Connecting Music

Προδιαγραφές Συστήματος

Del.1.2

Version 0.1

Ιωακειμίδης Χρήστος cioakeim@ece.auth.gr
Μπάλτσας Σπυρίδων mspyrido@ece.auth.gr
Σελιβάνωφ Γεώργιος selivanof@ece.auth.gr
Χατζηιωάννου Λαμπρινός chatziiol@ece.auth.gr

Ημερομηνία

Μέλη της Ομάδας Ανάπτυξης

Όνομα	ΟΑ	Email

Α. Συμεωνίδης	*	asymeon@issel.ee.auth.gr
Ιωακειμίδης Χρήστος	20	cioakeim@ece.auth.gr
Μπάλτσας Σπυρίδων	20	mspyrido@ece.auth.gr
Σελιβάνωφ Γεώργιος	20	selivanof@ece.auth.gr
Χατζηιωάννου Λαμπρινος	20	chatziiol@ece.auth.gr

Πίνακας Περιεχομένων

Λίστα Σχημάτων

1 Στατική Μοντελοποίηση

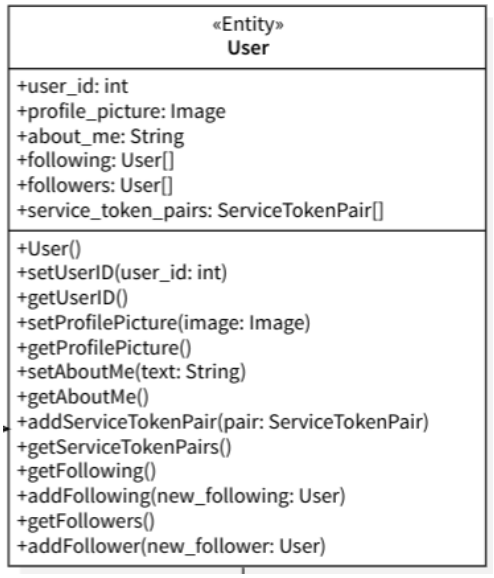
Κατόπιν διευκρίνησης στο discord server του μαθήματος, έχοντας ορίσει τα πακέτα ανεξαρτήτως των features, μα ομαδοποιώντας βάσει περιεχομένου, ακολουθούν τα πακέτα, όπως αυτά δημιουργήθηκαν, καλύπτοντας όλα τα υπάρχοντα features.

- [ ] Λείπει View Recommended boundary

1.1 <Πακέτο κλάσεων User>

1.1.1 Ορισμός Κλάσεων

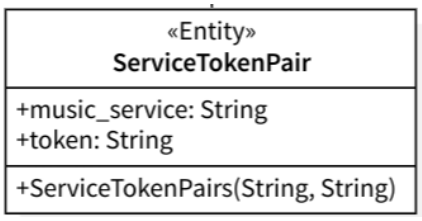
1.1.1.1 Ορισμός Κλάσης User



Χαρακτηριστικά Κλάσης:

- **user_id:**Integer - Το αναγνωριστικό του χρήστη
- **profile_picture:**Image - Η εικόνα προφίλ του χρήστη
- **about_me:**String - Το κείμενο περιγραφής του προφίλ του χρήστη
- **following:**User[] - Λίστα με τους χρήστες που ακολουθούνται από τον χρήστη
- **followers:**User[] - Λίστα με τους χρήστες που ακολουθεί ο χρήστης
- **service_token_pairs:**ServiceTokenPair[] - Όλα τα ServiceTokenPair του χρήστη

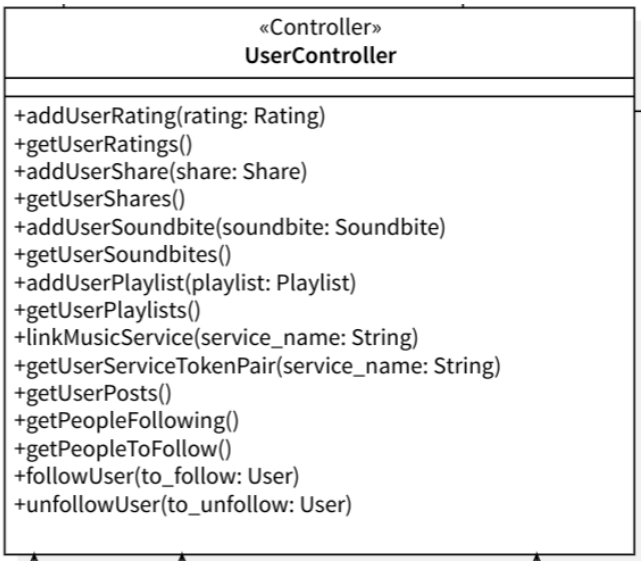
1.1.1.2 Ορισμός Κλάσης ServiceTokenPair



Χαρακτηριστικά Κλάσης:

- **music_service:**String - Το music_service στο οποίο αντιστοιχεί το token
- **token:**String - Το token

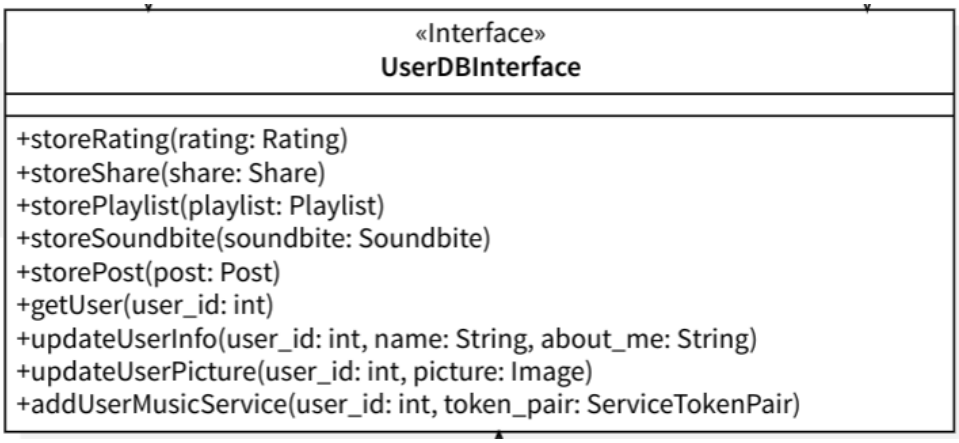
1.1.1.3 Ορισμός Κλάσης UserController



Μέθοδοι Κλάσης:

- **addUserRating**(Rating rating):Void - Αποθηκεύει ένα Rating του User στο UserDB
- **getUserRatings**():Rating[] - Επιστρέφει μια λίστα με όλα τα Rating του User
- **addUserShare**(Share share):Void - Αποθηκεύει ένα Sharetου User στο UserDB
- **getUserShares**():Share[] - Επιστρέφει μια λίστα με όλα τα Share του User
- **addUserSoundbite**(Soundbite soundbite):Void - Αποθηκεύει ένα Soundbite του User στο UserDB
- **getUserSoundbites**():Soundbite[] - Επιστρέφει μια λίστα με όλα τα Soundbite του User
- **addUserPlaylist**(Playlist playlist):Void - Αποθηκεύει ένα Playlist του User στο UserDB
- **getUserPlaylists**():Playlist[] - Επιστρέφει μια λίστα με όλα τα Playlist του User
- **linkMusicService**(String service_name):Void - Συνδέει ένα συγκεκριμένο external music service με τον User (Το επιθυμητό music service προσδιορίζεται από το όρισμα service_name)
- **getUserServiceTokenPair**(String service_name):ServiceTokenPair - Επιστρέφει το ServiceTokenPair του χρήστη για το συγκεκριμένο music service
- **getUserPosts**():Post[]- Επιστρέφει μια λίστα με όλα τα Posts του User
- **getPeopleFollowing**():User[]- Επιστρέφει μια λίστα με όλους τους χρήστες που ακολουθεί ο User
- **getPeopleToFollow**():User[]- Επιστρέφει μια λίστα με χρήστες που πιθανώς να θέλει να ακολουθήσει ο User
- **followUser**(User to_follow):Void- Προσθέτει τον χρήστη to_follow στην λίστα των χρηστών που ακολουθεί ο User
- **unfollowUser**(User to_unfollow):Void- Αφαιρεί τον χρήστη to_unfollow από την λίστα των χρηστών που ακολουθεί ο User

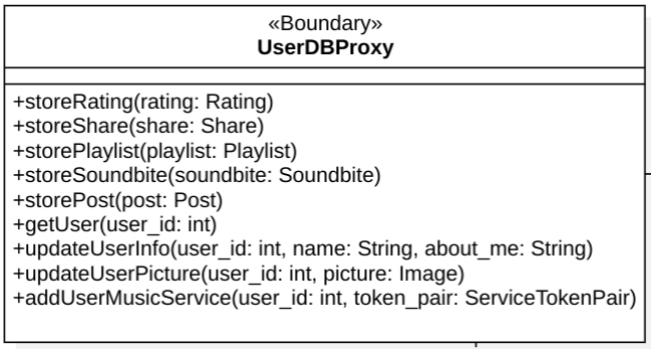
1.1.1.4 Ορισμός Κλάσης UserDBInterface



Μέθοδοι Κλάσης:

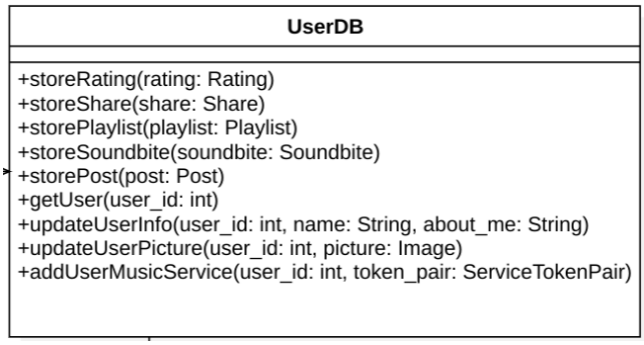
- **storeRating**(Rating rating):Void - Αποθηκεύει ένα Rating στο UserDB
- **storeShare**(Share share):Void - Αποθηκεύει ένα Shareστο UserDB
- **storePlaylist**(Playlist playlist):Void - Αποθηκεύει ένα Playlist στο UserDB
- **storeSoundbite**(Soundbite soundbite):Void - Αποθηκεύει ένα Soundbite στο UserDB
- **storePost**(Post post):Void - Αποθηκεύει ένα Post στο UserDB
- **getUser**(Integer user_id):User- Επιστρέφει ένα αντικείμενο User για τον χρήστη με το συγκεκριμένο user_id
- **updateUserInfo**(Integer user_id, String name, String about_me):Void - Ενημερώνει τις πληροφορίες για τον χρήστη με το συγκεκριμένο user_id
- **updateUserPicture**(Integer user_id, Image picture):Void - Ενημερώνει την εικόνα προφίλ για τον χρήστη με το συγκεκριμένο user_id
- **addUserMusicService**(Integer user_id, ServiceTokenPair token_pair):Void - Αποθηκεύει ένα ServiceTokenPair για τον χρήστη με το συγκεκριμένο user_id

1.1.1.5 Ορισμός Κλάσης UserDBProxy



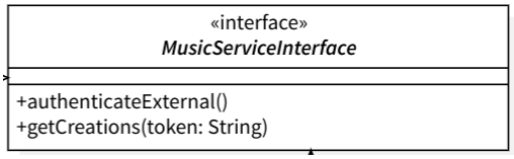
Η κλάση UserDBProxy αποτελεί απλώς ένα realization του UserDBInterface. Μεσολαβεί μεταξύ του UserController και του UserDB.

1.1.1.6 Ορισμός Κλάσης UserDB



Ομοίως με την κλάση UserDBProxy, η κλάση UserDB αποτελεί ένα realization του UserDBInterface. Είναι η κλάση που επικοινωνεί στην πραγματικότητα με την βάση δεδομένων. Οι μέθοδοί της καλούνται μόνο από το UserDBProxy.

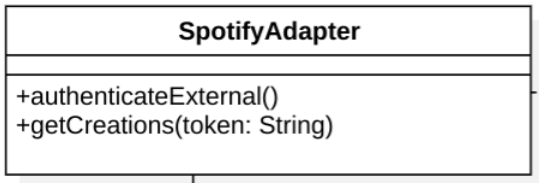
1.1.1.7 Ορισμός Κλάσης MusicServiceInterface



Μέθοδοι Κλάσης:

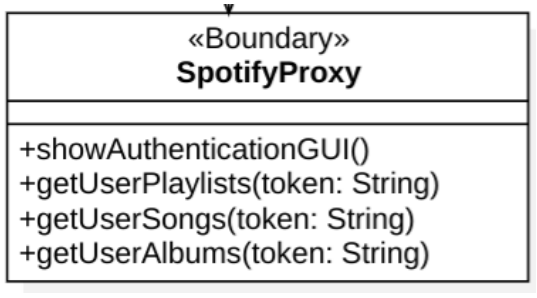
- **authenticateExternal**():Void - Εκκινεί την διαδικασία ταυτοποίησης μέσω του εξωτερικού συστήματος του music service
- **getCreations**(String token):Creation[] - Επιστρέφει όλα τα creations του χρήστη που υπάρχουν στο music service

1.1.1.8 Ορισμός Κλάσης SpotifyAdapter



Η κλάση SpotifyAdapter αποτελεί απλώς ένα realization του MusicServiceInterface για το Spotify.

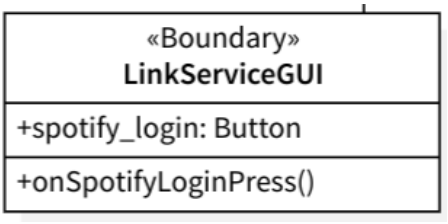
1.1.1.9 Ορισμός Κλάσης SpotifyProxy



Μέθοδοι Κλάσης:

- **showAuthenticationGUI**():Void - Εμφανίζει το login page του Spotify
- **getUserPlaylists**(String token):Playlist[] - Επιστρέφει όλα τα playlist που έχει δημιουργήσει ο χρήστης στο Spotify
- **getUserSongs**(String token):Song[] - Επιστρέφει όλα τα songs που έχει δημιουργήσει ο χρήστης στο Spotify
- **getUserAlbums**(String token):Album[] - Επιστρέφει όλα τα albums που έχει δημιουργήσει ο χρήστης στο Spotify

1.1.1.10 Ορισμός Κλάσης LinkServiceGUI



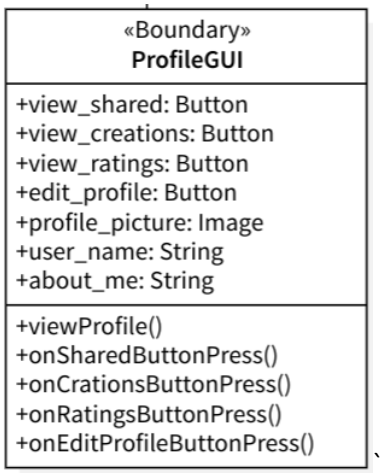
Χαρακτηριστικά Κλάσης:

- **spotify_login:**Button - Το κουμπί για την σύνδεση μέσω Spotify

Μέθοδοι Κλάσης:

- **onSpotifyLoginPress**():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού spotify_login

1.1.1.12 Ορισμός Κλάσης ProfileGUI



Χαρακτηριστικά Κλάσης:

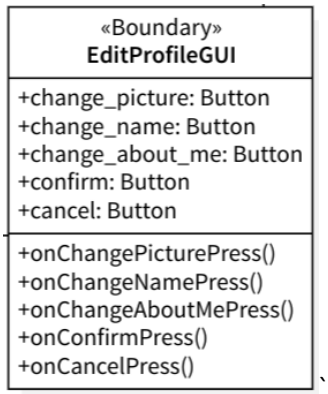
- **view_shared:**Button - Το κουμπί που εμφανίζει τα share του χρήστη
- **view_creations:**Button - Το κουμπί που εμφανίζει τα creation του χρήστη
- **view_ratings:**Button - Το κουμπί που εμφανίζει τα rating του χρήστη
- **edit_profile:**Button - Το κουμπί που ενεργοποιεί την επεξεργασία του προφίλ (διαθέσιμο μόνο όταν ο τρέχον χρήστης είναι ο κάτοχος του προβαλλόμενου προφίλ)
- **profile_picture:**Image- Η εικόνα προφίλ του χρήστη
- **user_name:**String- Το όνομα του χρήστη
- **about_me:**String- Η περιγραφή του προφίλ του χρήστη

Μέθοδοι Κλάσης:

- **viewProfile**():Void - Εμφανίζει το προφίλ
- **onSharedButtonPress**():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού view_shared
- **onCreationsButtonPress** ():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού view_creations
- **onRatingsdButtonPress** ():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού view_ratings
- **onEditProfileButtonPress** ():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού edit_profile

1.1.1.13 Ορισμός Κλάσης EditProfileGUI

Η κλάση EditProfileGUI κληρονομεί από την κλάση ProfileGUI.



