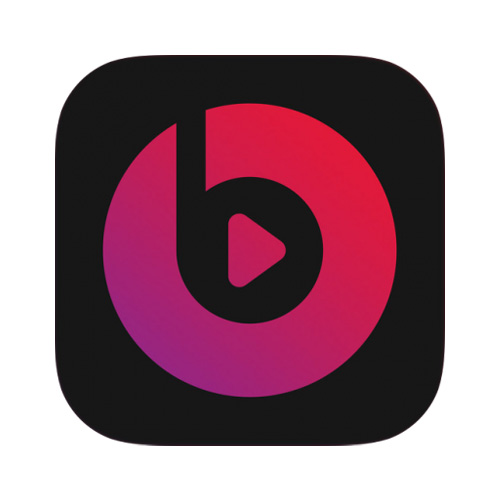
|  |
| --- |



**AudioSyncLinien**Elevate your music experience

**Προδιαγραφές Συστήματος**

**Μηχανική Λογισμικού Ι…Version 2.0**

**Δημήτρης Μανταλόβας**

**Ηλίας Φυτσιλής**

**Θεόδωρος Κεσόγλου**

**Φώτης Κατσάρας**

**12/5/2024**

**Μέλη της Ομάδας Ανάπτυξης**

| **Όνομα** | **OA** | **Email** |
| --- | --- | --- |
| Α. Συμεωνίδης | \* | [asymeon@issel.ee.auth.gr](mailto:asymeon@issel.ee.auth.gr) |
| Φώτης Κατσάρας | Ομάδα 17 | [fkatsaras@ece.auth.gr](mailto:fkatsaras@ece.auth.gr) |
| Δημήτρης Μανταλόβας | Ομάδα 17 | theodorok@ece.auth.gr |
| Θεόδωρος Κεσόγλου | Ομάδα 17 | dmantalo@ece.auth.gr |
| Ηλίας Φυτσιλής | Ομάδα 17 | fytsilias@ece.auth.gr |

**Πίνακας Περιεχομένων**

[**1 Στατική Μοντελοποίηση 5**](#_1fob9te)

[1.1 Search Songs, Artists: Πακέτο κλάσεων Gherkin Feature 1 5](#_3znysh7)

[1.1.1 Ορισμός Κλάσεων 5](#_chmh9pf22ds1)

[1.1.1.1 SearchGUI 5](#_16c7vgnu9tmf)

[1.1.1.2 SearchResult 6](#_1t3h5sf)

[1.1.1.3 SearchManager 6](#_1t3h5sf)

[1.1.1.4 SongDBProxy 7](#_1t3h5sf)

[1.1.1.5 AritstDBProxy 7](#_1t3h5sf)

[1.1.2 Διάγραμμα κλάσεων 8](#_g2jjgjrm2aj8)

[1.2 View Artist Information: Πακέτο κλάσεων Gherkin Feature 2: 9](#_3znysh7)

[1.2.1 Ορισμός Κλάσεων 9](#_qt3infg60im6)

[1.2.1.1 ArtistPageGUI 9](#_1t3h5sf)

[1.2.1.2 ArtistPageManager 10](#_1t3h5sf)

[1.2.1.3 Artist 10](#_qtmrb26r84ue)

[1.2.2 Διάγραμμα κλάσεων 11](#_guz8frnaxjbh)

[1.3 View Playlist Information: Πακέτο κλάσεων Gherkin Feature 3 12](#_3znysh7)

[1.3.1 Ορισμός Κλάσεων 12](#_1yhukl6f0r2m)

[1.3.1.1 ArtistPageGUI 12](#_1t3h5sf)

[1.3.1.2 PlaylistPageManager 13](#_1t3h5sf)

[1.3.1.3 Playlist 13](#_5wn980ij0acr)

[1.3.1.4 PlaylistDBProxy 14](#_1t3h5sf)

[1.3.2 Διάγραμμα κλάσεων 15](#_4d34og8)

[1.4 View Song Information: Πακέτο κλάσεων Gherkin Feature 4 15](#_3znysh7)

[1.4.1 Ορισμός Κλάσεων 16](#_ypox3u6x1ul)

[1.4.1.1 SongPageGUI 16](#_1t3h5sf)

[1.4.1.2 SongPageManager 17](#_mb0ke51wvwy2)

[1.4.1.3 Song 18](#_1t3h5sf)

[1.4.2 Διάγραμμα κλάσεων 19](#_4d34og8)

[1.5 Play a Song: Πακέτο κλάσεων Gherkin Feature 5 19](#_3znysh7)

[1.5.1 Ορισμός Κλάσεων 19](#_g78xpde56zka)

[1.5.1.1. SongPlaybackManager 19](#_lzje9ls63gvh)

[1.5.2 Διάγραμμα κλάσεων 20](#_iptuikz3zhth)

[1. 6 Like a Song: Πακέτο κλάσεων Gherkin Feature 6 21](#_3znysh7)

[1.6.1 Ορισμός Κλάσεων 21](#_2x2ep7t5dcb9)

[1.6.1.1 SongLikeManager 21](#_4kj8gv3qya40)

[1.6.2 Διάγραμμα κλάσεων 21](#_4d34og8)

[1.7 Add a Song to a Playlist: Πακέτο κλάσεων Gherkin Feature 7 21](#_3znysh7)

[1.7.1 Ορισμός Κλάσεων 21](#_sidouwd7f075)

[1.7.1.1 AddSongGUI 21](#_uto3rdij0y1r)

[1.7.1.2 ΑddSongManager 22](#_nopw5oqc5e9b)

[1.7.2 Διάγραμμα κλάσεων 23](#_4d34og8)

[1.8 Create a Playlist: Πακέτο κλάσεων Gherkin Feature 8 23](#_3znysh7)

[1.8.1 Ορισμός Κλάσεων 23](#_u44e196ij7t2)

[1.8.1.1 PlaylistCreationManager 23](#_tf56rh6ish1)

[1.8.1.2 MyPlaylistsPageGUI 23](#_98d2z8q90ego)

[1.8.1.3 PlaylistCreationToolGUI 24](#_6sklbz9o3z92)

[1.8.2 Διάγραμμα κλάσεων 25](#_9qgngnc2zmgv)

[1.9 Edit a Playlist: Πακέτο κλάσεων Gherkin Feature 9 25](#_3znysh7)

[1.9.1 Ορισμός Κλάσεων 25](#_88lz2nlcjrgt)

[1.9.1.1 PlaylistEditor 25](#_luqvb7fupp8b)

