος αυτή αναζητά και επιστρέφει μια οντότητα από την βάση δεδομένης μιας λέξης κλειδί.

Διάγραμμα κλάσεων

Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.

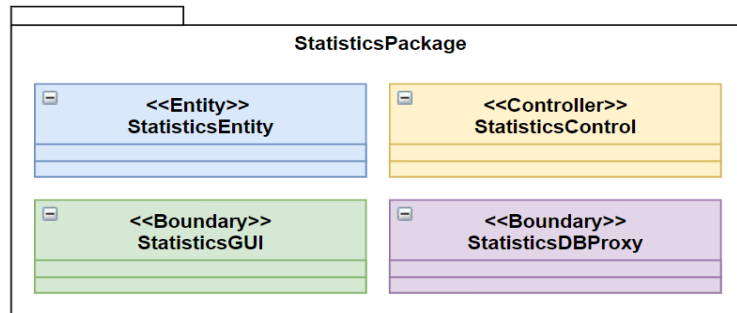


Εικόνα 4.7: Διάγραμμα Κλάσεων Πακέτου Profile

4.3 Πακέτα χαμηλής προτεραιότητας

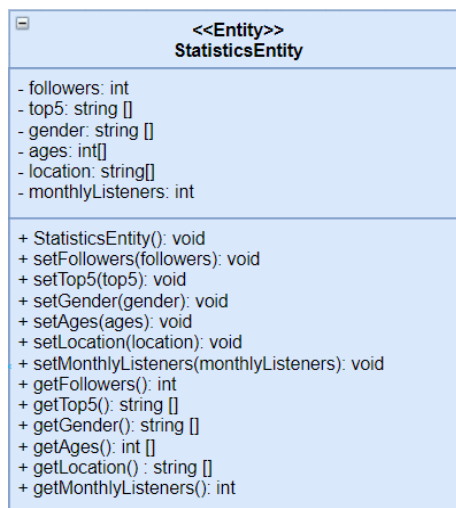
4.3.1 Statistics Package

Σκοπός αυτού του πακέτου είναι να “τραβήξει” από τη βάση και να παρουσιάσει στον Καλλιτέχνη τα στατιστικά που αφορούν τους ακροατές του. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Entity Classes

- StatisticsEntity



Αυτή είναι κλάση οντότητας των στατιστικών και περιέχει στοιχεία απαραίτητα για να έχει ο Καλλιτέχνης μια εποπτεία όσον αφορά τα χαρακτηριστικά του ακροατηρίου του. Συνοδεύεται από τους αντίστοιχους setters & getters.

Χαρακτηριστικά της κλάσης

- followers: Ο αριθμός των followers ενός Καλλιτέχνη
- top5: Μια λίστα με τους κορυφαίους 5 ακροατές του Καλλιτέχνη
- gender: Στοιχεία για την κατανομή των ακροατών βάσει των φύλων τους
- ages: Στοιχεία για την κατανομή των ακροατών βάσει των ηλικιών τους
- location: Στοιχεία για την κατανομή των ακροατών βάσει της τοποθεσίας τους
- monthlyListeners: Αριθμός των μηνιαίων ακροατών του Καλλιτέχνη

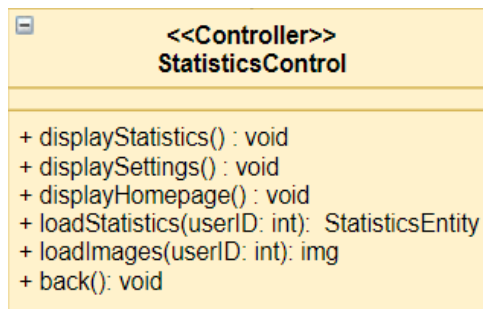
Μέθοδοι της κλάσης

- `StatisticsEntity()`: Constructor της κλάσης `StatisticsEntity` που αναλαμβάνει την αρχικοποίηση των μεταβλητών
- `setFollowers(followers)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `Followers`
- `setTop5(top5)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `Top5`