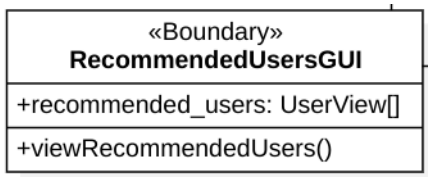
Χαρακτηριστικά Κλάσης:

- **change_picture:**Button - Το κουμπί που αλλάζει την εικόνα προφίλ του χρήστη
- **change_name:**Button - Το κουμπί που αλλάζει το όνομα του χρήστη
- **change_about_me:**Button - Το κουμπί που αλλάζει την περιγραφή του προφίλ του χρήστη
- **confirm:**Button - Το κουμπί που οριστικοποιεί τις τρέχουσες αλλαγές
- **cancel:**Button- Το κουμπί που ακυρώνει τις τρέχουσες αλλαγές

Μέθοδοι Κλάσης:

- **onChangePicuterPress**():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού change_picture
- **onChangeNamePress():**Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού change_name
- **onChangeAboutMePress**():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού change_about_me
- **onConfirmPress():**Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού confirm
- **onCancelPress():**Void - Η ενέργεια που θα εκτελεστεί με το πάτημα του κουμπιού cancel

1.1.1.14 Ορισμός Κλάσης RecommendedUsersGUI



Χαρακτηριστικά Κλάσης:

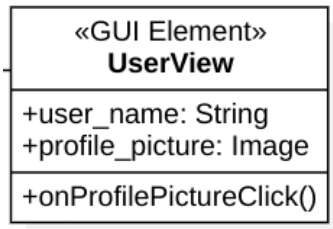
- **recommended_users:**UserView[] - Η λίστα με τα GUI Elements που αντιπροσωπεύουν τους προτεινόμενους χρήστες

Μέθοδοι Κλάσης:

- **viewRecommendedUsers**():Void - Προβάλει τους προτεινόμενους χρήστες

1.1.1.15 Ορισμός Κλάσης UserView

Το UserView είναι ένα GUI Element που χρησιμοποιείται για την συνοπτική προβολή ενός χρήστη. Χρησιμοποιείται όταν πρέπει να εμφανιστούν πολλαπλοί χρήστες στην ίδια σελίδα.



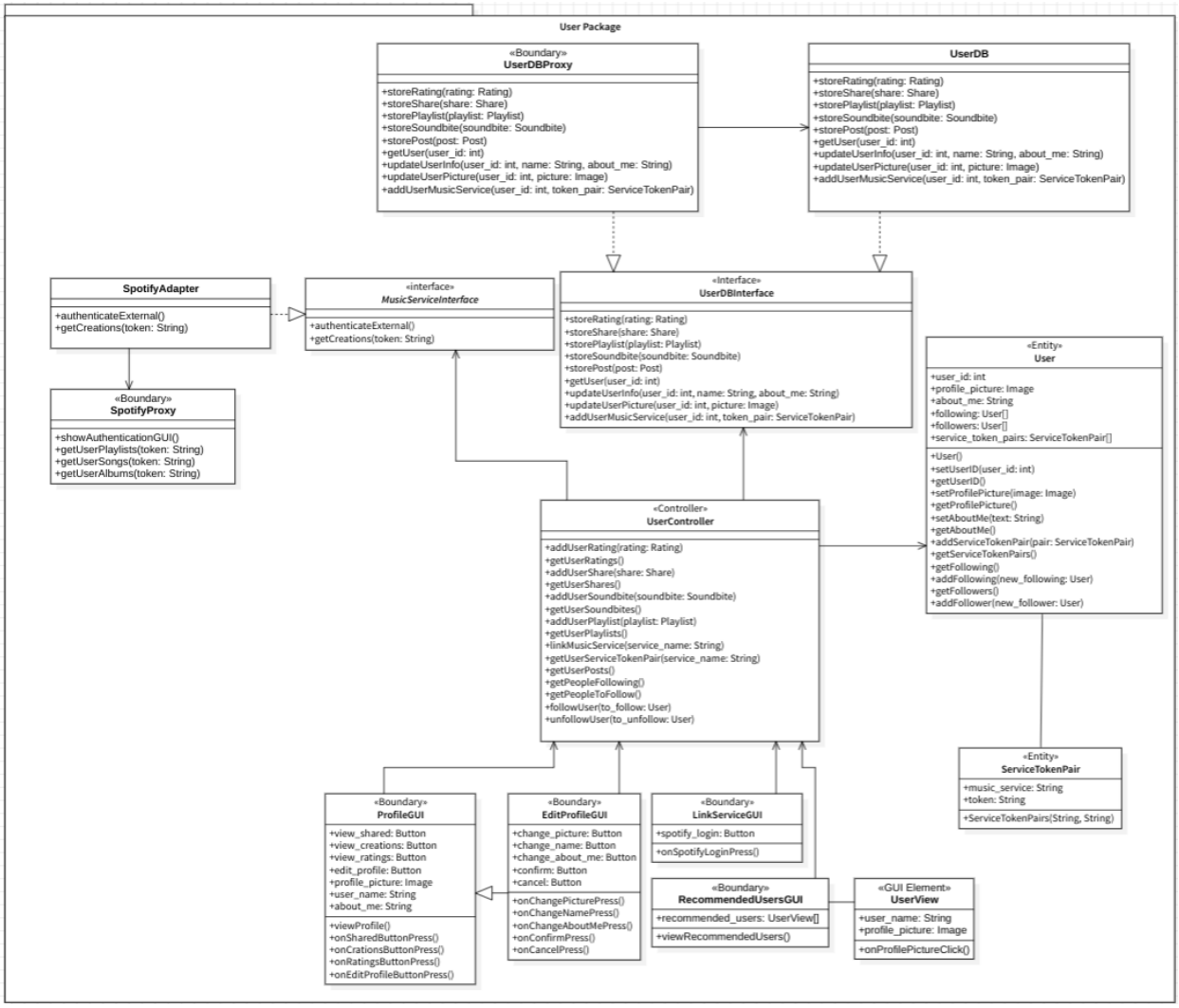
Χαρακτηριστικά Κλάσης:

- **user_name:**String - Το όνομα του χρήστη
- **profile_picture**:Image- Η εικόνα προφίλ του χρήστη

Μέθοδοι Κλάσης:

- **onProfilePictureClick**():Void - Η ενέργεια που θα εκτελεστεί με το πάτημα τις εικόνας προφίλ (προβολή του προφίλ)

1.1.2 Διάγραμμα κλάσεων

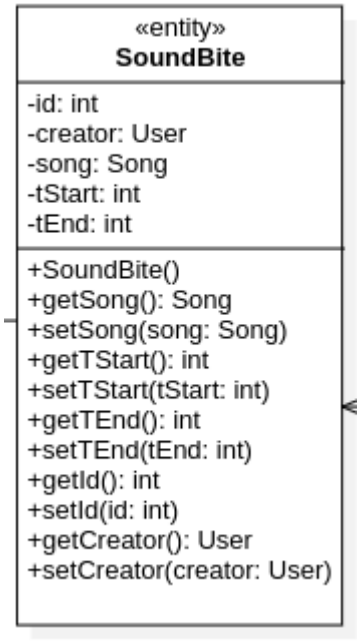


Εικόνα 2: Διάγραμμα κλάσεων για το package User

1.2 <Πακέτο κλάσεων Create Event>

1.2.1 Ορισμός κλάσεων

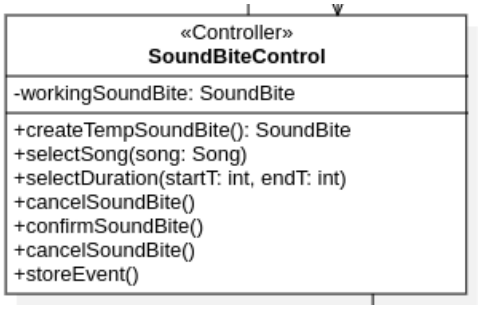
1.2.1.1 Ορισμός κλάσης SoundBite



Χαρακτηριστικά:

- id: int - το αναγνωριστικό id του SoundBite event από το DataBase
- creator: User – το αντικείμενο τύπου User που αντιστοιχεί στον χρήστη που δημιούργησε το SoundBite
- song: Song – το τραγούδι που χρησιμοποιείται στο SoundBite
- tStart: int – η αρχή του clip από το αρχικό τραγούδι που χρησιμοποιείται (σε seconds)
- tEnd: int – το τελος του clip από το αρχικό τραγούδι που χρησιμοποιείται (σε seconds)

1.2.1.2 Ορισμός κλάσης SoundBiteControl



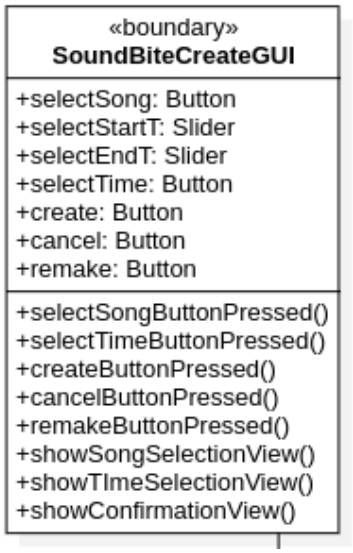
Χαρακτηριστικά:

- workingSoundBite: SoundBite - το αντικείμενο SoundBite στο οποίο έχει αντιστοιχηθέι ο controller

Μέθοδοι:

- CreateTempSoundBite(): SoundBite - δημιουργεί ένα νέο αντικείμενο SoundBite και το συνδέει με το workingSoundBite
- SelectSong(song: Song): void - τροποποιεί το song attribute του workingSoundBite
- SelectDuration(startT: int, endT: int): void - τροποποιεί τα χρονικά όρια του workingSoundBite
- ConfirmSoundBite(): void - επιβεβαιώνει το workingSoundBite ως finished object και το αποθηκεύει μόνιμα στο DataBase μέσω του storeEvent()
- CancelSoundBite(): void - ακυρώνει την διαδικασία δημιουργίας και καταστρέφει το workingSoundBite
- StoreEvent(): void - κοινή μέθοδος όλων των event controllers που αποθηκεύει το αντίστοιχο event στο DataBase.

1.2.1.3 Ορισμός κλάσης SoundBiteCreateGUI



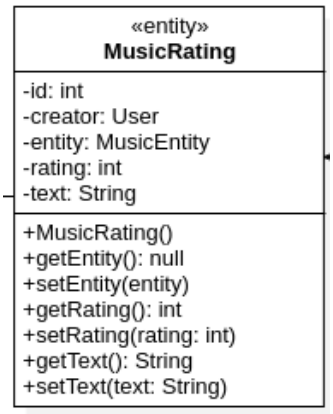
Χαρακτηριστικά:

- SelectSong: Button - κουμπί στο initial view που επιτρέπει τον χρήστη να επιλέξει το τραγούδι του SoundBite
- SelectStartT(EndT): Slider – slider στο time selection view που επιλέγει με ακρίβεια δευτερολέπτου την αρχή (το τέλος) του clip που θα χρησιμοποιηθεί στο SoundBIte
- SelectTime: Button - κουμπί στο time selection view που επιβεβαιώνει την επιλογή των χρόνων με βάση τις τιμές των sliders
- Create: Button - κουμπί στο confirmation view που επιτρέπει τον χρήστη να επιβεβαιώσει την δημιουργία του SoundBIte
- Cancel: Button - κουμπί στο confirmation view που επιτρέπει τον χρήστη να ακυρώσει το SoundBite και να επιστρέψει στο feed
- Remake: Button - κουμπί στο confirmation view που επιτρέπει τον χρήστη να επαναλάβει από την αρχή την διαδικασία δημιουργίας SoundBite

Μέθοδοι:

- SelectSongButtonPressed(): void - ο χρήστης μεταβαίνει σε ένα song selection GUI το οποίο του επιτρέπει να αναζητήσει και να επιλέξει κάποιο τραγούδι από το DataBase και έπειτα να τον μεταφέρει στο time selection view
- SelectTimeButtonPressed(): void - επιλέγεται το τωρινό χρονικό διάστημα των sliders και αν είναι valid ο χρήστης μεταβαίνει στο confirmation view
- CreateButtonPressed(): void - εκτελούνται οι λειτουργίες του createButton από παραπάνω.
- CancelButtonPressed(): void - εκτελούνται οι λειτουργίες του cancelButton από παραπάνω.
- RemakeButtonPressed(): void - εκτελούνται οι λειτουργίες του remakeButton από παραπάνω
- ShowSongSelectionView(): void - μετατρέπει το GUI στο song selection (initial) view
- ShowTimeSelectionView(): void - μετατρέπει το GUI στο time selection view
- ShowConfirmationView(): void - μετατρέπει το GUi στο confirmation view

1.2.1.4 Ορισμός κλάσης MusicRating



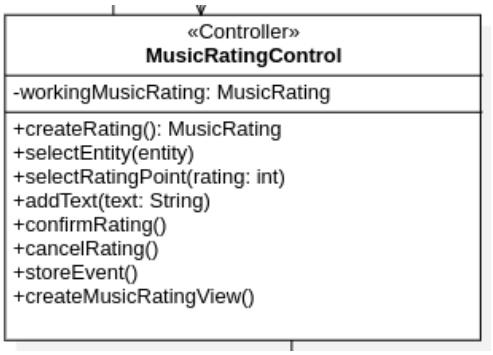
Χαρακτηριστικά:

- Id: int - μοναδικό αναγνωριστικό του αντικειμένου MusicRating event από το DataBase
- Creator: User - αντικείμενο τύπου User που αντιστοιχεί στον χρήστη που δημιούργησε το MusicRating
- Entity: MusicEntity - το αντικείμενο τύπου MusicEntity που βαθμολογείται
- Rating: int - αριθμός από ένα εώς δέκα που αντιπροσωπεύει τη βαθμολογία του MusicEntity
- Text: String - κείμενο που συνοδεύεται με το rating