[1.9.1.2 PlaylistCreationToolGUI 26](#_9pps0wyodydx)

[1.9.1.3 ConfirmationPromptGUI 26](#_8mrk5yodgnqb)

[1.9.2 Διάγραμμα κλάσεων 27](#_hh99c5iwkzk5)

[1.10 Follow Artists: Πακέτο κλάσεων Gherkin Feature 10 28](#_3znysh7)

[1.10.1 Ορισμός Κλάσεων 28](#_g3yc24x5m5gs)

[1.10.1.1 ArtistFollowManager 28](#_rb21j2z7l3i7)

[1.10.2 Διάγραμμα κλάσεων 29](#_4d34og8)

[1.11 Get Song Recommendations: Πακέτο κλάσεων Gherkin Feature 11 29](#_3znysh7)

[1.11.1 Ορισμός Κλάσεων 29](#_9725mzxtlqpk)

[1.11.1.1 RecommendationsController 29](#_xdmi07ne9xic)

[1.11.1.2 RecommendationsSystemProxy 29](#_c28qnoga3ik)

[1.11.1.3 ΜyRecommendationsPageGUI 30](#_7n9odepdmqpx)

[1.11.2 Διάγραμμα κλάσεων 31](#_4d34og8)

[1.12 Navigate to Homepage: Πακέτο κλάσεων Gherkin Feature 12\* 31](#_uv75983ma0cm)

[1.12.1 Ορισμός Κλάσεων 31](#_scktec94zfjp)

[1.12.1.1 UserEntity 31](#_p9wda6td2q8n)

[1.12.1.2 NavigationManager 32](#_p9zpcyeovob2)

[1.12.1.3 ΗοmePageGUI 32](#_92xnfyu7brm)

[1.12.1.4 UserDBProxy 33](#_d37xc0ndva5g)

[**2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν 34**](#_2s8eyo1)

[2.1 Πρότυπο Observer 34](#_4lozgf41toky)

[2.2 Πρότυπο Strategy 35](#_sp8uk3bb5mjb)

[2.3 Πρότυπο Proxy 36](#_hzxi0fx5m0ob)

[**3 Δυναμική Μοντελοποίηση 3**](#_17dp8vu)**6**

3.1 Πακέτο ακρόασης ενός τραγουδιού από τον χρήστη…………………………………………………………….36

3.1.1 Αφήγηση Σεναρίου…………………………………………………………………………………………………….36

3.2 Πακέτο δημιουργίας playlist απο τον χρήστη………………………………………………………….…………….36

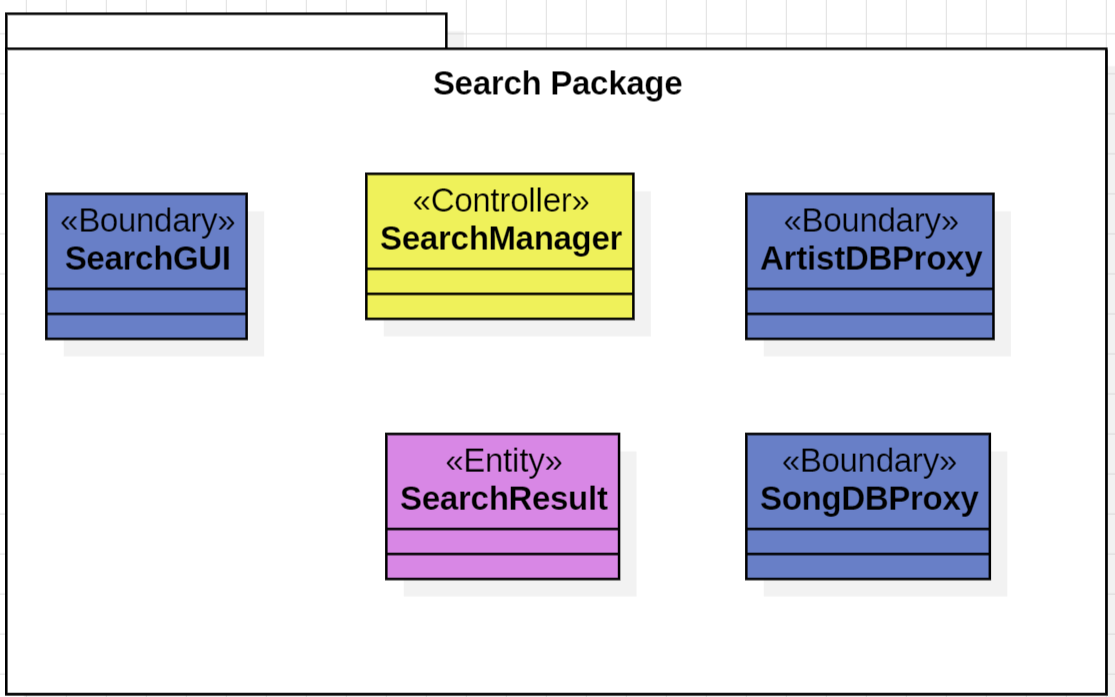
3.2.1 Αφήγηση Σεναρίου……………………………………………………………………………………………………37

3.3 Πακέτο επισκόπησης προτεινόμενων τραγουδιών από τον χρήστη………………………………………37

3.3.1 Αφήγηση Σεναρίου…………………………………………………………..…………………………………………37

# 1 Στατική Μοντελοποίηση

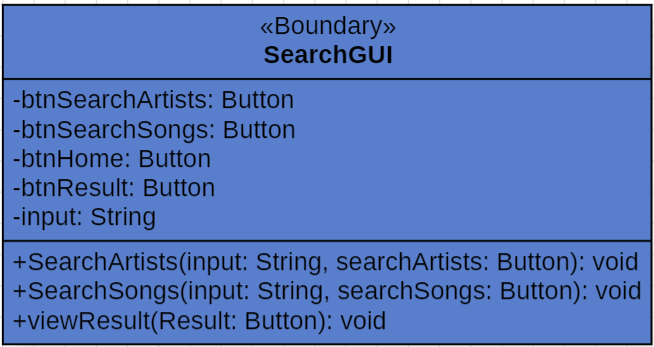
### 1.1 Search Songs, Artists: Πακέτο κλάσεων Gherkin Feature 1