- `setGender(gender)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `Gender`
- `setAges(ages)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `Ages`
- `setLocation(location)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `Location`
- `setMonthlyListeners(monthlyListeners)`: Η συνάρτηση αυτή θέτει το όρισμα που δέχεται στο `MonthlyListeners`
- `getFollowers()`: Η συνάρτηση αυτή επιστρέφει το περιεχόμενο της μεταβλητής `Followers`
- `getTop5()`: Η συνάρτηση αυτή επιστρέφει το περιεχόμενο της μεταβλητής `Top5`
- `getGender()`: Η συνάρτηση αυτή επιστρέφει το περιεχόμενο της μεταβλητής `Gender`
- `getAges()`: Η συνάρτηση αυτή επιστρέφει το περιεχόμενο της μεταβλητής `Ages`
- `getLocation()`: Η συνάρτηση αυτή επιστρέφει το περιεχόμενο της μεταβλητής `Location`
- `getMonthlyListeners()`: Η συνάρτηση αυτή επιστρέφει το περιεχόμενο της μεταβλητής `MonthlyListeners`

Control Classes

- **StatisticsControl**



Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για την προβολή των στατιστικών του Καλλιτέχνη. Ελέγχει επίσης την ανακατεύθυνση και προβολή των κατάλληλων διεπαφών στον χρήστη.

Μέθοδοι της κλάσης

- `displayStatistics()`: Αυτή η συνάρτηση καλεί την `display()` συνάρτηση του `StatisticsGUI`.
- `displaySettings()`: Αυτή η συνάρτηση καλεί την `display()` συνάρτηση του `SettingsGUI`, ώστε ο χρήστης να πάει στις ρυθμίσεις του.
- `displayHomepage()`: Αυτή η συνάρτηση δίνει τον έλεγχο στον `HomePage Controller`, ώστε ο χρήστης να πάει στην αρχική του σελίδα.
- `back()`: Αυτή η συνάρτηση που καλείται μέσα από το `StatisticsGUI`, επιτρέπει στον χρήστη να γυρίσει πίσω στην αρχική του σελίδα. Καλεί την προαναφερθείσα `showHomepage()`.
- `loadStatistics(userID)`: Έχοντας ως όρισμα ένα `userID`, αυτή η συνάρτηση μας επιστρέφει από τη βάση ένα `UserEntity` με όλα τα στοιχεία του συγκεκριμένου χρήστη.
- `loadImages(userID)`: Έχοντας ως όρισμα ένα `userID`, αυτή η συνάρτηση μας επιστρέφει από τη βάση τις αντίστοιχες εικόνες-γραφήματα που θα συνοδεύσουν τα αριθμητικά στατιστικά του χρήστη.

Boundary Classes

- StatisticsGUI



Η κλάση αυτή υλοποιεί τη διεπαφή που παρέχει στον Καλλιτέχνη πληροφορίες - στατιστικά για το ακροατήριό του.

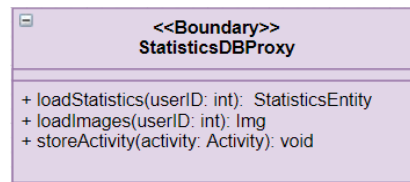
Χαρακτηριστικά της κλάσης

- lblTitle: Τίτλος “Statistics” στην κορυφή της σελίδας
- lblFollowersTitle: Τίτλος με κείμενο “Followers”
- lblFollowers: Αριθμός των Followers του Artist
- lblTop5Title: Τίτλος με κείμενο “Followers”
- lblTop5: Μια λίστα με τους κορυφαίους 5 ακροατές
- lblGenderTitle: Τίτλος με κείμενο “Gender”
- imgGender: Γράφημα με την κατανομή των ακροατών βάσει των φύλων τους
- lblAgesTitle: Τίτλος με κείμενο “Ages”
- imgAges: Γράφημα με την κατανομή των ακροατών βάσει των ηλικιών τους
- lblLocationTitle: Τίτλος με κείμενο “Audience Location”
- lblLocation: Μια λίστα με τις 5 πόλεις στις οποίες υπάρχουν οι περισσότεροι ακροατές
- lblMonthlyListenersTitle: Τίτλος με κείμενο “Monthly Listeners”
- lblMonthlyListeners: Αριθμός μηνιαίων ακροατών παγκοσμίως
- imgMonthlyListeners: Εικόνα σχετική με τους μηνιαίους ακροατές
- buttonBack: Κουμπί για επιστροφή στην αρχική σελίδα του Ακροατή
- buttonSettings: Κουμπί για ανακατεύθυνση στη σελίδα ρυθμίσεων του χρήστη