1.2.1.5 Ορισμός κλάσης MusicRatingControl

Χαρακτηριστικά:

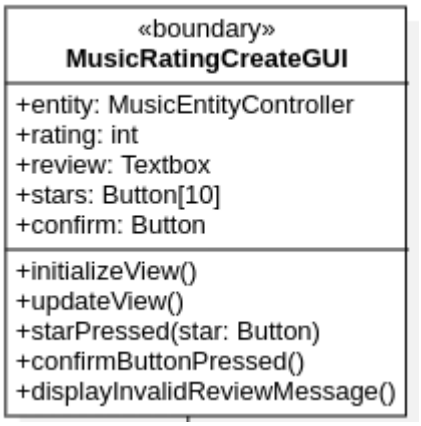
- WorkingMusicRating: MusicRating - το αντικείμενο το οποίο διαχειρίζεται την τωρινή στιγμή ο controller



Μέθοδοι:

- CreateRating(): MusicRating - δημιουργεί ένα νέο αντικείμενο MusicRating το οποίο επιστρέφει και συνδέει με το workingMusicRating
- SelectEntity(entity: MusicEntity): void - επιλέγει το entity του workingMusicRating
- SelectRatingPoint(rating: int): void - επιλέγει το rating του workingMusicRating
- AddText(text: String): void - προσθέτει το review text στο workingMusicRating
- ConfirmRating(): void - επιβεβαιώνει το workingMusicRating ως τελειωμένο και το αποθηκεύει το DataBase μέσω του storeEvent()
- CancelRating(): void - ακυρώνει τη διαδικασία του MusicRating και καταστρέφει το workingMusicRating
- StoreEvent(): void - κοινή μέθοδος όλων των event controllers που αποθηκεύει το αντίστοιχο event στο DataBase.
- CreateMusicRatingView(): void - μεταβαίνει στο MusicRatingView του workingMusicRating

1.2.1.6 Ορισμός κλάσης MusicRatingCreateGUI



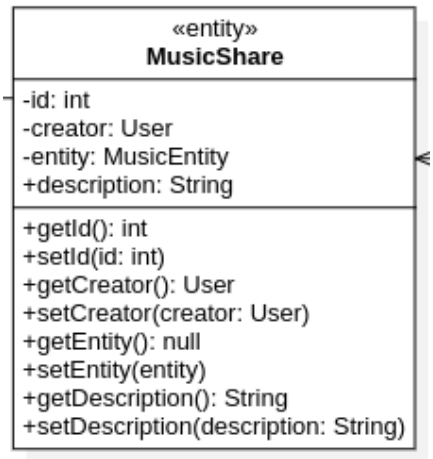
Χαρακτηριστικά:

- Entity: MusicEntityController - ο controller που είναι υπεύθυνος για την παροχή του musicEntityView στο GUI
- Rating: int - αριθμός της βαθμολογίας του MusicEntity που προβάλλεται με αστέρια
- Review: Textbox - ένα textbox στο οποίο ο χρήστης προσθέτει το review
- Stars: Button[10] - 10 κουμπία ένα για κάθε αστέρι τα οποία θέτουν την επιθυμητή βαθμολογία του χρήστη
- Confirm: Button - κουμπί που επιβεβαιώνει και αποθηκεύει το musicRating

Μέθοδοι:

- InitializeView(): void - προβάλλει το GUI στον χρήστη
- UpdateView(): void - ενημερώνει τα χαρακτηριστικά της οθόνης με βάση τις επιλογές του χρήστη
- StarPressed(star: Button): void - ενημερώνει το rating attribute και το GUI
- ConfirmButtonPressed(): void - επιβεβαιώνει την δημιουργία του review και καλεί την αντίστοιχη συνάρτηση του controller
- DisplayInvalidReviewMessage(): void - προβάλλει μήνυμα σφάλματος στον χρήση για τυχόν λάθος στο review

1.2.1.7 Ορισμός κλάσης MusicShare



Χαρακτηριστικά:

- Id: int - μοναδικό αναγνωριστικό του αντικειμένου MusicShare event από το DataBase
- Creator: User - αντικείμενο τύπου User που αντιστοιχεί στον χρήστη που δημιούργησε το MusicShare
- Entity: MusicEntity - το αντικείμενο τύπου MusicEntity που μοιράζεται
- Description: String - κείμενο που συνοδεύει το share του event

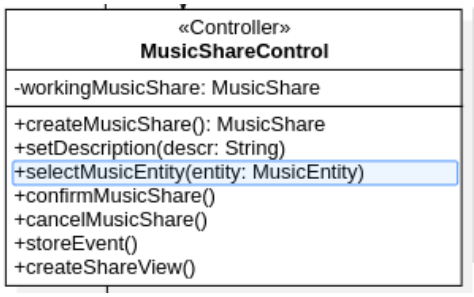
1.2.1.8 Ορισμός κλάσης MusicShareControl

Χαρακτηριστικά:

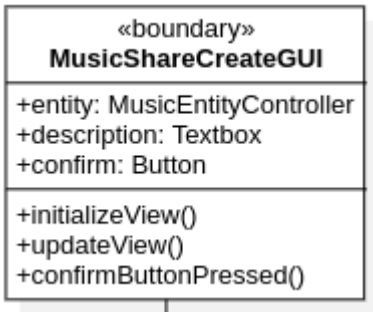
- WorkingMusicShare: MusicShare - αντικείμενο τύπου MusicShare που διαχειρίζεται ο controller

Μέθοδοι:

- CreateMusicShare(): MusicShare - δημιουργεί και επιστρέφει ένα αντικείμενο MusicShare και το συνδέει με το workingMusicShare
- SetDescrption(descr: String): void - τροποποιεί το description του workingMusicShare
- SelectMusicEntity(entity: MusicEntity): void - τροποποιεί το musicEntitiy του workingMusicShare
- ConfirmMusicShare(): void - επιβεβαιώνει την δημιουργία του MusicShare και το αποθηκεύει στο dataBase
- CancelMusicShare(): void - ακυρώνει το MusicShare και καταστρέφει το workingMusicShare
- StoreEvent(): void - κοινή μέθοδος όλων των event controllers που αποθηκεύει το αντίστοιχο event στο DataBase.
- CreateShareView(): void - προβάλλει στον χρήστη το MusicShareView του event



1.2.1.9 Ορισμός κλάσης MusicShareCreateGUI



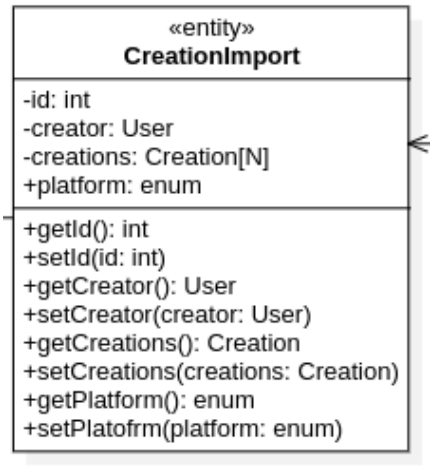
Χαρακτηριστικά:

- Entity: MusicEntityController – ο controller που συνδεεται με το musicEntity που δημοσιεύεται και προωθεί το GUI του entityView στην κλαση
- Description: Textbox - το textbox στο οποίο ο χρήστης εισάγει το description
- Confirm: Button - κουμπί που επιβεβαιώνει την δημιουργία του share και επιστρέφει τον χρήστη στο feed

Μέθοδοι:

- InitializeView(): void - αρχικοποιεί το GUI του musicShare
- UpdateView(): ενημερώνει την οθόνη με τις νέες αλλαγές στα attributes
- ConfirmButtonPressed(): void - επιβεβαιώνει την δημιουργία του event καλώντας την συνάρτηση του αντίστοιχου controller
-

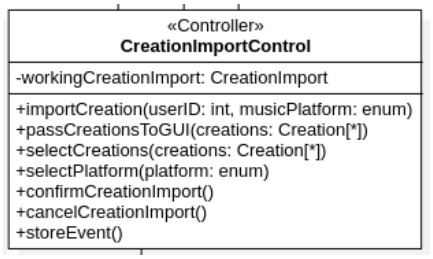
1.2.1.10 Ορισμός κλάσης CreationImport



Χαρακτηριστικά:

- Id: int - μοναδικό αναγνωριστικό του αντικειμένου CreationImport από το Data Base
- Creator: User - αντικείμενο τύπου User που αντιστοιχεί στον δημιουργό του event CreationImport
- Creations: Creation[N] - λίστα από τα creations που εισήχθησαν από external Music Services
- Platform: enum - το συγκεκριμένο Music Service από το οποίο εισήχθησαν τα creations

1.2.1.11 Ορισμός κλάσης CreationImportControl



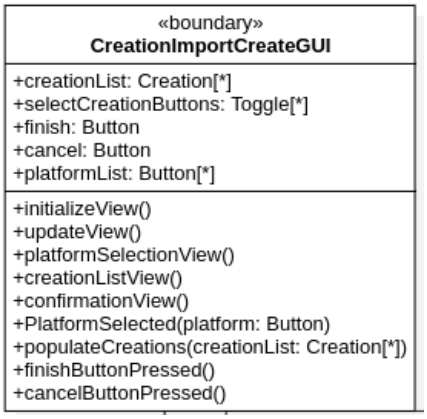
Χαρακτηριστικά:

- WorkingCreationImport: CreationImport - αντικείμενο τύπου CreationImport που διαχειρίζεται ο controller

Μέθοδοι:

- ImportCreation(userID: int, musicPlatform: enum): void - εισάγει τον χρήστη σε ένα περιβάλλον ανάλογα με το musicPlatform στο οποίο θα επιλέξει τα creations που θέλει να εισάγει
- PassCreationsToGUI(creations: Creation[*]): void - μεταφέρει την απαραίτητη πληροφορία των creations στο GUI για την προβολή τους ώστε ο χρήστης να επιλέξει κάποια από αυτά
- SelectCreations(creations: Creation[*]): void - τροποποιεί το creations attribute του workingCreationImport
- SelectPlatform(platform: enum): void - τροποποιεί το platform attribute του workingCreationImport
- ConfirmCreationImport(): void - επικυρώνει το αντικείμενο workingCreationImport και το αποθηκεύει μόνιμα στο database
- CancelCreationImport(): void - ακυρώνει την διαδικασία της δημιουργίας του event και καταστρέφει το workingCreationImport
- StoreEvent(): void - κοινή μέθοδος όλων των event controllers που αποθηκεύει το αντίστοιχο event στο DataBase.

1.2.1.12 Ορισμός κλάσης CreationImportCreateGUI



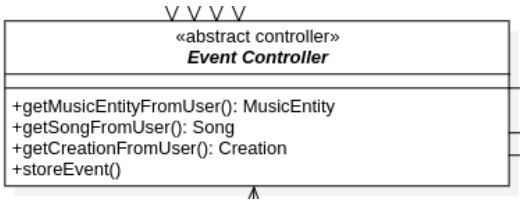
Χαρακτηριστικά:

- CreationList: Creation[*] - λίστα από Creation αντικείμενα τα οποία είναι προς επιλογή από τον χρήστη για εισαγωγή
- SelectCreationButtons: Toggle[*] - λίστα από toggles, ένα για κάθε creation ώστε ο χρήστης να μπορεί να επιλέξει πολλαπλά creations ταυτόχρονα
- Finish: Button - κουμπί το οποίο κάνει trigger το τέλος της διαδικασίας
- Cancel: Button - κουμπί που ακυρώνει τη διαδικασία
- PlatformList: Button[*] - λίστα από κουμπιά, ένα για κάθε διαθέσιμη πλατφόρμα music service

Μέθοδοι:

- InitializeView(): void - αρχικοποιεί το GUI
- UpdateView(): void - ενημερώνει το GUi με βάση τα νέα attributes
- PlatformSelectionView(): void - μεταφέρει το GUI στην οθόνη με την επιλογή πλατφόρμας
- CreationListView(): void - μεταφέρει το GUi στην οθόνη με την προβολή των διαθέσιμων creations
- ConfirmationView(): void - μεταφέρει το GUI στην οθόνη επιβεβαίωση του event
- PlatformSelected(platform: Button): void - μέσω του controller αλλάζει την πλατφόρμα του αντικειμένου και μεταβαίνει στην οθόνη επιλογής creations
- PopulateCreations(creationList: Creation[*]): void - με βάση το όρισμα αλλάζει το attribute της κλάσης και ενημερώνει το view
- FinishButtonSelected(): void - με βάση τα active toggles μεταφέρει την επιλογή στον controller και ολοκληρώνει την διαδικασία
- CancelButtonPressed(): void - ακυρώνεται η διαδικασία μέσω του controller

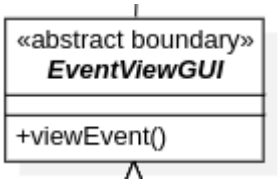
1.2.1.13 Ορισμός κλάσης EventControl



Μέθοδοι:

- GetMusicEntityFromUser(): MusicEntity - συνάρτηση που ανακατευθύνει τον χρήστη σε μια interactive διαδικασία στην οποία επικοινωνεί με το database για να επιλέξει ένα MusicEntity το οποίο μετά επιστρέφεται
- GetSongFromUser(): Song - η ίδια διαδικασία με την παραπάνω μόνο που η αναζήτηση περιορίζεται σε τραγούδια
- GetCreationFromUser(): Creation - η ίδια διαδικασία με την παραπάνω μόνο που η αναζήτηση περιορίζεται σε creations
- StoreEvent(): void – virtual συνάρτηση η οποία υλοποιείται σε κάθε concrete controller στην οποία το event που έχει δημιουργηθεί αποθηκεύεται στο database

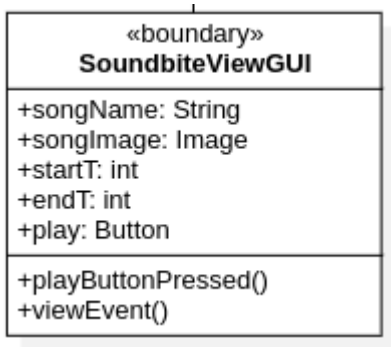
1.2.1.14 Ορισμός κλάσης EventViewGUI



Μέθοδοι:

- ViewEvent(): void – virtual συνάρτηση που υλοποιείται σε κάθε concrete controller στην οποία το αποτέλεσμα του event προβάλλεται στο GUI

1.2.1.15 Ορισμός κλάσης SoundBiteViewGUI



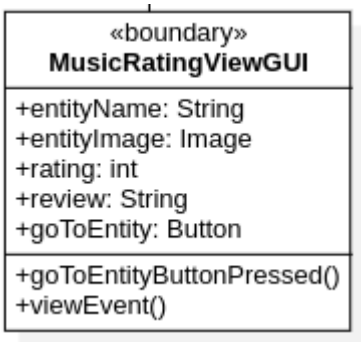
Χαρακτηριστικά:

- SongName: String - το όνομα του τραγουδιού του SoundBite
- SongImage: Image - η εικόνα του τραγουδιού (album cover)
- StartT: int - η αρχή του clip
- EndT: int - το τέλος του clip
- Play: Button - κουμπί που αναπαράγει το SoundBite

Μέθοδοι:

- PlayButtonPressed(): αναπαράγεται το SoundBite
- ViewEvent(): έχει οριστεί στο EventViewGUI

1.2.1.16 Ορισμός κλάσης MusicRatingViewGUI



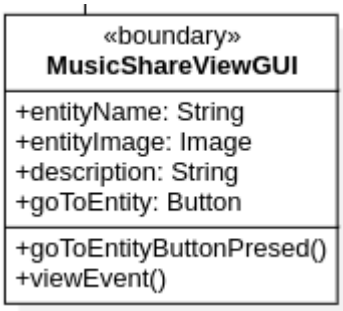
Χαρακτηριστικά:

- EntityName: String - το όνομα του MusicEntity που βαθμολογείται
- EntityImage: Image - η εικόνα που συνοδεύεται με το MusicEntity
- Rating: int - η βαθμολογία του MusicEntity
- Review: String - το review text
- GoToEntity: Button - κουμπί που μεταβαίνει στο entity view του MusicEntity

Μέθοδοι:

- GoToEntityButtonPressed(): void - ο χρήστης μεταβαίνει στο entity view του MusicEntity
- ViewEvent(): έχει οριστεί στο EventViewGUI

1.2.1.17 Ορισμός κλάσης MusicShareViewGUI



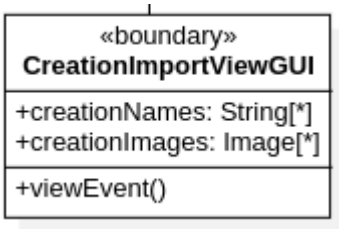
Χαρακτηριστικά:

- EntityName: String - το όνομα του MusicEntity που μοιράστηκε
- EntityImage: Image - η εικόνα που συνοδεύεται με το entity
- Description: String - η περιγραφή του share
- GoToEntityButton: Button - κουμπί που μεταβαίνει στο entity view του MusicEntity

Μέθοδοι:

- GoToEntityButtonPressed(): void - ο χρήστης μεταβαίνει στο entity view του MusicEntity
- ViewEvent(): void - έχει οριστεί στο EventViewGUI

1.2.1.18 Ορισμός κλάσης CreationImportViewGUI



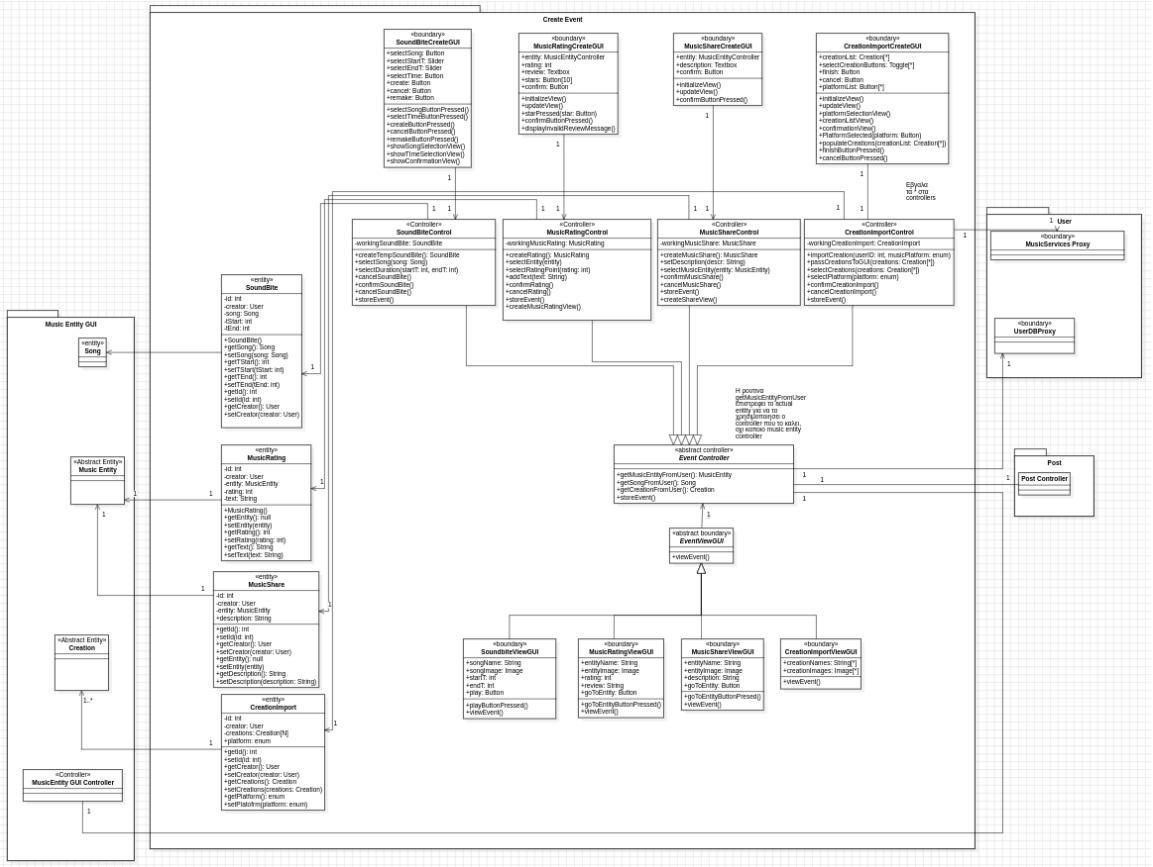
Χαρακτηριστικά:

- CreationNames: String[*] - λίστα από τα ονόματα των creations που εισήχθησαν
- CreationImages: Image[*] - λίστα από τις εικόνες των παραπάνω creations

Μέθοδοι:

- ViewEvent(): void - έχει οριστεί στο EventViewGUI

1.2.2 Διάγραμμα κλάσεων

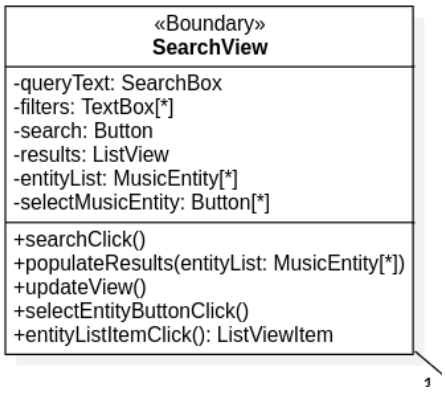


Εικόνα 2: Διάγραμμα κλάσεων για το package Create Event

1.3 <Πακέτο κλάσεων Music Entity>

1.3.1 Ορισμός κλάσεων

1.4.1.1 Ορισμός κλάσης SearchView



Χαρακτηριστικά:

- QueryText: SearchBox - Γραφική περιοχή εισόδου στην οποία ο χρήστης εισάγει τους όρους αναζήτησής του
- Filters: TextBox[] - Γραφικές περιοχές εισόδου στις οποίες ο χρήστης μπορεί να εισάγει παραπάνω φίλτρα για την αναζήτησή του
- Search: Button - Κουμπί το οποίο ξεκινά την διαδικασία αναζήτησης
- Results: ListView - Λίστα γραφικών στοιχείων των αποτελεσμάτων που δημιουργούνται μετά το πέρας της αναζήτησης
- EntityList: MusicEntity[] - Διάνυσμα στοιχείων τύπου MusicEntity
- SelectMusicEntity: Button[] - Διάνυσμα κουμπιών με τα οποία ο χρήστης επιλέγει τα αποτελέσματα που τον ενδιαφέρουν

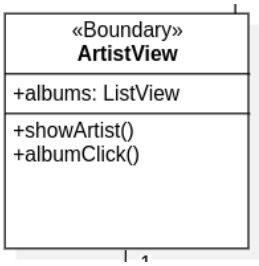
Μέθοδοι:

- SearchClick(): void - Συνάρτηση του κουμπιού *search* με το οποίο ξεκινά η διαδικασία αναζήτησης των αποτελεσμάτων
- PopulateResults(entityList: MusicEntity[]) : void - Ενημερώνει το GUI με το διάνυσμα αποτελεσμάτων που βρέθηκαν μέσω της παραμέτρου entityList
- UpdateView() : void - ενημερώνει το GUI για τυχόν άλλες αλλαγές
- SelectEntityButtonClick() : void - Συνάρτηση που καλείται όταν πατηθεί οποιοδήποτε από τα κουμπιά του διανύσματος *selectMusicEntity*
- EntityListItemClick() : ListViewItem - Συνάρτηση που καλείται όταν πατηθεί οποιοδήποτε από τα αποτελέσματα του χαρακτηριστικού *results*. Επιστρέφει το αποτέλεσμα που πατήθηκε που είναι τύπου ListViewItem

1.4.1.2 Ορισμός κλάσης ArtistView

Χαρακτηριστικά:

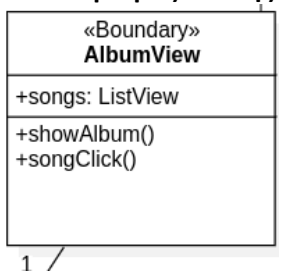
- Albums : ListView - Λίστα γραφικών στοιχείων των άλμπουμ του καλλιτέχνη



Μέθοδοι:

- ShowArtist() : void - Συνάρτηση η οποία παρουσιάζει τον καλλιτέχνη που επέλεξε ο χρήστης χρησιμοποιώντας το χαρακτηριστικό *album* και τα χαρακτηριστικά που κληρονομεί από την κλάση *MusicEntityView*
- AlbumClick() : void - Συνάρτηση η οποία καλείται όταν ο χρήστης πατήσει επάνω σε ένα άλμπουμ

1.4.1.3 Ορισμός κλάσης AlbumView



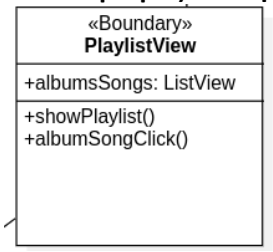
Χαρακτηριστικά:

- Songs: ListView - Λίστα γραφικών στοιχείων των τραγουδιών του άλμπυμ

Μέθοδοι:

- ShowAlbum(): void - Συνάρτηση η οποία παρουσιάζει το άλμπουμ που επέλεξε ο χρήστης χρησιμοποιώντας το χαρακτηριστικό songs και τα χαρακτηριστικά που κληρονομεί από την κλάση *MusicEntityView*
- AlbumClick(): void - Συνάρτηση η οποία καλείται όταν ο χρήστης πατήσει επάνω σε ένα τραγούδι

1.4.1.4 Ορισμός κλάσης PlaylistView



Χαρακτηριστικά:

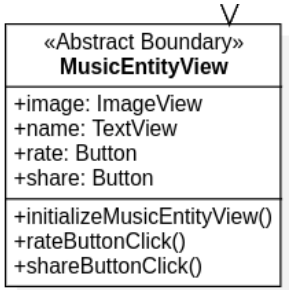
- AlbumsSongs: ListView - Λίστα γραφικών στοιχείων των άλμπουμ και των τραγουδιών του playlist

Μέθοδοι:

- ShowPlaylist(): void - Συνάρτηση η οποία παρουσιάζει το playlist που επέλεξε ο χρήστης χρησιμοποιώντας το χαρακτηριστικό *albumsSongs* και τα χαρακτηριστικά που κληρονομεί από την κλάση *MusicEntityView*
- albumSongClick: void - Συνάρτηση η οποία καλείται όταν ο χρήστης πατήσει επάνω σε ένα άλμπουμ ή ένα τραγούδι του playlist

1.4.1.5 Ορισμός κλάσης MusicEntityView

Σημείωση: Αυτή η κλάση είναι αφαιρετική (abstract class), που σημαίνει ότι διαθέτει μόνον μεθόδους και χαρακτηριστικά που δεν είναι αρχικοποιημένα, ή μπορεί είναι αρχικοποιημένα σε μια άκυρη τιμή. Αρχικοποιούνται από τις κλάσεις που θα κληρονομήσουν την υπάρχουσα.



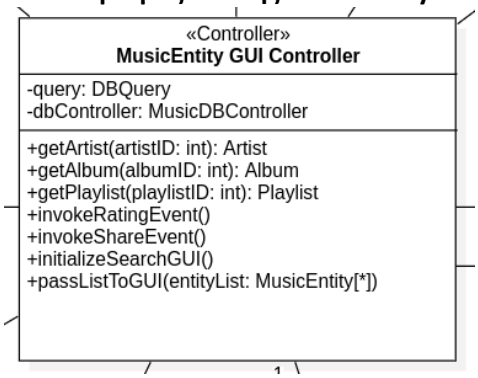
Χαρακτηριστικά:

- Image: ImageView - Γραφικό στοιχείο που εμφανίζει μια εικόνα μιας μουσικής οντότητας ανάλογα την κλάση που κληρονομεί την παρούσα κλάση
- Name: TextView - Εμφανίζει το όνομα της μουσικής οντότητας προς εμφάνιση
- rate: Button - Κουμπί το οποίο επιτρέπει στον χρήστη να βαθμολογήσει και να γράψει μια κριτική
- share: Button - Κουμπί το οποίο επιτρέπει στον χρήστη να διαμοιραστεί μια μουσική οντότητα

Μέθοδοι:

- InitializeMusicEntityView(): void - Αρχικοποιεί την GUI κλάση που κληρονομεί την παρούσα
- RateButtonClick(): void - Συνάρτηση που καλείται όταν πατηθεί το κουμπί *rate*. Εκκινεί την διαδικασία της βαθμολόγησης
- ShareButtonClick(): void - Συνάρτηση που καλείται όταν πατηθεί το κουμπί *share*. Εκκινεί την διαδικασία του διαμοιρασμού

1.4.1.6 Ορισμός κλάσης MusicEntity GUI Controller



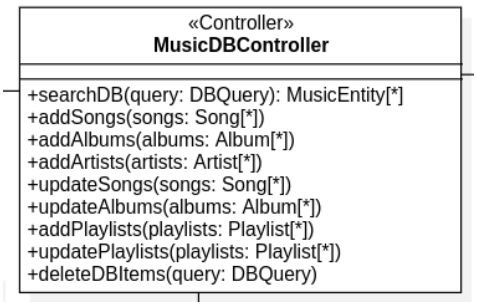
Χαρακτηριστικά:

- Query: DBQuery - Αντικείμενο τύπου DBQuery
- DbController: MusicDBController - Αντικείμενο τύπου MusicDBController

Μέθοδοι:

- GetArtist(artistID: int): Artist - Επιστρέφει ένα αντικείμενο τύπου Artist σύμφωνα με το ID του που περνιέται σαν παράμετρος
- GetAlbum(albumID: int): Album - Επιστρέφει ένα αντικείμενο τύπου Album σύμφωνα με το ID του που περνιέται σαν παράμετρος
- GetPlaylist(playlistID: int): Playlist - Επιστρέφει ένα αντικείμενο τύπου Playlist σύμφωνα με το ID του που περνιέται σαν παράμετρος
- InvokeRatingEvent(): void - Εκκινεί ένα γεγονός βαθμολόγησης μιας οντότητας
- InvokeShareEvent(): void - Εκκινεί ένα γεγονός διαμοιρασμού μιας οντότητας
- InitializeSearchGUI(): void - Αρχικοποιεί το GUI αναζήτησης που ανήκει στην κλάση *SearchView*
- PassListToGUI(entityList: MusicEntity[]): void - Μεταφέρει οντότητες στα αντίστοιχα GUI που τα χρειάζονται κάθε φορά. Έχει σαν παράμετρο ένα διάνυσμα μουσικών οντοτήτων