#### 1.1.1 Ορισμός Κλάσεων

##### 1.1.1.1 SearchGUI

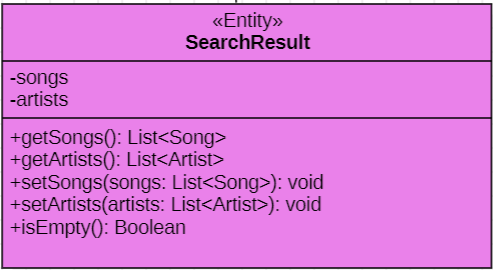
Η κλαση αυτη εκφραζει την διεπαφη της σελιδας αναζητησης



**Χαρακτηριστικά**

1. **btnSearchArtists: Button** : Κουμπί έναρξης αναζήτησης καλλιτεχνών
2. **btnSearchSongs: Button** : Κουμπί έναρξης αναζήτησης τραγουδιών
3. **btnHome: Button :** Κουμπί επιστροφής στην αρχική οθόνη
4. **btnResult: Button** : Κουμπί προβολης αποτελεσματος μιας αναζητησης

##### 1.1.1.2 SearchResult

****

**Χαρακτηριστικά**

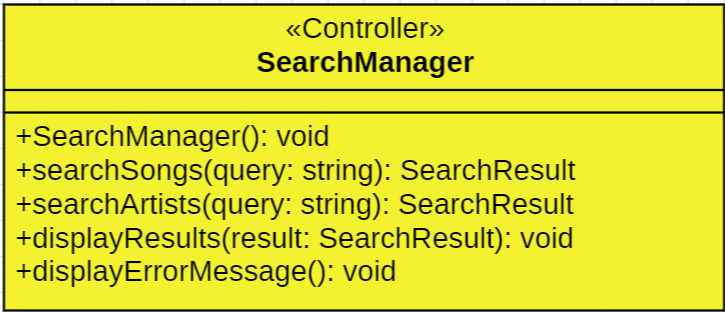
1. **songs: List[Song] :**  Μια λίστα από αντικείμενα τύπου Song που αποτελούν το αποτέλεσμα αναζήτησης τραγουδιών
2. **artists: List[Artist] :** Μια λίστα από αντικείμενα τυπου Artist που αποτελούν το αποτέλεσμα αναζήτησης καλλιτεχνών

**Μεθοδοι**

1. **SearchResult():** Μεθοδος δομησης του αντικειμενου.
2. **getSongs() -> List[Song] :** Επιστρεφει μια λιστα αντικειμενων Song με τα αποτελεσματα αναζητησης τραγουδιων
3. **getArtists() -> List[Artist] :** Επιστρεφει μια λιστα αντικειμενων Artist με τα αποτελεσματα αναζητησης καλλιτεχνων
4. **getSongs(songs: List[Song]) -> void:** Θετει στην μεταβλητη songs του αντικειμενου μια λιστα αντικειμενων Song με τα αποτελεσματα αναζητησης τραγουδιων
5. **isEmpty() -> Boolean :** Επιστρεφει **true** εαν το αποτελεσμα αναζητησης ειναι κενο

##### 1.1.1.3 SearchManager

Η κλαση αυτη συντονιζει την διαδικασια αναζητησης επικοινωνώντας με τις SongDBProxy, ArtistDBProxy και την SearchGUI

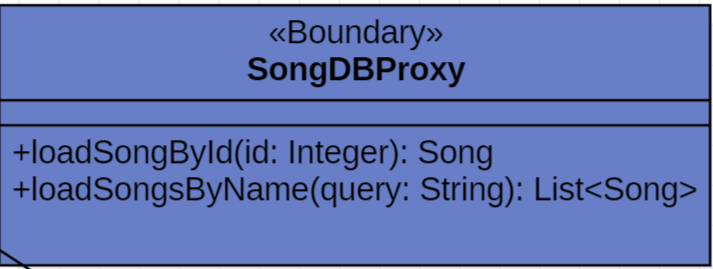
****

**Μεθοδοι**

1. **searchSongs(query: string) -> SearchResult :** Επιστρεφει ενα αντικειμενο SearchResult με songs = List<Song> μεσω της SearchDB
2. **searchArtists(query: string) -> SearchResult:** Επιστρεφει ενα αντικειμενο SearchResult με songs = List<Artis> μεσω της SearchDB
3. **displayResults(result: SearchResult):** Εμφανιζει τα αποτελεσματα αναζητησης στην οθονη.
4. **displayErrorMessage() :** Εμφανιζει ενα μηνυμα στην οθονη σε περιπτωση που η αναζητηση δεν ειχε καποιο αποτελεσμα

##### 1.1.1.4 SongDBProxy

Η κλαση αυτη επικοινωνει με την βαση δεδομενων για την αναζητηση τραγουδιων.

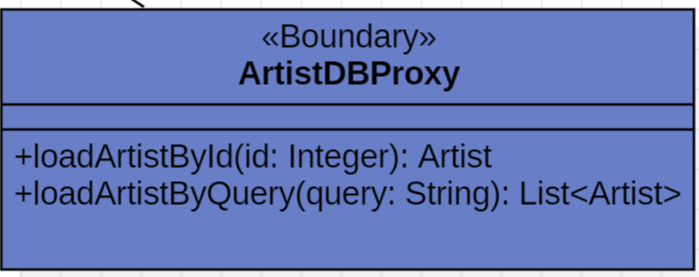


**Μεθοδοι**

1. **loadSongs(id: Integer) :** Επικοινωνει με τη βαση και επιστρεφει ενα αντικειμενο Song που βρισκεται στη βαση και ταιριαζει με το id του ορισματος.
2. **loadSongs(query: String):** Επικοινωνει με τη βαση και επιστρεφει μια λιστα αντικειμενων τυπου Song με τιτλους που ταιριαζουν στο query. Αν το query δεν ταιριαζει με καποιον τιτλο τραγουδιου της βασης επιστρεφεται κενη λιστα.