Μέθοδοι της κλάσης

- StatisticsGUI(): Constructor της διεπαφής StatisticsGUI
- onClickBack(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει στο button “Back”, και καλεί τη συνάρτηση back() του Controller.
- onClickSettings(): Η συνάρτηση ενεργοποιείται όταν ο χρήστης πατήσει στο button “Settings”, και καλεί τη συνάρτηση displaySettings() του Controller.
- display(): Αυτή η συνάρτηση καλείται από την displayStatistics() του ελεγκτή και είναι υπεύθυνη για τον σχεδιασμό της διεπαφής προβολής των στατιστικών του Καλλιτέχνη.

- **StatisticsDBProxy**



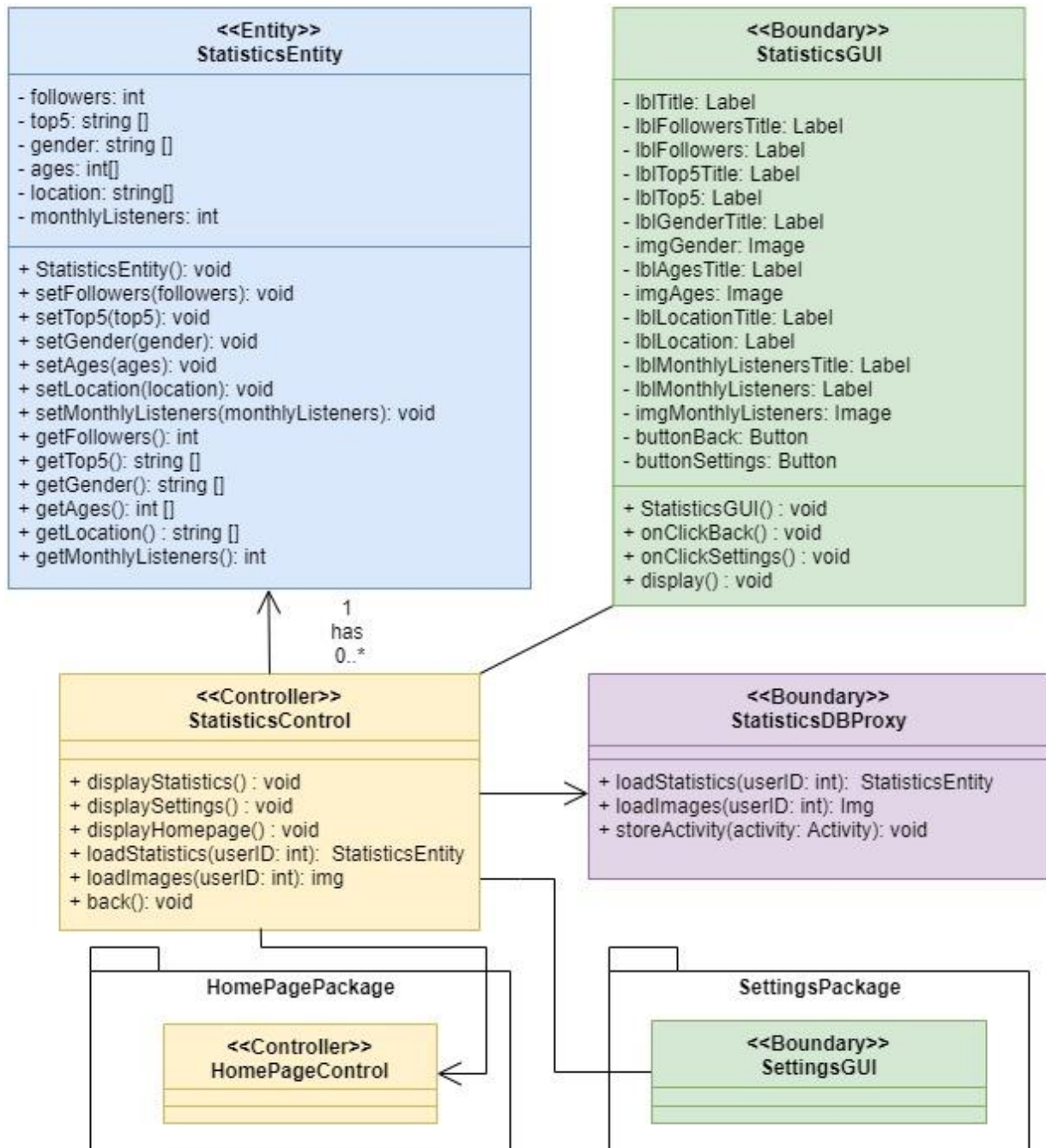
Η κλάση αυτή είναι υπεύθυνη για την αποθήκευση των στατιστικών που αφορούν τους ακροατές των Καλλιτεχνών.