1.4.1.7 Ορισμός κλάσης MusicDBController

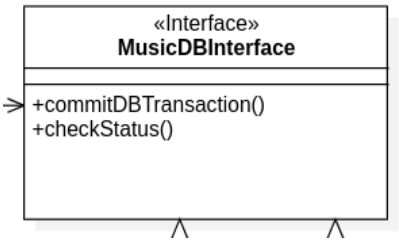


Μέθοδοι:

- SearchDB(query: DBQuery): MusicEntity[] - Αναζητά στην DB για μουσικές οντότητες που ικανοποιούν τα εκάστοτε φίλτρα που περιγράφονται με ένα αντικείμενο τύπου DBQuery, το οποίο και περνιέται σαν παράμετρος
- AddSongs(songs: Song[]): void - Προσθέτει τραγούδια στην DB μέσω ενός διανύσματος αντικειμένων τύπου Song που διαθέτει σαν παράμετρο
- AddAlbums(albums: Album[]): void - Προσθέτει άλμπουμ στην DB μέσω ενός διανύσματος αντικειμένων τύπου Album που διαθέτει σαν παράμετρο
- AddArtists(artists: Artists[]): void - Προσθέτει καλλιτέχνες στην DB μέσω ενός διανύσματος αντικειμένων τύπου Artist που διαθέτει σαν παράμετρο
- UpdateSongs(songs: Song[]): void - Ενημερώνει τα διάφορα στοιχεία των υπαρχόντων τραγουδιών στην DB με τα στοιχεία των αντικειμένων τύπου Song που δέχεται σαν παράμετρο
- UpdateAlbums(albums: Album[]): void - Ενημερώνει τα διάφορα στοιχεία των υπαρχόντων άλμπουμ στην DB με τα στοιχεία των αντικειμένων τύπου Album που δέχεται σαν παράμετρο
- AddPlaylists(playlists: Playlist[]): void - Προσθέτει καλλιτέχνες στην DB μέσω ενός διανύσματος αντικειμένων τύπου Playlist, τα οποία και δέχεται σαν παράμετρο
- UpdatePlaylists(playlists: Playlist[]): void - Ενημερώνει τα διάφορα στοιχεία των υπαρχόντων playlist στην DB με τα στοιχεία των αντικειμένων τύπου Playlist που δέχεται σαν παράμετρο
- DeleteDBItems(query: DBQuery): void - Διαγράφει δεδομένα από την DB με βάση των φίλτρων που περιγράφονται από ένα αντικείμενο τύπου DBQuery που το δέχεται σαν παράμετρο

1.4.1.8 Ορισμός κλάσης MusicDBInterface

Σημείωση: Αυτή η κλάση είναι μία διεπαφή (interface), που σημαίνει ότι διαθέτει μόνον μεθόδους οι οποίες δεν είναι αρχικοποιημένες. Αρχικοποιούνται από τις κλάσεις που θα χρησιμοποιούν την υπάρχουσα διεπαφή. Επομένως εδώ οι μέθοδοι περιγράφονται πολύ περιληπτικά

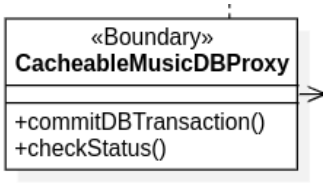


Μέθοδοι:

- CommitDBTransaction(): void - πραγματοποιεί τα αιτήματα της κλάσης *MusicDBController* μεταφέροντας τα δεδομένα και αποθηκεύοντάς τα με προσοχή στην DB
- CheckStatus(): void - ενημερώνει για την κατάσταση της DB ώστε να αποφευχθούν τυχόντα data races που ενδεχομένως να προκύψουν και να βοηθήσει στον συγχρονισμό των διαφόρων αιτημάτων.

1.4.1.9 Ορισμός κλάσης CacheableMusicDBProxy

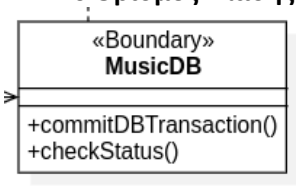
Σημείωση: Η παρούσα κλάση ικανοποιεί την ΜΛΑ-1.



Μέθοδοι:

- CommitDBTransaction(): void - πραγματοποιεί τα αιτήματα της κλάσης *MusicDBController* μεταφέροντας τα δεδομένα και αποθηκεύοντάς τα με προσοχή στην DB, χρησιμοποιώντας μια cached μεθοδολογία, ώστε να μην επιβαρύνεται η DB άσκοπα με αιτήματα που επιστρέφουν την ακριβώς ίδια απάντηση.
- CheckStatus(): void - ενημερώνει για την κατάσταση της DB ώστε να αποφευχθούν τυχόντα data races που ενδεχομένως να προκύψουν και να βοηθήσει στον συγχρονισμό των διαφόρων αιτημάτων.

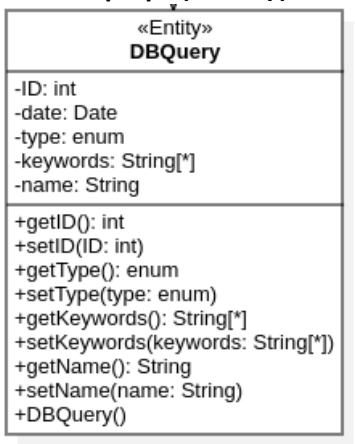
1.4.1.10 Ορισμός κλάσης MusicDB



Μέθοδοι:

- CommitDBTransaction(): void - πραγματοποιεί τα αιτήματα της κλάσης *MusicDBController* επικοινωνώντας απευθείας με την DB server, σε αρκετά χαμηλότερο επίπεδο.
- CheckStatus(): void - Ελέγχει την σύνδεση της DB με την εφαρμογή, έτσι ώστε να αντιμετωπιστούν εγκαίρως από την εφαρμογή τυχόν προβλήματα σύνδεσης και άλλων τεχνικών ζητημάτων που μπορεί να προκύψουν στον server

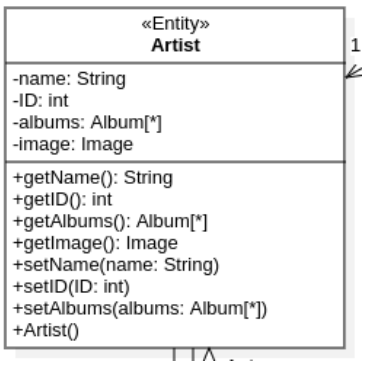
1.4.1.11 Ορισμός κλάσης DBQuery



Χαρακτηριστικά:

- ID: int - Ένας αριθμός με τον οποίο αναγνωρίζεται η μουσική οντότητα που ψάχνουμε
- Date: Date - Αντικείμενο τύπου Date για την ημερομηνία
- Type: enum - Enumerator για το είδος της μουσικής οντότητας που αναζητούμε
- keywords: String[] - Διάνυσμα αντικειμένων τύπου String για τις λέξεις-κλειδιά που μπορεί να βοηθήσουν ώστε να βρεθεί μια μουσική οντότητα
- name: String - Το όνομα της μουσικής οντότητας υπό αναζήτηση

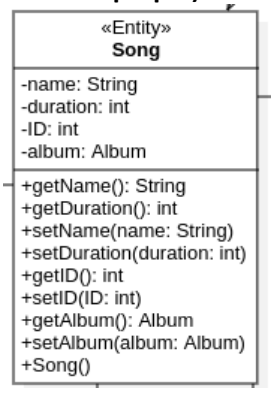
1.4.1.12 Ορισμός κλάσης Artist



Χαρακτηριστικά:

- Name: String - Το όνομα του καλλιτέχνη
- ID: int - Αριθμός με τον οποίο αναγνωρίζεται ο καλλιτέχνης
- Albums: Album[] - Διάνυσμα αντικειμένων τύπου Album. Αποτελούν τα άλμπουμ του καλλιτέχνη
- image: Image - Αντικείμενο τύπου Image, η φωτογραφία του καλλιτέχνη

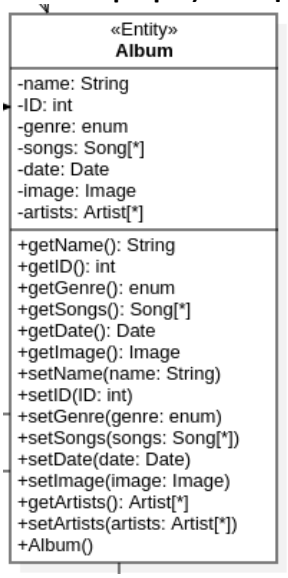
1.4.1.13 Ορισμός κλάσης Song



Χαρακτηριστικά:

- Name: String - Το όνομα του τραγουδιού
- Duration: int - Η διάρκεια του τραγουδιού σε δευτερόλεπτα
- ID: int - Αριθμός με τον οποίο αναγνωρίζεται το τραγούδι
- Album: Album - Αντικείμενο τύπου Album, αντιπροσωπεύει το άλμπουμ στο οποίο ανήκει

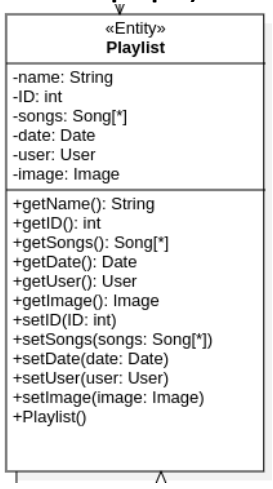
1.4.1.14 Ορισμός κλάσης Album



Χαρακτηριστικά:

- name: Το όνομα του άλμπουμ
- ID: Αριθμός με τον οποίο αναγνωρίζεται το άλμπουμ
- genre: Enumerator για το είδος μουσικής που περιέχει το άλμπουμ
- date: Η ημερομηνία κυκλοφορίας του άλμπουμ
- songs: Διάνυσμα αντικειμένων τύπου Song, τα τραγούδια που αποτελούν το άλμπουμ
- image: Η φωτογραφία του άλμπουμ
- artists: Διάνυσμα αντικειμένων τύπου Artist, είναι οι καλλιτέχνες στους οποίους ανήκει το άλμπουμ

1.4.1.15 Ορισμός κλάσης Playlist

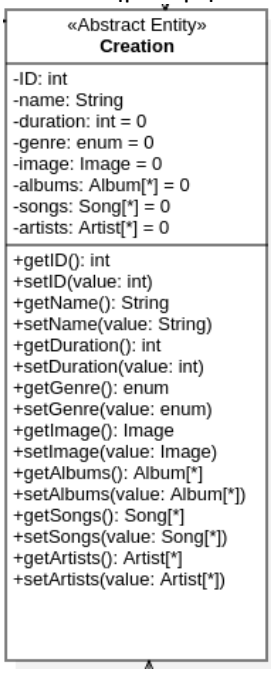


Χαρακτηριστικά:

- Name: String - Το όνομα του playlist
- ID: int - Αριθμός με τον οποίο αναγνωρίζεται ένα playlist
- Songs: Song[] - Διάνυσμα αντικειμένων τύπου Song, τα τραγούδια που περιέχονται στο playlist
- Date: Date - Ημερομηνία δημιουργίας του playlist
- user: User - Ο χρήστης στον οποίον ανήκει το playlist
- image: Image - Η φωτογραφία του playlist

1.4.1.16 Ορισμός κλάσης Creation

Σημείωση: Αυτή η κλάση είναι αφαιρετική (abstract class), που σημαίνει ότι διαθέτει μόνον μεθόδους και χαρακτηριστικά που δεν είναι αρχικοποιημένα, ή είναι αρχικοποιημένα σε μια άκυρη τιμή. Αρχικοποιούνται από τις κλάσεις που θα κληρονομήσουν την υπάρχουσα.

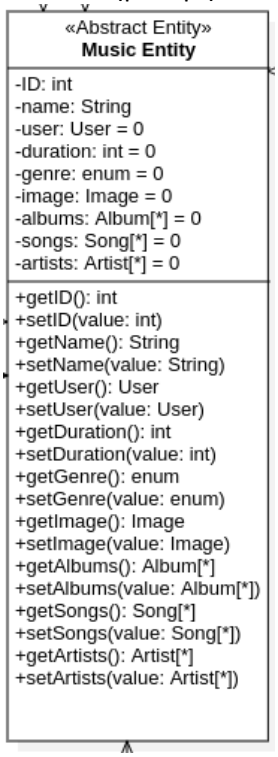


Χαρακτηριστικά:

- ID: int - Αριθμός με τον οποίο αναγνωρίζεται ένα αντικείμενο τύπου Creation
- Name: String - Το όνομα του αντικειμένου
- Duration: int - Η διάρκεια σε δευτερόλεπτα του αντικειμένου
- Genre: enum - Enumerator που δείχνει το είδος της μουσικής στο οποίο ανήκει το αντικείμενο
- Image: Image - Η φωτογραφία του αντικειμένου
- Albums: Album[] - Διάνυσμα αντικειμένων τύπου Album, είναι τα άλμπουμ που ανήκουν στο αντικείμενο
- songs: Song[] - Διάνυσμα αντικειμένων τύπου Song, είναι τα τραγούδια που ανήκουν στο αντικείμενο
- Artists: Artist[] Διάνυσμα αντικειμένων τύπου Artist, είναι τα τραγούδια που ανήκουν στο αντικείμενο

1.4.1.17 Ορισμός κλάσης Music Entity

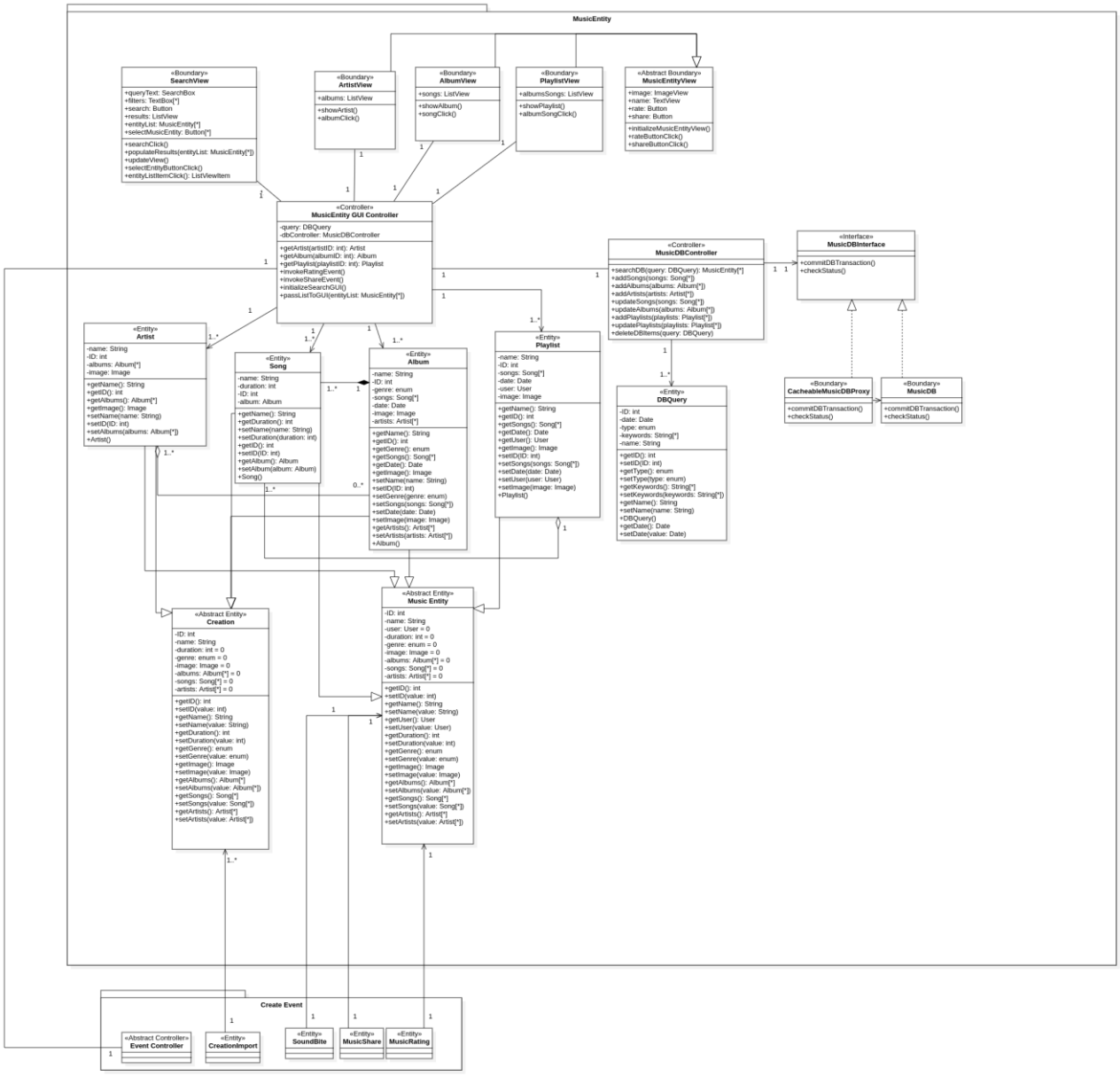
Σημείωση: Αυτή η κλάση είναι αφαιρετική (abstract class), που σημαίνει ότι διαθέτει μόνον μεθόδους και χαρακτηριστικά που δεν είναι αρχικοποιημένα, ή είναι αρχικοποιημένα σε μια άκυρη τιμή. Αρχικοποιούνται από τις κλάσεις που θα κληρονομήσουν την υπάρχουσα.