##### 1.1.1.5 AritstDBProxy

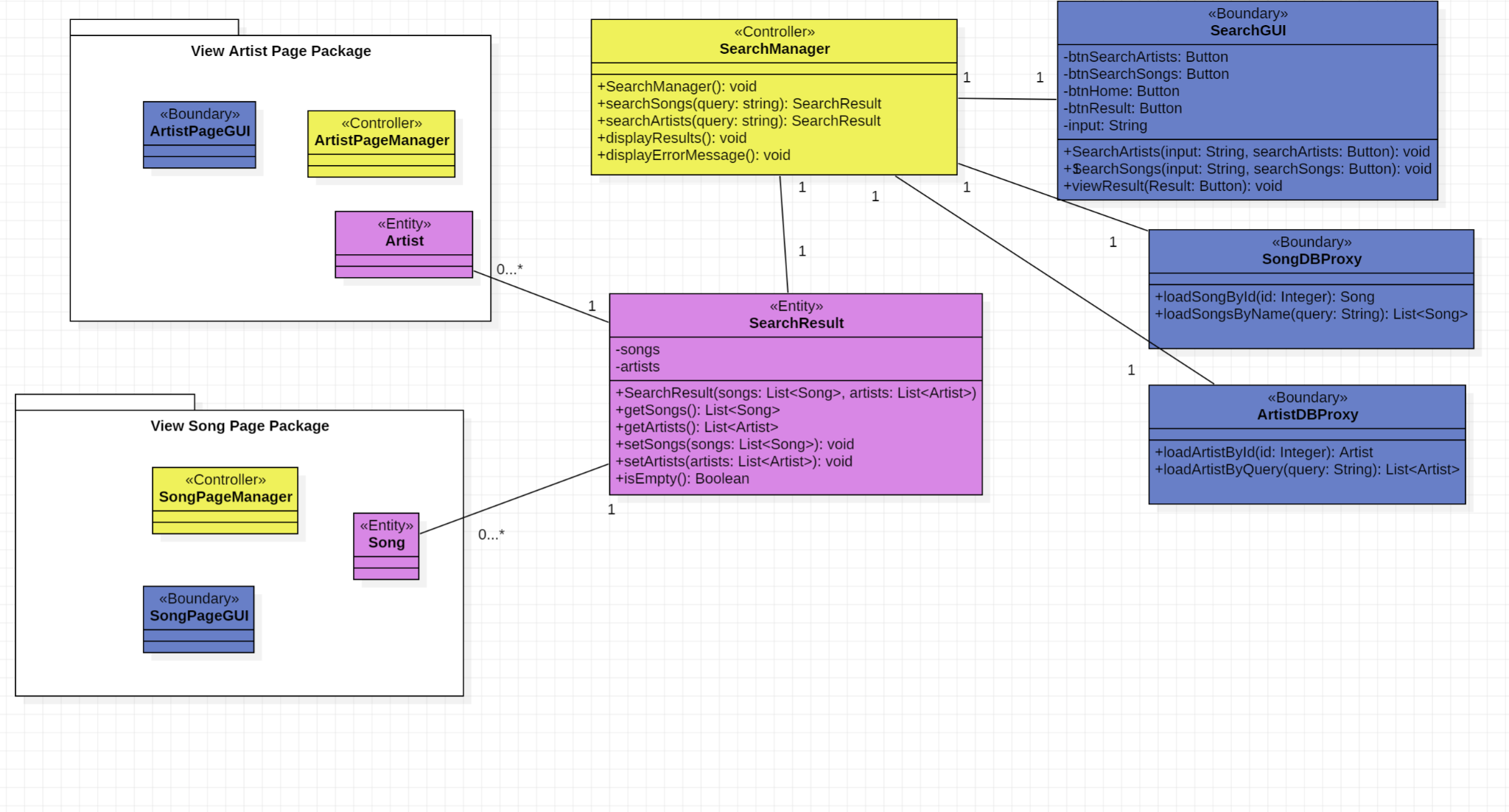
Η κλαση αυτη επικοινωνει με την βαση δεδομενων για την αναζητηση καλλιτεχνων



**Μεθοδοι**

1. **loadArtistById(id: Integer) :** Επικοινωνει με τη βαση και επιστρεφει ενα αντικειμενο Artist που βρισκεται στη βαση και ταιριαζει με το id του ορισματος.
2. **loadArtistByQuery(query: String):** Επικοινωνει με τη βαση και επιστρεφει μια λιστα αντικειμενων τυπου Artist με ονοματα που ταιριαζουν στο query. Αν το query δεν ταιριαζει με καποιο ονομα καλλιτεχνη της βασης, επιστρεφεται κενη λιστα.

#### 1.1.2 Διάγραμμα κλάσεων



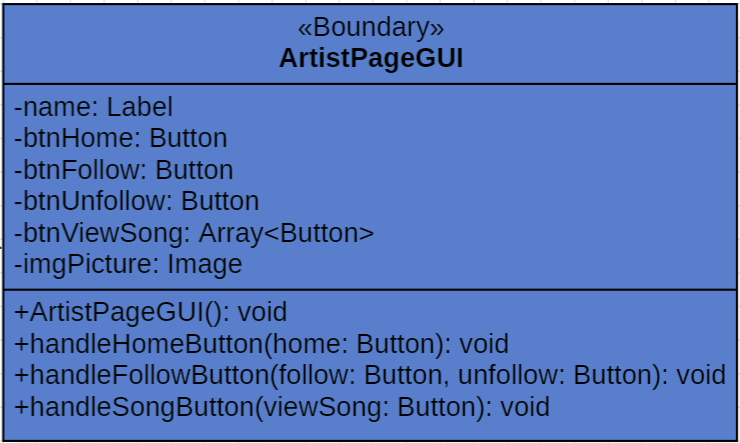
### 1.2 View Artist Information: Πακέτο κλάσεων Gherkin Feature 2:

### 

#### 1.2.1 Ορισμός Κλάσεων

##### 1.2.1.1 ArtistPageGUI

Η κλαση αυτη εκφραζει την διεπαφη της σελιδας ενος καλλιτεχνη



**Χαρακτηριστικα**

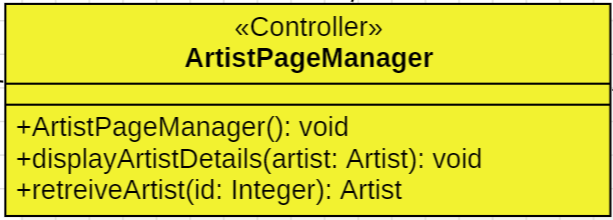
1. **name:** Μια ετικετα με το ονομα του καλλιτεχνη
2. **btnHome:** Κουμπι επιστροφης στην αρχικη οθονη
3. **btnFollow:** Κουμπι για να επισημανει ο χρηστης τον καλλιτεχνη ως <<followed>>
4. **btnUnfollow:** Κουμπι για να επισημανει ο χρηστης τον καλλιτεχνη ως <<unfollowed>>
5. **btnViewSong:** Μια λιστα απο κουμπια που αντιπροσωπεύουν τραγουδια του καλλιτεχνη.
6. **imgPicture:**  Μια εικονα με την φωτογραφια του καλλιτεχνη

**Μεθοδοι**

1. **ArtistPageGUI():** Συναρτηση δομησης της κλασης
2. **handleHomeButton(home: Button):** Συναρτηση διαχειρησης της χρησης του κουμπιου για επιστροφη στην αρχικη οθονη
3. **handleFollowButton(follow: Button, unfollow: Button)**: Καλειται η συναρτηση followArtist() ή η unfollowArist() του ελεγκτη ArtistFollowManager() για να γινει <<Followed>> ή <<Unfollowed>> o καλλιτεχνης αναλογα με το κουμπι που επιλεχθηκε
4. **handleSongButton(viewSong: Button):** Καλειται η συναρτηση retreiveSong(id) του SongPageManager για την ανακτηση του τραγουδιου με το συγκεκριμενο id απο την βαση και επειτα η displaySongDeatails(Song) για την προβολη της σελιδας του τραγουδιου που επιλεχθηκε.