Μέθοδοι της κλάσης

- **loadStatistics(userID):** Αυτή η συνάρτηση φορτώνει στον controller ένα StatisticsEntity, που κατασκευάζει βάσει των Activities που έχουμε αποθηκεύσει.
- **loadImages(userID):** Δουλειά αυτής της μεθόδου είναι να φορτώνει στον controller τα γραφήματα με τα στατιστικά, που και πάλι έχει δομήσει βάσει των Activities που έχουμε αποθηκεύσει.
- **storeActivity(activity):** Αυτή είναι η συνάρτηση που αποθηκεύει στη βάση όλα τα Activities του κάθε χρήστη.

Διάγραμμα κλάσεων

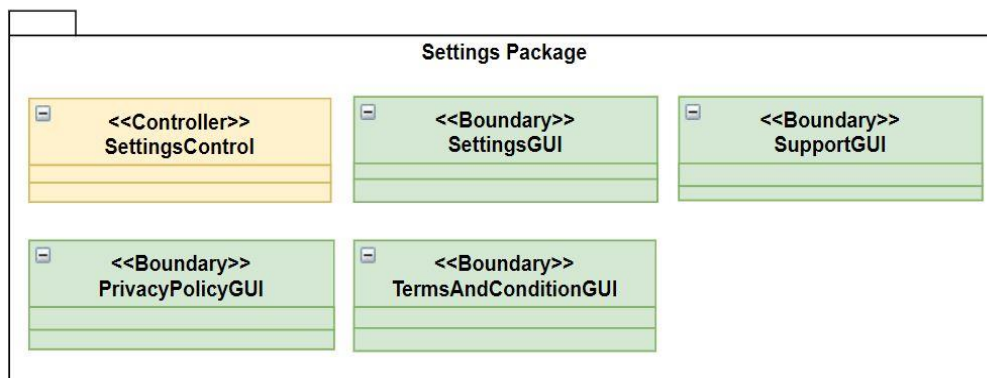
Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.8: Διάγραμμα Κλάσεων Πακέτου Statistics

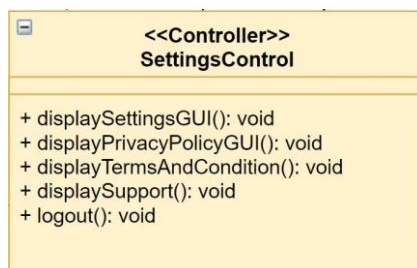
4.3.2 Settings Package

Σκοπός αυτού του πακέτου είναι η προβολή της οθόνης ρυθμίσεων και ο έλεγχος πλοήγησης του χρήστη σε αυτή. Περιλαμβάνει την οθόνη των βασικών επιλογών ρυθμίσεων, αλλά και τις οθόνες 'Support', 'Privacy Policy', 'Terms And Conditions'. Ακόμα, από τη βασική οθόνη ρυθμίσεων ο χρήστης έχει την δυνατότητα να κάνει 'Logout' και έπειτα να βρεθεί πάλι στην Signup Page. Παρακάτω φαίνονται οι κλάσεις που υλοποιούν την προαναφερθείσα λειτουργικότητα.



Control Classes

- **SettingsControl**



Αυτή η κλάση υλοποιεί όλες τις απαραίτητες συναρτήσεις για τον έλεγχο της πλοήγησης του χρήστη στην κεντρική οθόνη ρυθμίσεων. Ελέγχει, δηλαδή, την ανακατεύθυνση και προβολή των κατάλληλων διεπαφών στον χρήστη.