Χαρακτηριστικά:

- ID: int - Αριθμός με τον οποίο αναγνωρίζεται ένα αντικείμενο τύπου Creation
- Name: String - Το όνομα του αντικειμένου
- Duration: int - Η διάρκεια σε δευτερόλεπτα του αντικειμένου
- genre: Enumerator που δείχνει το είδος της μουσικής στο οποίο ανήκει το αντικείμενο
- image: Image - Η φωτογραφία του αντικειμένου
- Albums: Album[] - Διάνυσμα αντικειμένων τύπου Album, είναι τα άλμπουμ που ανήκουν στο αντικείμενο
- Songs: Song[] - Διάνυσμα αντικειμένων τύπου Song, είναι τα τραγούδια που ανήκουν στο αντικείμενο
- Artists: Artist[] - Διάνυσμα αντικειμένων τύπου Artist, είναι τα τραγούδια που ανήκουν στο αντικείμενο

1.3.2 Διάγραμμα κλάσεων



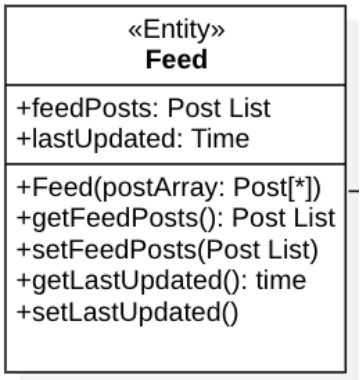
Εικόνα 2: Διάγραμμα κλάσεων για το package Music Entity

1.4 <Πακέτο κλάσεων Feed>

Χειρίζεται την ύπαρξη του Feed, ως του κεντρικού γραφικού περιβάλλοντος της εφαρμογής. Ουσιαστικά, το Feed αποτελεί μία σειρά από Post, δοθείσης ότι ο χρήστης έχει αρκετά μεγάλο κύκλο που ακολουθεί για την ύπαρξη αρκετών Posts, μέσα από τα οποία ο χρήστης μπορεί να “δει” τι ακούει ο κοινωνικός του περίγυρος.

1.4.1 Ορισμός κλάσεων

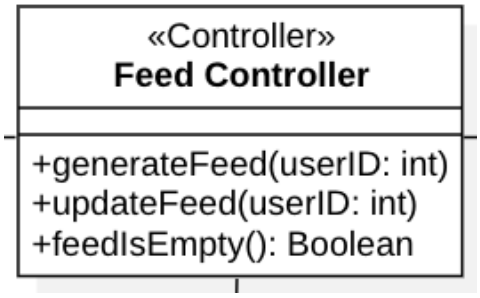
1.4.1.1 Ορισμός κλάσης Feed



Χαρακτηριστικά κλάσης:

- **feedPosts:** Post List - Η λίστα των διαθέσιμων post από χρήστες που ακολουθεί.
- **lastUpdated: Time** - Η τελευταία φορά στην οποία τα περιεχόμενα του feed ελέγχθηκαν. Απαραίτητο για την "ενημερωμένη" φύση του feed.

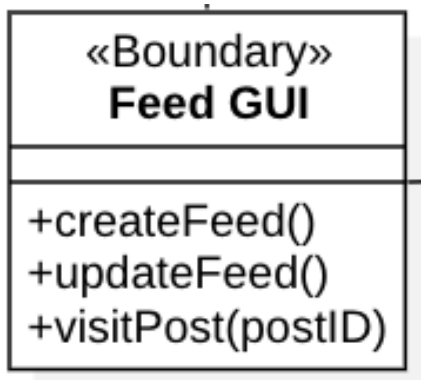
1.4.1.1 Ορισμός κλάσης Feed Controller



Μέθοδοι κλάσης:

- **GenerateFeed (userID: int):** - Λαμβάνοντας τις απαραίτητες πληροφορίες από τον UserController, εφόσον υπάρχουν Posts, δημιουργεί το Feed.
- **UpdateFeed (userID: int):** - Επικοινωνεί με τον UserController για την πιθανότητα ύπαρξης νέων Posts προς εισαγωγή στο Feed.
- **FeedIsEmpty (userID: int):** - Αν η λίστα feedPosts είναι άδεια τότε κάνει redirect στον UserController για την κλήση του ViewRecommended GUI

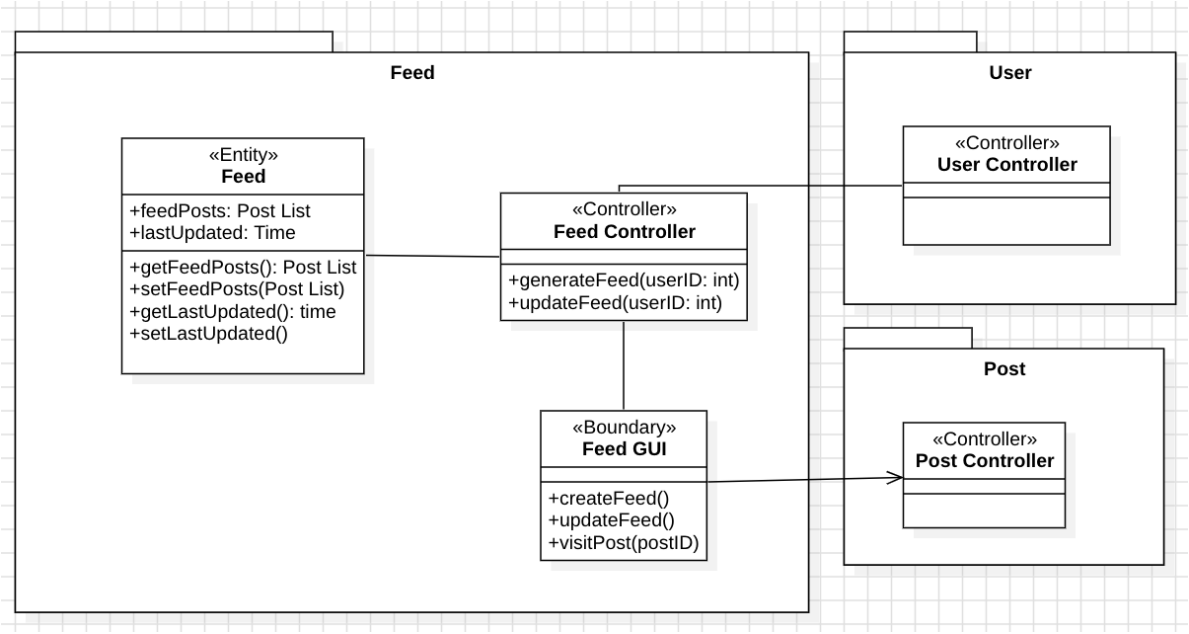
1.4.1.1 Ορισμός κλάσης Feed GUI



Μέθοδοι κλάσης:

- **CreateFeed():** - Κατά την εκκίνηση του συγκεκριμένου GUI αρχικοποιείται και προβάλλεται το Feed entity.
- **UpdateFeed():** - Αν ο χρήστης το επιλέξει μπορεί να ζητήσει χειροκίνητο έλεγχο για ενημερώσεις στο feed του
- **visitPost(postID: int):** - Αν ο χρήστης επιλέξει κάποιο Post τότε θα γίνει redirect στην σελίδα του αντίστοιχου

1.4.2 Διάγραμμα κλάσεων



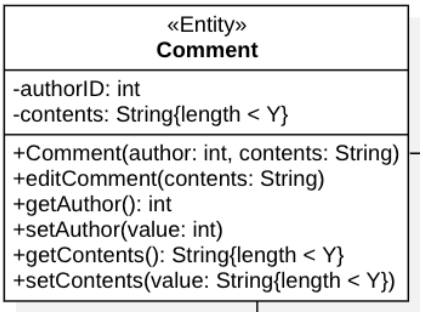
Εικόνα 2: Διάγραμμα κλάσεων για το package Feed

1.5 <Πακέτο κλάσεων Posts>

Χειρίζεται την ύπαρξη των Posts, ως δομικών στοιχείων εντός του Feed

1.5.1 Ορισμός κλάσεων

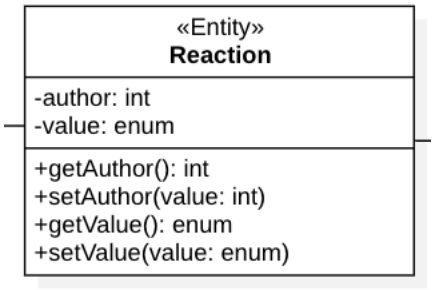
1.5.1.1 Ορισμός κλάσης Comment



Μέθοδοι κλάσης:

- **CreateFeed():** - Κατά την εκκίνηση του συγκεκριμένου GUI αρχικοποιείται και προβάλλεται το Feed entity.
- **UpdateFeed():** - Αν ο χρήστης το επιλέξει μπορεί να ζητήσει χειροκίνητο έλεγχο για ενημερώσεις στο feed του
- **visitPost(postID: int):** - Αν ο χρήστης επιλέξει κάποιο Post τότε θα γίνει redirect στην σελίδα του αντίστοιχου

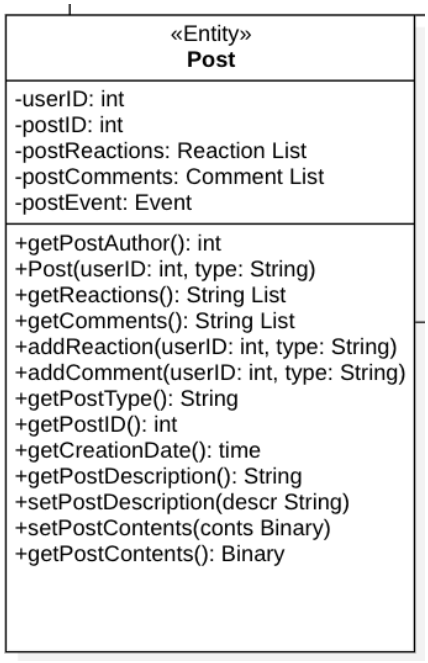
1.5.1.1 Ορισμός κλάσης Reaction



Μέθοδοι κλάσης:

- **CreateFeed():** - Κατά την εκκίνηση του συγκεκριμένου GUI αρχικοποιείται και προβάλλεται το Feed entity.
- **UpdateFeed():** - Αν ο χρήστης το επιλέξει μπορεί να ζητήσει χειροκίνητο έλεγχο για ενημερώσεις στο feed του
- **visitPost(postID: int):** - Αν ο χρήστης επιλέξει κάποιο Post τότε θα γίνει redirect στην σελίδα του αντίστοιχου

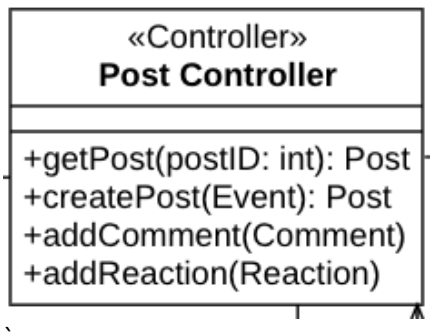
1.5.1.1 Ορισμός κλάσης Post



Μέθοδοι κλάσης:

- **CreateFeed():** - Κατά την εκκίνηση του συγκεκριμένου GUI αρχικοποιείται και προβάλλεται το Feed entity.
- **UpdateFeed():** - Αν ο χρήστης το επιλέξει μπορεί να ζητήσει χειροκίνητο έλεγχο για ενημερώσεις στο feed του
- **visitPost(postID: int):** - Αν ο χρήστης επιλέξει κάποιο Post τότε θα γίνει redirect στην σελίδα του αντίστοιχου

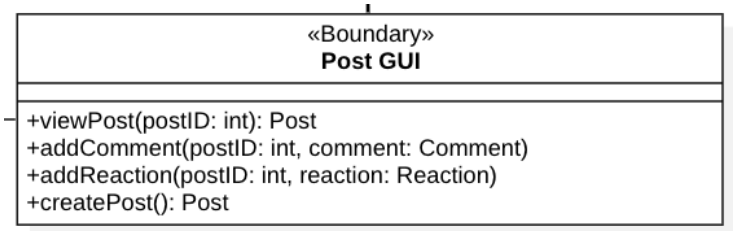
1.5.1.1 Ορισμός κλάσης Post Controller



Μέθοδοι κλάσης:

- **CreateFeed():** - Κατά την εκκίνηση του συγκεκριμένου GUI αρχικοποιείται και προβάλλεται το Feed entity.
- **UpdateFeed():** - Αν ο χρήστης το επιλέξει μπορεί να ζητήσει χειροκίνητο έλεγχο για ενημερώσεις στο feed του
- **visitPost(postID: int):** - Αν ο χρήστης επιλέξει κάποιο Post τότε θα γίνει redirect στην σελίδα του αντίστοιχου