##### 1.2.1.2 ArtistPageManager

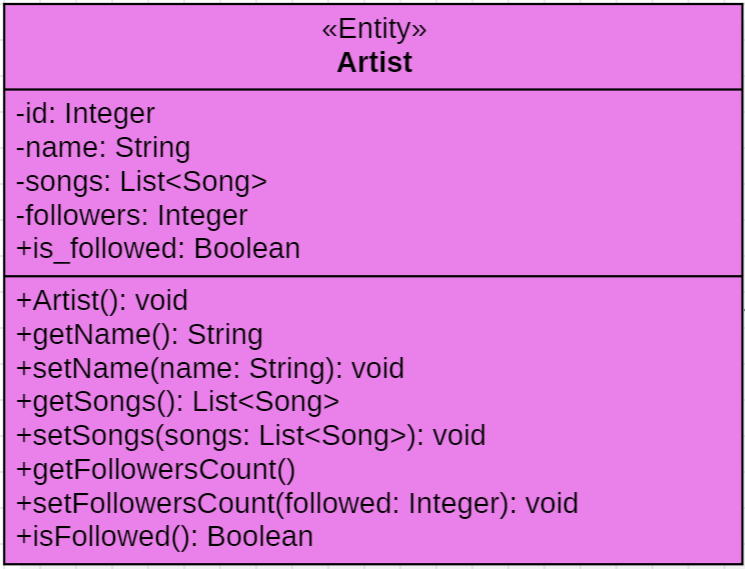
Η κλαση αυτη συντονιζει την προβολη των στοιχειων του καλλιτεχνη, επικοινωνωντας με τις ArtistDBProxy και την ArtistPageGUI



**Μεθοδοι**

1. **ArtistPageManager():** Συναρτηση δομησης της κλασης
2. **displayArtistDetails(artist: Artist):** Προβαλλει τα στοιχεια του καλλιτεχνη στην οθονη
3. **retrieveArtist(id: Integer):** Επιστρεφει απο την βαση τον καλλιτεχνη με το συγκεκριμενο Id

##### 1.2.1.3 Artist

Η κλαση αυτη περιεχει τα δεδομενα της οντοτητας ενος καλλιτεχνη / συγκροτηματος****