Μέθοδοι της κλάσης

- **displaySettingsGUI():** Πρόκειται για τη συνάρτηση εμφάνισης της κεντρικής σελίδας ρυθμίσεων στον χρήστη που του προσφέρει τις μετέπειτα επιλογές του.
- **displayPrivacyPolicyGUI():** Πρόκειται για τη συνάρτηση εμφάνισης της σελίδας που περιέχει όλες τις απαραίτητες πληροφορίες για την Πολιτική Απορρήτου του Mob, στην οποία ο χρήστης μπορεί να ανατρέχει.
- **displayTermsAndCondition():** Πρόκειται για τη συνάρτηση εμφάνισης της σελίδας που περιέχει όλους τους όρους και τις προϋποθέσεις που έχει αποδεχτεί ο χρήστης εφόσον έχει γίνει μέλος του Mob.
- **displaySupport():** Πρόκειται για τη συνάρτηση εμφάνισης της σελίδας που μπορεί να παρέχει υποστήριξη στον χρήστη ανά πάσα στιγμή, μέσω συχνών ερωτήσεων του Mob Community.

- `logout()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης επιλέξει να εξέλθει από το σύστημα. Επικοινωνεί με την `LoginDBProxy` και αποσυνδέει τον χρήστη, ανακατευθύνοντας τον ξανά στην `Login Page`.

Boundary Classes

• SettingsGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της κεντρικής σελίδας ρυθμίσεων.

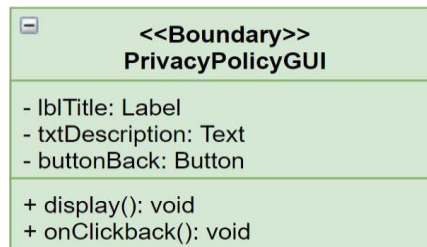
Χαρακτηριστικά της κλάσης

- `lblSetting`: Ονομασία της σελίδας Ρυθμίσεων
- `buttonSettings`: Κουμπί ρυθμίσεων
- `buttonBack`: Κουμπί επιστροφής στην προηγούμενη σελίδα
- `lblTerms`: Ονομασία και περιγραφή της επιλογής 'Terms And Conditions'
- `buttonTerms`: Κουμπί για την επιλογή 'Terms And Conditions'
- `lblPolicy`: Ονομασία και περιγραφή της επιλογής 'Privacy Policy'
- `buttonPolicy`: Κουμπί για την επιλογή 'Privacy Policy'
- `lblSupport`: Ονομασία και περιγραφή της επιλογής 'Support'
- `buttonSupport`: Κουμπί για την επιλογή 'Support'
- `lblLogout`: Ονομασία και περιγραφή της επιλογής 'Logout'
- `buttonLogout`: Κουμπί για την επιλογή 'Logout'

Μέθοδοι της κλάσης

- `display()`: Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- `onClickBack()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back.
- `onClickTerms()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί της επιλογής 'Terms And Conditions' και καλεί την `displayTermsAndConditions()` του controller.
- `onClickPolicy()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί της επιλογής 'Privacy Policy' και καλεί την `displayPrivacyPolicyGUI()` του controller.
- `onClickSupport()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί της επιλογής 'Support' και καλεί την `displaySupport()` του controller.
- `onClickLogout()`: Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί της επιλογής 'Logout' και καλεί την `logout()` του controller.

● PrivacyPolicyGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας Πολιτικής Απορρήτου.

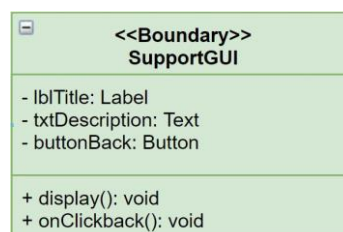
Χαρακτηριστικά της κλάσης

- lblTitle: Ονομασία της σελίδας
- txtDescription: Όλες οι πληροφορίες σχετικά με την Πολιτική Απορρήτου του Mob
- buttonBack: Κουμπί για επιστροφή στην προηγούμενη σελίδα

Μέθοδοι της κλάσης

- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- onClickback(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και καλεί την displaySettingsGUI() του controller.

● SupportGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας υποστήριξης του χρήστη.

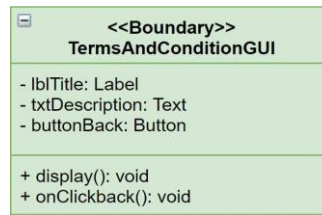
Χαρακτηριστικά της κλάσης

- lblTitle: Ονομασία της σελίδας
- txtDescription: Όλες οι πληροφορίες σχετικά με τις υπηρεσίες του Mob για την υποστήριξη των χρηστών του
- buttonBack: Κουμπί για επιστροφή στην προηγούμενη σελίδα

Μέθοδοι της κλάσης

- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- onClickback(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και καλεί τη displaySettingsGUI() του ελεγκτή.