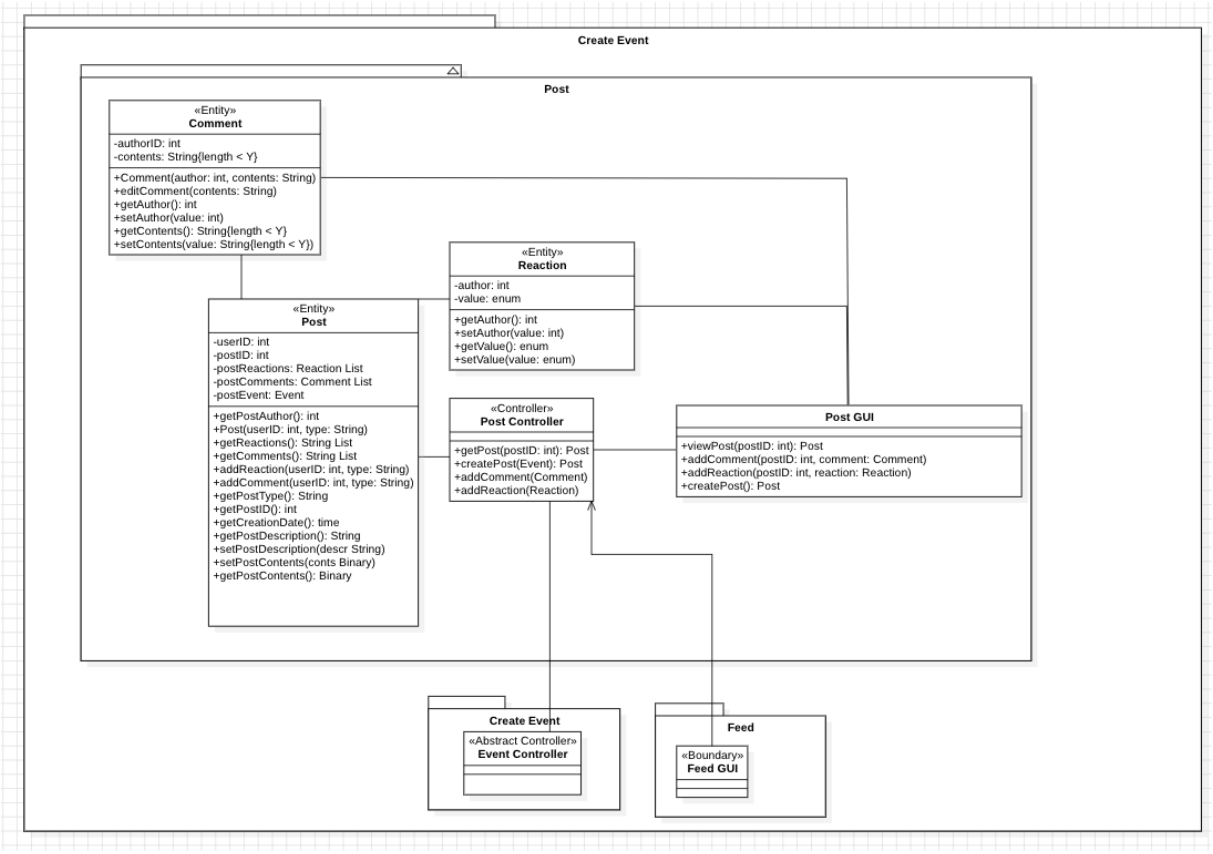
1.5.1.1 Ορισμός κλάσης Post GUI



Μέθοδοι κλάσης:

- **CreateFeed():** - Κατά την εκκίνηση του συγκεκριμένου GUI αρχικοποιείται και προβάλλεται το Feed entity.
- **UpdateFeed():** - Αν ο χρήστης το επιλέξει μπορεί να ζητήσει χειροκίνητο έλεγχο για ενημερώσεις στο feed του
- **visitPost(postID: int):** - Αν ο χρήστης επιλέξει κάποιο Post τότε θα γίνει redirect στην σελίδα του αντίστοιχου

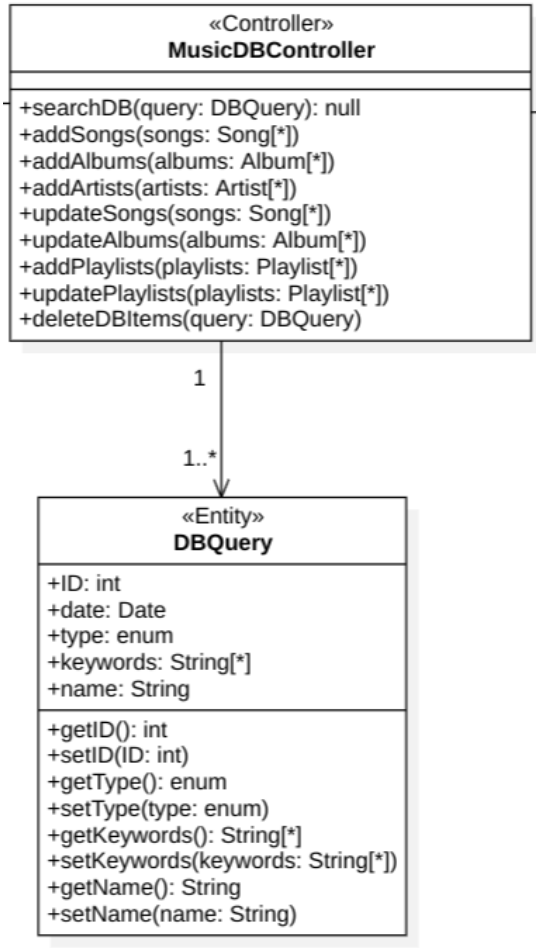
1.5.2 Διάγραμμα κλάσεων



Εικόνα 2: Διάγραμμα κλάσεων για το package Post

2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

2.1 Πρότυπο Command

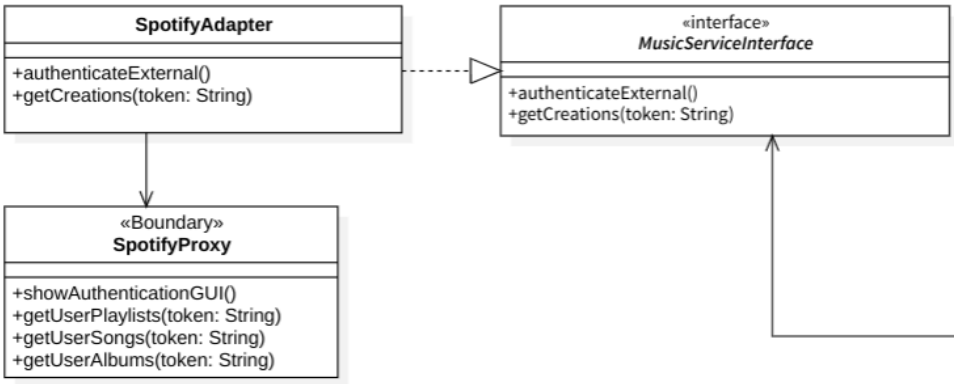


Εικόνα 2: Ενδεικτική εικόνα για το πρότυπο σχεδιασμού Command

Περιγραφή:
Στο σύστημά μας έγινε χρήση του **πρότυπου συμπεριφοράς** Command. Το πρότυπο αυτό χρησιμοποιείται κατά την επικοινωνία του συστήματος με την βάση δεδομένων MusicDB. Καθώς τα query προς την συγκεκριμένη βάση δεδομένων είναι πολύπλοκα και ποικίλουν ως προς το περιεχόμενό τους, η χρήση του συγκεκριμένου πρότυπου σχεδιασμού επιτρέπει την δημιουργία συντηρήσιμου και επεκτάσιμου κώδικα. Ταυτόχρονα, επιτρέπει την διασφάλιση της ασφάλειας κατά την επικοινωνία με την συγκεκριμένη βάση δεδομένων, καθώς περιορίζει τον τύπο και το είδος των queries που μπορούν να εκτελεστούν σε αυτήν.

Σχεδιαστικές Ανάγκες που Ικανοποιούνται:
- ΜΛΑ 3: Η εφαρμογή πρέπει να ακολουθεί τον GDPR
- Δημιουργία συντηρίσιμου
- Δημιουργία επεκτάσιμου κώδικα

2.2 Πρότυπο Adapter

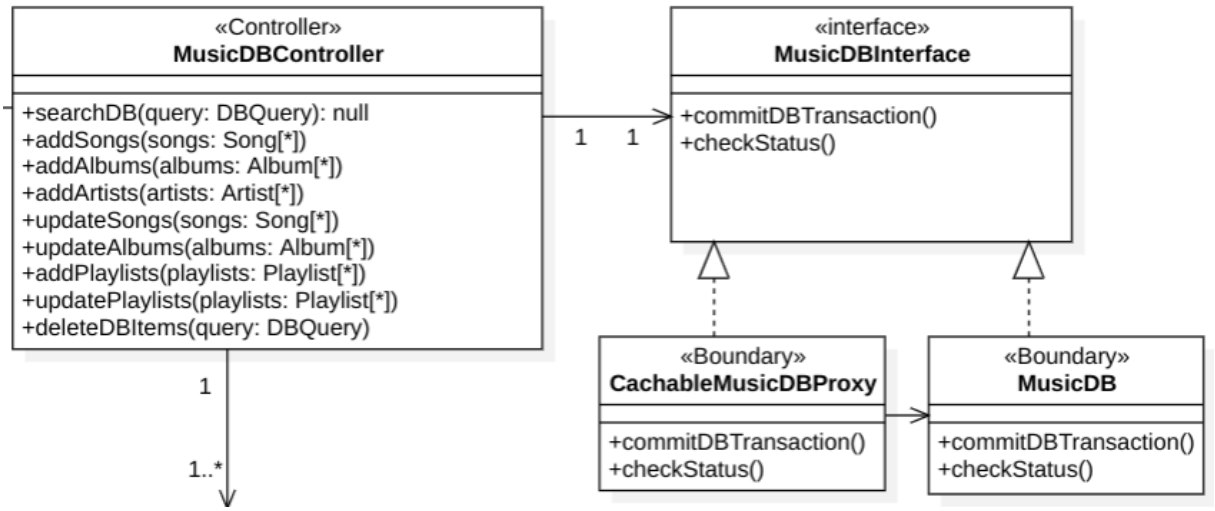


Εικόνα 2: Ενδεικτική εικόνα για το πρότυπο σχεδιασμού Adapter

Περιγραφή:
Στο σύστημά μας έγινε χρήση του **δομικού προτύπου** Adapter. Για την υλοποίηση αυτού του προτύπου, ορίζουμε ένα interface (MusicServiceInterface) με τις απαραίτητες ενέργειες που πρέπει να μπορούν να εκτελεστούν στο εξωτερικό music service. Οι adapters (π.χ. SpotifyAdapter) μετατρέπουν τις διεπαφές των εξωτερικών συστημάτων σε διεπαφές συμβατές με το σύστημά μας, επιτυγχάνοντας την αποτελεσματική επικοινωνία των δύο συστημάτων. Επιπλέον, με την χρήση του συγκεκριμένου προτύπου διευκολύνεται σημαντικά η μελλοντική ενσωμάτωση νέων music service στο σύστημα, καθώς οι ενέργειες που απαιτείται να εκτελεστούν είναι πλέον ορισμένες στο MusicServiceInterface και αρκεί μονάχα η δημιουργία του κατάλληλου adapter.

Σχεδιαστικές Ανάγκες που Ικανοποιούνται:
- Η εφαρμογή πρέπει να υποστηρίζει διάφορα External Music Services
- Δημιουργία επεκτάσιμου κώδικα

2.3 Πρότυπο Proxy



Εικόνα 2: Ενδεικτική εικόνα για το πρότυπο σχεδιασμού Adapter

Περιγραφή:

Στο σύστημά μας έγινε χρήση του **δομικού προτύπου** Proxy. Το πρότυπο αυτό χρησιμοποιείται κατά την επικοινωνία του συστήματος με οποιαδήποτε βάση δεδομένων. Η χρήση του επιτρέπει τον έλεγχο των queries που εκτελούνται στις βάσεις, διασφαλίζοντας την αξιοπιστία και την ασφάλεια του συστήματός μας. Επιπλέον, η εφαρμογή του προτύπου Proxy επιτρέπει την ευέλικτη διαχείριση τις επικοινωνίας με τη βάση δεδομένων, καθιστώντας ευκολότερη και ανθεκτικότερη την εφαρμογή πολιτικών ελέγχου πρόσβασης στις βάσεις. Σε ορισμένες βάσεις του συστήματος (όπως η MusicDB), μπορεί να γίνει χρήση ειδικού Proxy που αποθηκεύει προσωρινά δεδομένα στην μνήμη με τεχνικές caching, μειώνοντας τον αριθμό των queries που εκτελούνται στην βάση δεδομένων και κατ'επέκταση βελτιώνοντας την απόδοση και απόκριση του συστήματος (ειδικά σε περιβάλλοντα πολλών παράλληλων χρηστών).

Σχεδιαστικές Ανάγκες που Ικανοποιούνται:

- ΜΛΑ 3: Η εφαρμογή πρέπει να ακολουθεί τον GDPR
- ΜΛΑ 1: Το σύστημα πρέπει να υποστηρίζει 10.000 ταυτόχρονους χρήστες
- Δημιουργία συντηρήσιμου
- Δημιουργία επεκτάσιμου κώδικα

3 Δυναμική Μοντελοποίηση

Βάσει της οδηγίας για επιλογή των τριών πιο σημαντικών σεναρίων Gherkin, αποφασίσαμε στα 3 συγκεκριμένα

3.1 Gherkin Scenario Share Music Entity

Σημείωση: Ικανοποιεί την ΛΑ-6

Οδηγία: Δώστε μία σύντομη περιγραφή για τον τρόπο με τον οποίο αλληλεπιδρούν οι (εμπλεκόμενες στο συγκεκριμένο Gherkin Scenario X) κλάσεις που προδιαγράψατε στην ενότητα 1) ούτως ώστε να παρέχουν τη λειτουργικότητα που περιγράψατε στο Gherkin Scenario X του Gherkin Feature Y.

Ακολούθως εισάγετε εικόνα του UML διαγράμματος ακολουθιών.