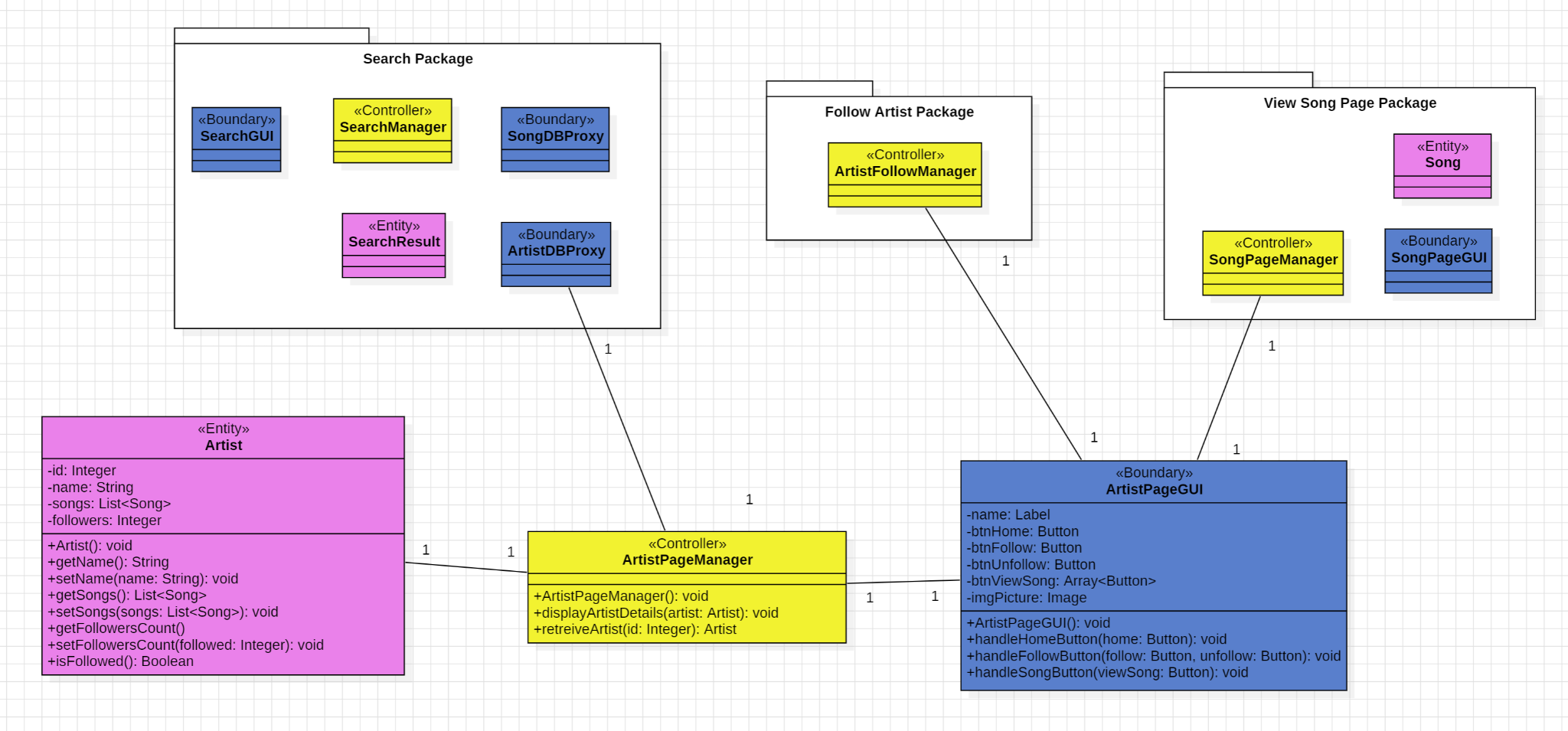
**Χαρακτηριστικα**

1. **id:** Αποτελεί τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του κάθε καλλιτεχνη
2. **name** : Το ονομα του καλλιτεχνη / συγκροτηματος
3. **songs** : Μια λιστα αντικειμενων τυπου **Song** τα οποια ανηκουν στον συγκεκριμενο καλλιτεχνη
4. **followers:** Ενας ακεραιος που δειχνει τον αριθμο των ακολουθων του καλλιτεχνη
5. **is\_followed** : Ενα flag που καθοριζει αν ο χρηστης ακολουθει τον καλλιτεχνη

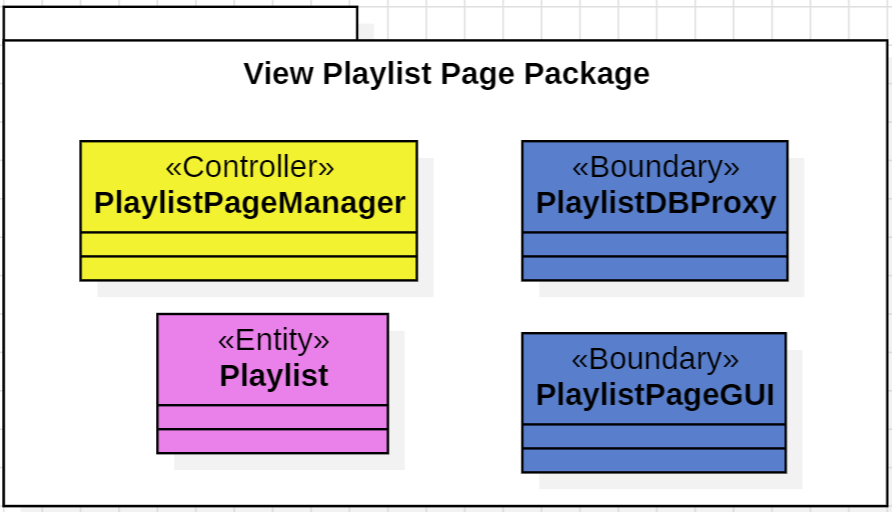
**Μεθοδοι**

1. **Artist():** Μεθοδος δομησης του αντικειμενου Artist
2. **getName() :** Επιστεφει το ονομα του καλλιτεχνη
3. **setName(name: String) :** Τροποποιει το ονομα του καλλιτεχνη
4. **getSongs() :** Επιστρεφει μια λιστα απο τα τραγουδια (αντικειμενα Song) που ανηκουν στον καλλιτεχνη
5. **setSongs(songs: List<Song>) :** Αλλαζει τα τραγουδια (αντικειμενα Song) που ανηκουν στον καλλιτεχνη
6. **getFollowersCount() :** Επιστρεφει τον αριθμο των ακολουθων του χρηστη
7. **setFollowersCount(followers: Integer) :** Τροποποιει τον αριθμο των ακολουθων του χρηστη
8. **isFollowed() -> bool :** Επιστρεφει **true** εαν ο χρηστης ακολουθει τον καλλιτεχνη

#### 1.2.2 Διάγραμμα κλάσεων

****

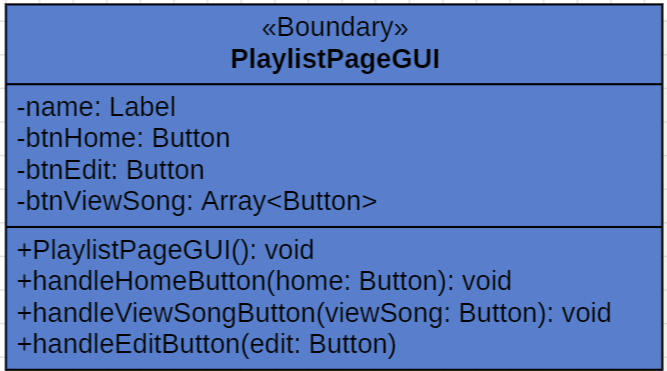
### 1.3 View Playlist Information: Πακέτο κλάσεων Gherkin Feature 3



#### 1.3.1 Ορισμός Κλάσεων

##### 1.3.1.1 PlaylistPageGUI

Η κλαση αυτη εκφραζει την διεπαφη της σελιδας ενος playlist



**Χαρακτηριστικα**

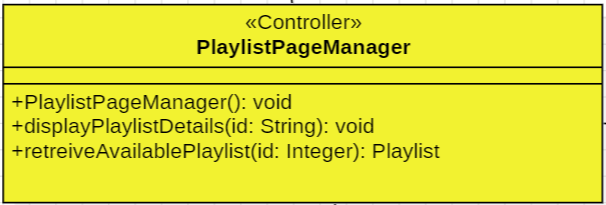
1. **name:** Μια ετικετα με τον τιτλο του playlist
2. **btnHome:** Κουμπι επιστροφης στην αρχικη οθονη
3. **btnEdit:** Κουμπι για την επεξεργασια του playlist
4. **btnViewSong:**  Μια λιστα απο κουμπια που αντιπροσωπευουν τραγουδια που ανηκουν στο playlist
5. **btnDeleteSong:** Μια λιστα απο κουμπια που χρησιμοποιουνται για την διαγραφη τραγουδιων απο το playlist

**Μεθοδοι**

1. **PlaylistPageGUI():** Συναρτηση δομησης της κλασης
2. **handleHomeButton(home: Button):** Συναρτηση διαχειρησης της χρησης του κουμπιου για επιστροφη στην αρχικη οθονη
3. **handleViewSongButton(viewSong: Button):** Καλειται η συναρτηση retreiveSong(id) του SongPageManager (Πακετο View Song Page) για την ανακτηση του τραγουδιου με το συγκεκριμενο id απο την βαση και επειτα η displaySongDeatails(Song) για την προβολη της σελιδας του τραγουδιου που επιλεχθηκε.
4. **handleDeleteSongButton(deleteSong: Button):** Καλει την συναρτηση removeSong(song: Song) του PlaylistEditor (Πακετο Edit Playlist) για να αφαιρεσει το επιλεγμενο τραγουδι απο το playlist
5. **handleEditButton(editTitle: Button):** Καλει την συναρτηση displayEditButtons() του PlaylistEditor (Πακετο Edit Playlist) για να εμφανιστουν τα κουμπια επεξεργασιας του playlist