• TermsAndConditionGUI



Η κλάση αυτή υλοποιεί τη διεπαφή της σελίδας που περιέχει τις πληροφορίες για τους όρους και προϋποθέσεις του Mob.

Χαρακτηριστικά της κλάσης

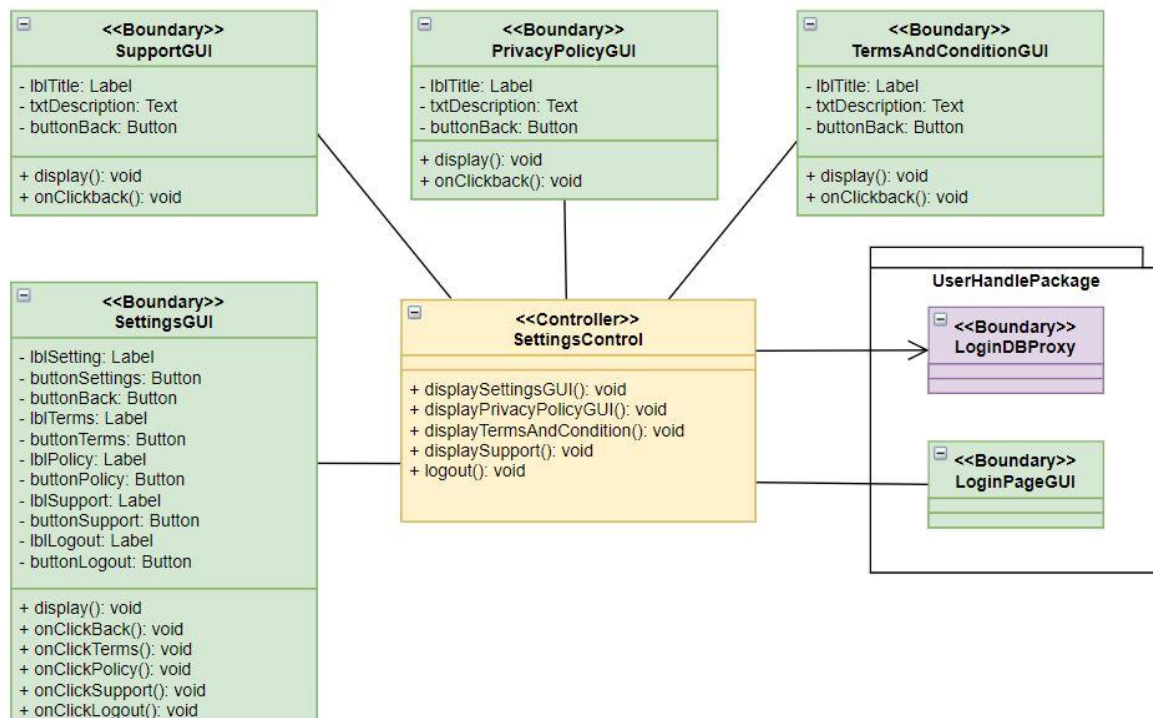
- lblTitle: Ονομασία της σελίδας
- txtDescription: Όλες οι πληροφορίες σχετικά με τους όρους και τις προϋποθέσεις που έχει συμφωνήσει ο χρήστης κατά την εγγραφή του στο Mob
- buttonBack: Κουμπί για επιστροφή στην προηγούμενη σελίδα

Μέθοδοι της κλάσης

- display(): Η συνάρτηση αυτή καλείται όταν χρειάζεται να εμφανιστεί η συγκεκριμένη σελίδα.
- onClickback(): Η συνάρτηση αυτή καλείται όταν ο χρήστης πατήσει το κουμπί Back και καλεί τη displaySettingsGUI() του ελεγκτή.

Διάγραμμα κλάσεων

Σημείωση: Παραλείψαμε τη δεικτοδότηση σε βέλη για ένα προς ένα συσχετίσεις μεταξύ κλάσεων.



Εικόνα 4.9: Διάγραμμα Κλάσεων Πακέτου Settings