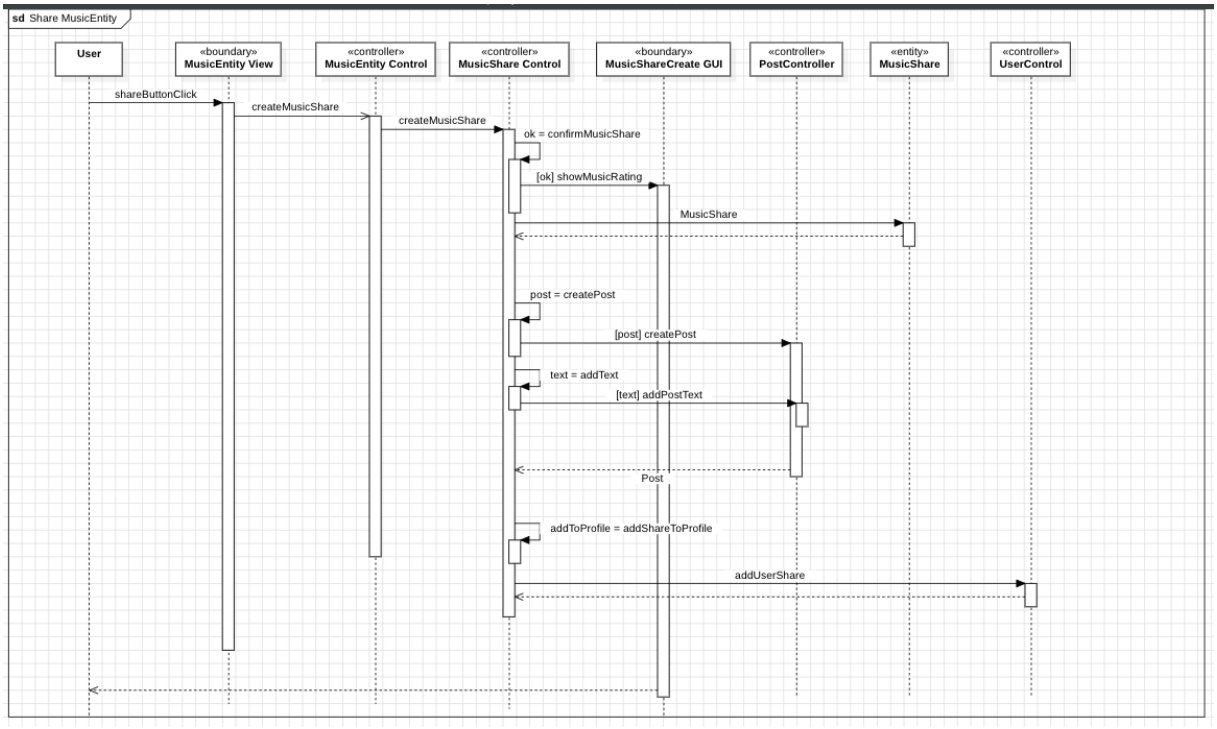
3.1.1 Αφήγηση του σεναρίου Share Music Entity

ΤΟΔΟ

Ο χρήστης βρίσκεται στην οθόνη *MusicEntityView*, και πατάει το κουμπί *rate*. Τότε, η κλάση αυτή καλεί την συνάρτηση *invokeRatingEvent* του *MusicEntityController*. Έπειτα αυτός με την σειρά του καλεί την συνάρτηση *createMusicView* του MusicRatingControl, ο οποίος με την σειρά του δημιουργεί μια οθόνη *MusicRatingCreateGUI*, μέσω της συνάρτησης *initializeView*, την οποία και επιστρέφει στον χρήστη. Ο χρήστης επιλέγει πόσα αστέρια θα βάλει και η οθόνη ανανεώνεται αυτόματα μέσω της συνάρτησης *updateView*. Επιπλέον ο χρήστης έχει την δυνατότητα να εισάγει ένα κείμενο για την κριτική του, καθώς και μπορεί να αποφασίσει αν την κριτική που έγραψε θέλει να την κάνει post η όχι. Έτσι έχουμε τα ακόλουθα 2 υπο-σενάρια:

- Ο χρήστης εισάγει κείμενο αξιολόγησης: Αν το κείμενό του ικανοποιεί όλους τους περιορισμούς, το οποίο εξετάζεται μέσω της συνάρτησης *confirmRating*, τότε η αξιολόγησή του αποθηκεύεται επιτυχώς και το γεγονός αυτό αποθηκεύεται μέσω της συνάρτησης *storeEvent*. Αλλιώς η οθόνη εμφανίζει στον χρήστη ένα μήνυμα που εξηγεί τον λόγο που απορρίφθηκε το κείμενό του μέσω της συνάρτησης *displayInvalidReviewMessage* και η διαδικασία σταματά.
- Ο χρήστης θέλει να κάνει post την αξιολόγησή του: Τότε το σύστημα δημιουργεί ένα post που περιέχει την αξιολόγησή του, αν φυσικά το κείμενο που ενδεχομένως έχει εισάγει είναι αποδεκτό

3.1.2 Σχήμα σεναρίου Share Music Entity



Εικόνα 2: Σχήμα σεναρίου Share Music Entity

3.2 Gherkin Scenario Rate Music Entity

Σημείωση: Ικανοποιεί την ΛΑ-6

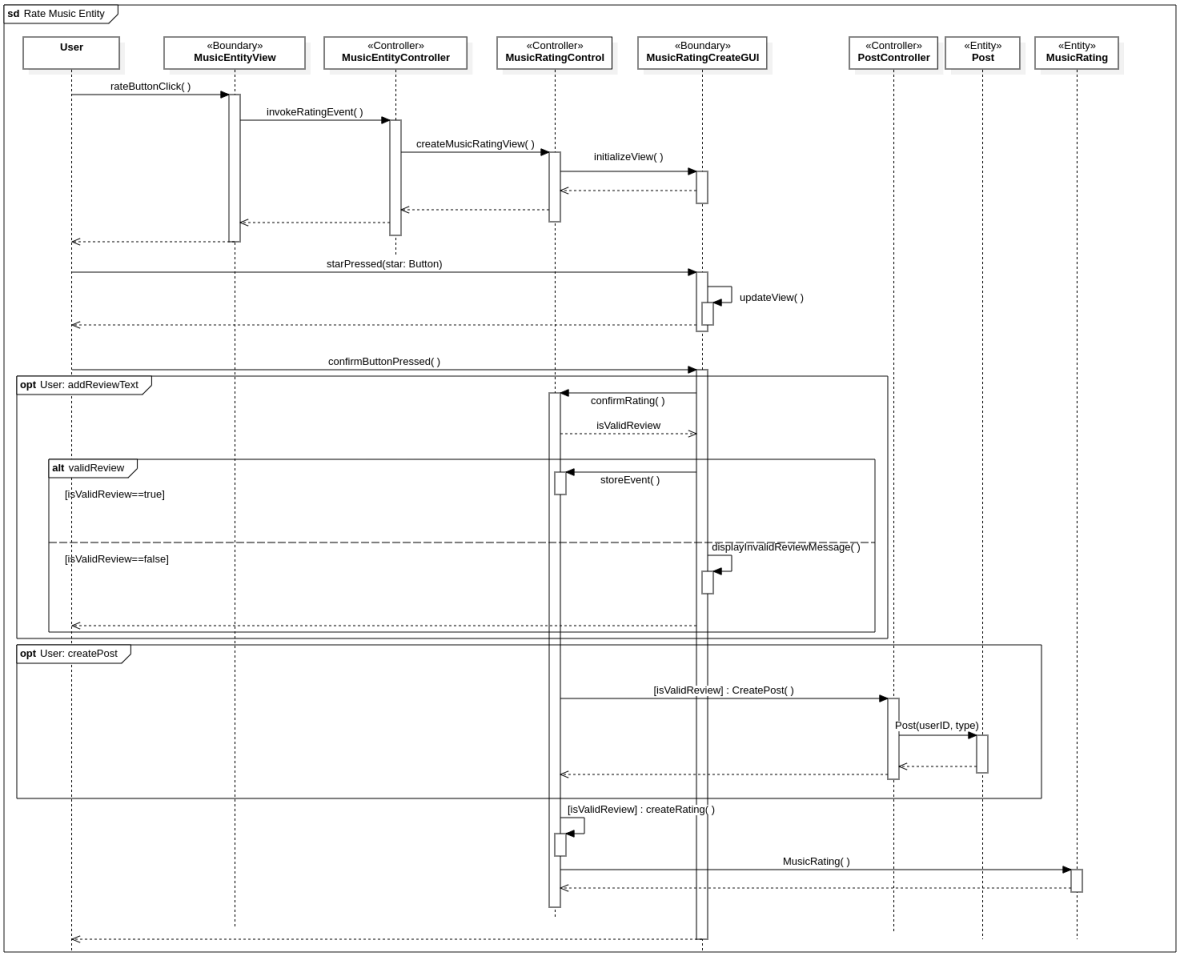
3.2.1 Αφήγηση του σεναρίου Rate Music Entity

Ο χρήστης βρίσκεται στην οθόνη *MusicEntityView*, και πατάει το κουμπί *rate*. Τότε, η κλάση αυτή καλεί την συνάρτηση *invokeRatingEvent* του *MusicEntityController*. Έπειτα αυτός με την σειρά του καλεί την συνάρτηση *createMusicView* του MusicRatingControl, ο οποίος με την σειρά του δημιουργεί μια οθόνη *MusicRatingCreateGUI*, μέσω της συνάρτησης *initializeView*, την οποία και επιστρέφει στον χρήστη. Ο χρήστης επιλέγει πόσα αστέρια θα βάλει και η οθόνη ανανεώνεται αυτόματα μέσω της συνάρτησης *updateView*. Επιπλέον ο χρήστης έχει την δυνατότητα να εισάγει ένα κείμενο για την κριτική του, καθώς και μπορεί να αποφασίσει αν την κριτική που έγραψε θέλει να την κάνει post η όχι. Έτσι έχουμε τα ακόλουθα 2 υπο-σενάρια:

- Ο χρήστης εισάγει κείμενο αξιολόγησης: Αν το κείμενό του ικανοποιεί όλους τους περιορισμούς, το οποίο εξετάζεται μέσω της συνάρτησης *confirmRating*, τότε η αξιολόγησή του αποθηκεύεται επιτυχώς και το γεγονός αυτό αποθηκεύεται μέσω της συνάρτησης *storeEvent*. Αλλιώς η οθόνη εμφανίζει στον χρήστη ένα μήνυμα που εξηγεί τον λόγο που απορρίφθηκε το κείμενό του μέσω της συνάρτησης *displayInvalidReviewMessage* και η διαδικασία σταματά.
- Ο χρήστης θέλει να κάνει post την αξιολόγησή του: Τότε το σύστημα δημιουργεί ένα post που περιέχει την αξιολόγησή του, αν φυσικά το κείμενο που ενδεχομένως έχει εισάγει είναι αποδεκτό

Τελικά το σύστημα, αν πάλι φυσικά το κείμενο που ενδεχομένως έχει εισάγει ο χρήστης είναι αποδεκτό, προχωράει στην δημιουργία ενός αντικειμένου *MusicRating*.

3.2.2 Σχήμα σεναρίου Rate Music Entity



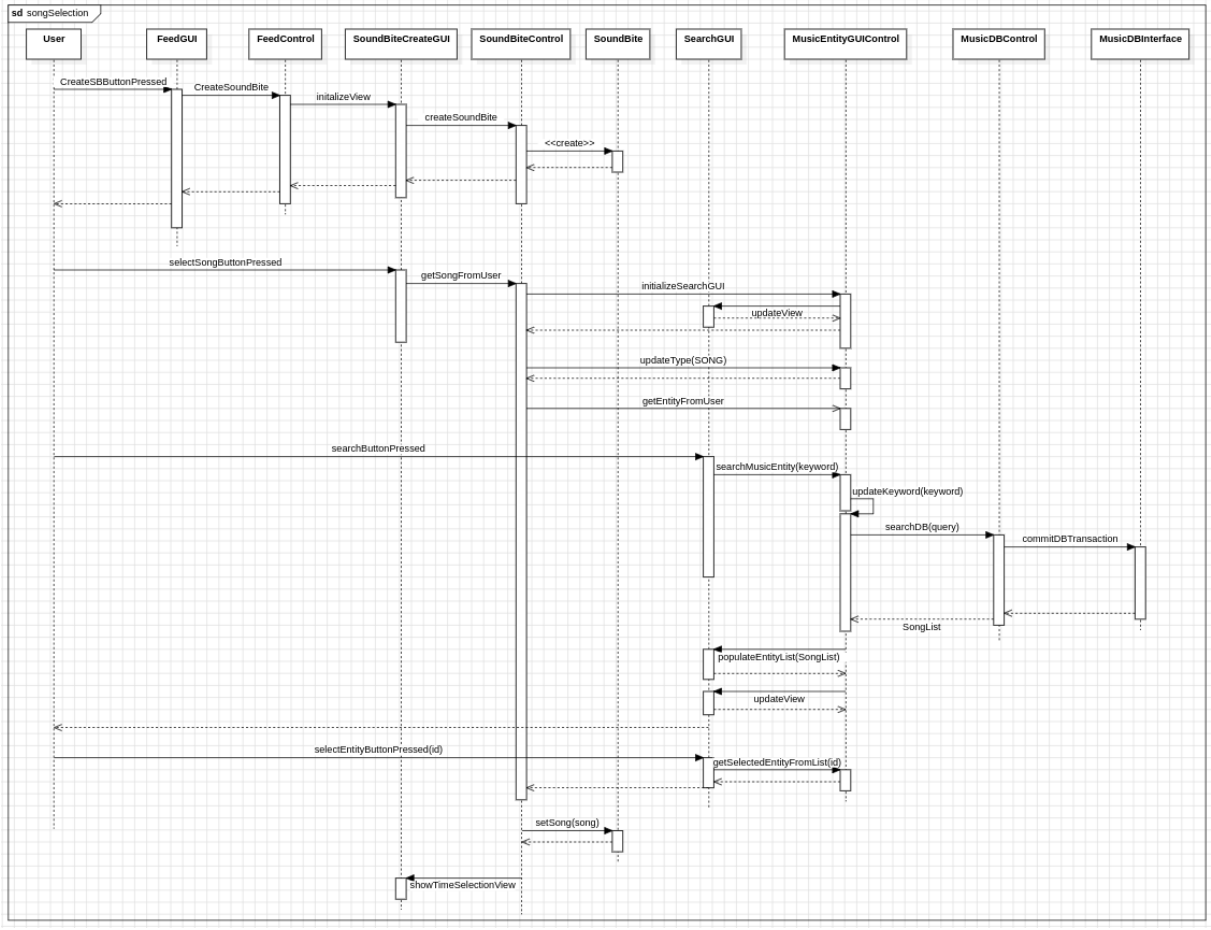
Εικόνα 2: Σχήμα σεναρίου Rate Music Entity

3.3 Gherkin Scenario Successful Song Selection

3.3.1 Αφήγηση του σεναρίου Successful Song Selection

Αρχικά ο χρήστης βρίσκεται στο feed και επιλέγει το κουμπί “createSoudBite”. Αυτό το κουμπί κάνει trigger τον feedController και ο χρήστης μεταβαίνεται στο initial SoundBite creation GUI, δημιουργείται ένας controller και ένα SoundBite το οποίο θα επεξεργαστεί ο χρηστης. Στην αρχική οθόνη ο χρήστης επιλέγει το κουμπί “select song” και μεταβαίνει σε ένα περιβάλλον αναζήτησης τραγουδιού. Καθορίζοντας πρώτα την αναζήτηση ως μόνο για τραγούδια μέσω του υπεύθυνου GUI και controller ο χρήστης επικοινωνεί με το database για να αναζητήσει τραγούδια με κάποια keywords, τα οποία μεταφέρονται στο interface με το Database και επιστρέφονται τα αποτελέσματα της αναζήτησης. Έπειτα ο χρήστης πατά το κουμπί που αντιστοιχεί στο τραγούδι που επιλέγει, και το αντίστοιχο τραγούδι επιστρέφεται στον soundBiteController, ο οποίος στη συνέχεια μεταφέρει τον χρήστη στο περιβάλλον επιλογής χρόνου.

3.3.2 Σχήμα σεναρίου Successful Song Selection



Εικόνα 2: Σχήμα σεναρίου Successful Song Selection

Παράρτημα Ι – Γλωσσάριο

Το σετ των ακρωνυμίων που χρησιμοποιείτε στο έγγραφο

ΛΑ	Λειτουργική Απαίτηση
ΜΛΑ	Μη Λειτουργική Απαίτηση
ΟΑ	Ομάδα Εργασίας
GUI	Γραφικό Περιβάλλον Χρήστη
DB	DataBase