##### 1.3.1.2 PlaylistPageManager

Η κλαση αυτη συντονιζει την προβολη των στοιχειων του playlist, επικοινωνωντας με τις PlaylistDBProxy και την PlaylistPageGUI

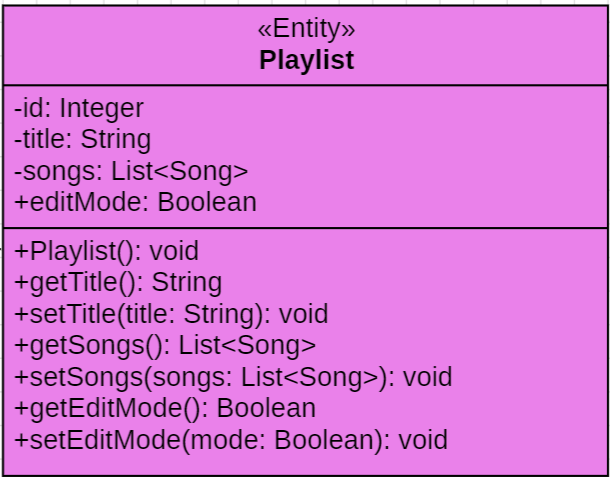


**Μεθοδοι**

1. **PlaylistPageManager()** Συναρτηση δομησης του ελεγκτη
2. **displayPlaylistDetails(id: String):** Προβαλλει τα στοιχεια του playlist στην οθονη
3. **retreiveAvailablePlaylist(id: Integer):** Επιστρεφει ενα playlist απο τη βαση δεδομενων με βαση το id του

##### 1.3.1.3 Playlist

Η κλαση αυτη περιεχει τα δεδομενα της οντοτητας ενος playlist



**Χαρακτηριστικα**

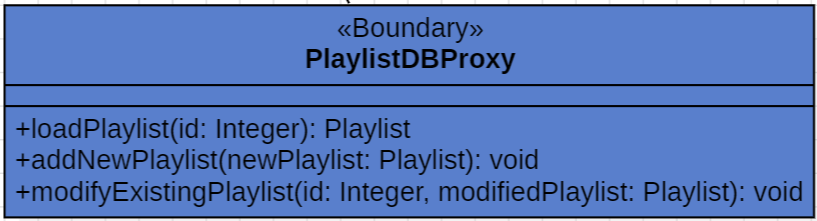
1. **id :** Αποτελεί τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του κάθε playlist
2. **title :** Ο τιτλος του playlist
3. **songs:** Μια λιστα αντικειμενων Song - τα τραγουδια που περιεχει το playlist
4. **editMode:** Ενα boolean flag που καθοριζει αν ο χρηστης μπορει να επεξεργαστει το playlist

**Μεθοδοι**

1. **Playlist():** Συναρτηση δομησης της κλασης
2. **getTitle():** Επιστρεφει τον τιτλο του playlist
3. **setTitle(title: String):** Τροποποιει τον τιτλο του playlist
4. **getSongs(): List<Song>:** Επιστρεφει τα τραγουδια που περιεχει το playlist
5. **setSongs(songs: List<Song>):** Τροποποιει τα τραγουδια που περιεχει το playlist
6. **getEditMode():** Επιστρεφει το flag για την τροποποιηση του playlist
7. **setEditMode(mode: Boolean):** Αλλαζει το flag τροποποιησης του playlist

##### 1.3.1.4 PlaylistDBProxy

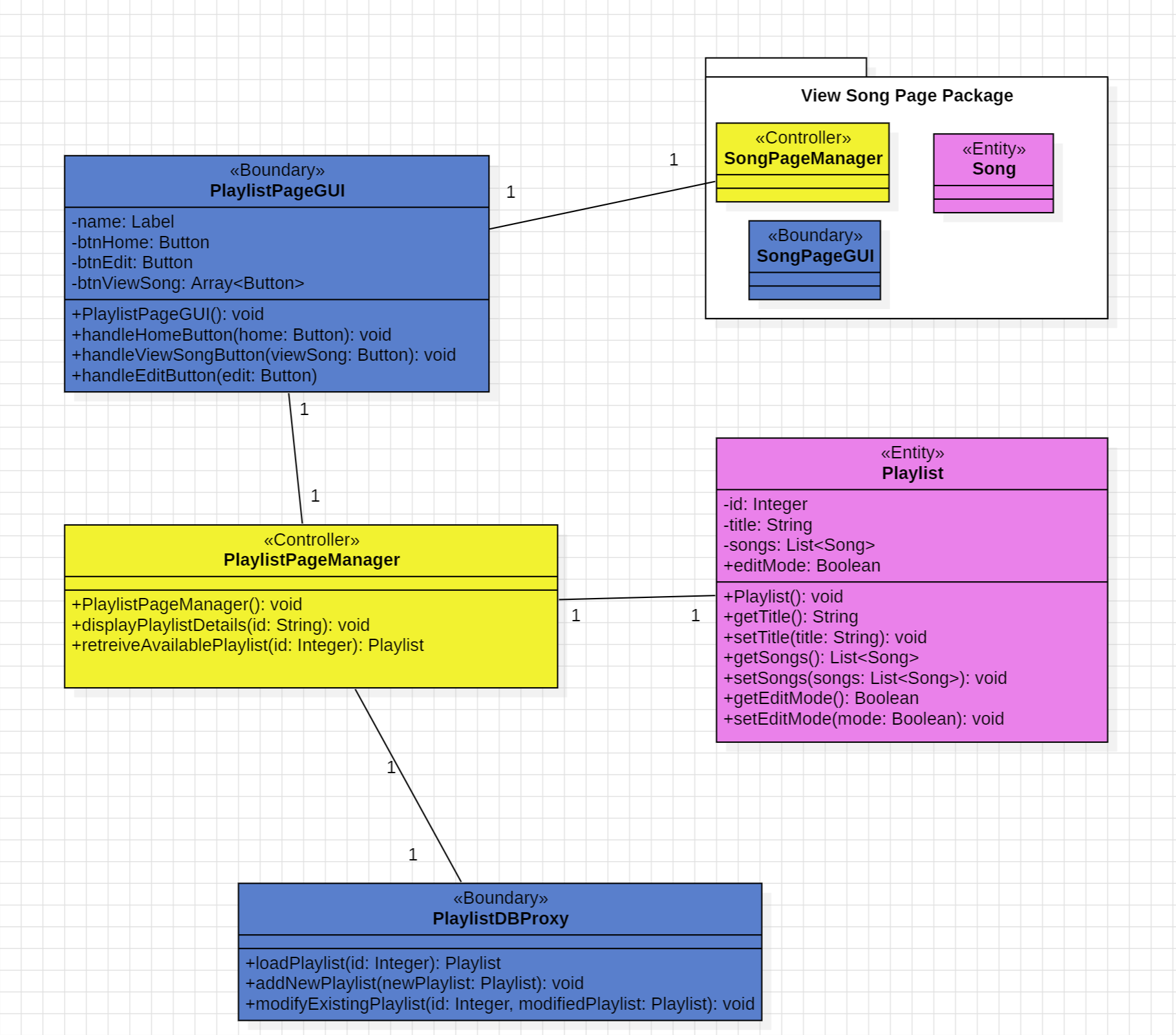
Η κλαση αυτη επικοινωνει με την βαση δεδομενων για την αναζητηση και την τροποποιηση playlist που υπάρχουν στη βαση



**Μεθοδοι**

1. **loadPlaylist(id):** Επιστρεφει απο την βαση το playlist με το id του input
2. **addNewPlaylist(id, newPlaylist):** Καταχωρει στη βαση ενα καινουριο playlist
3. **modifyExistingPlaylist(id, modifiedPlaylist):** Καταχωρει στο playlist της βασης με το id του input, τα στοιχεια του modifiedPlaylist

#### 1.3.2 Διάγραμμα κλάσεων

****

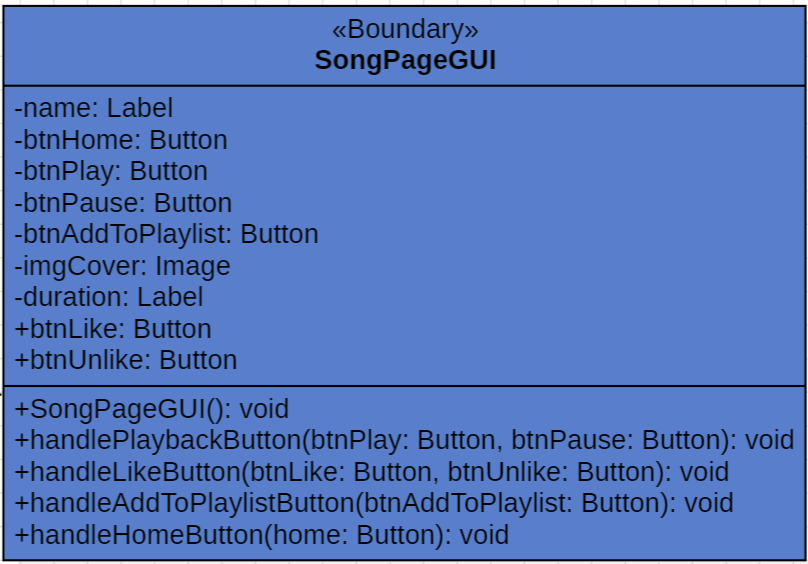
### 1.4 View Song Information: Πακέτο κλάσεων Gherkin Feature 4



#### 1.4.1 Ορισμός Κλάσεων

##### 1.4.1.1 SongPageGUI

Η κλαση αυτη εκφραζει την διεπαφη της σελιδας ενος τραγουδιου



**Χαρακτηριστικα**

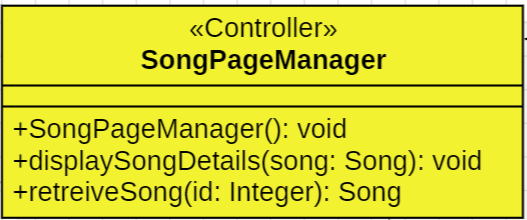
1. **name**: Μια ετικετα με το ονομα του τραγουδιου
2. **btnHome:** Κουμπι επιστροφης στην αρχικη οθονη
3. **btnPlay:** Κουμπι εναρξης αναπαραγωγης τραγουδιου
4. **btnPause:** Κουμπι παυσης αναπαραγωγης τραγουδιου
5. **btnAddToPlaylist:** Κουμπι προσθηκης τραγουδιου σε playlist
6. **imgCover:** Εικονα τραγουδιου
7. **duration:** Ετικετα συνολικης διαρκειας τραγουδιου
8. **btnLike:** Κουμπι για την επισημανση του τραγουδιου ως liked απο τον χρηστη
9. **btnUnlike:** Κουμπι για την επισημανση του τραγουδιου ως unliked απο τον χρηστη

**Μεθοδοι**

1. **SongPageGUI():** Συναρτηση δομησης της κλασης
2. **handlePlaybackButton(btnPlay, btnPause):** Καλει την συναρτηση playSong / pauseSong του SongPlaybackManager (Πακετο Play a Song) για την αναπαραγωγη/ παυση αναπαραγωγης του τραγουδιου, αναλογα με το input.
3. **handleLikeButton(btnLike, btnUnlike):** Καλει την συναρτηση likeSong/ unlikeSong του SongLikeManager (Πακετο Like a Song) για την επισημανση του τραγουδιου ως liked / unliked αντιστοιχα.
4. **handleAddtoPlaylistButton(btnAddToPlaylist):** Καλει την συναρτηση addSongToPlaylist() του AddSongManager() (Πακετο Add Song To a Playlist) για την προσθηκη του τραγουδιου σε μια απο τις δημιουργημενες playlist του χρηστη

##### 1.4.1.2 SongPageManager

Η κλαση αυτη συντονιζει την προβολη των στοιχειων του τραγουδιου, επικοινωνωντας με τις Song, SongDBProxy και την SongPageGUI



**Μεθοδοι**

1. **SongPageManager():** Συναρτηση δομησης της κλασης
2. **displaySongDetails(song):** Προβαλλει τα στοιχεια του τραγουδιου στην οθονη
3. **retrieveSong(id):** Επιστρεφει απο την βαση το τραγουδι με το id που εχει δοθει ως input

##### 1.4.1.3 Song

##### 

Η κλαση αυτη περιεχει τα δεδομενα της οντοτητας ενος τραγουδιου

****

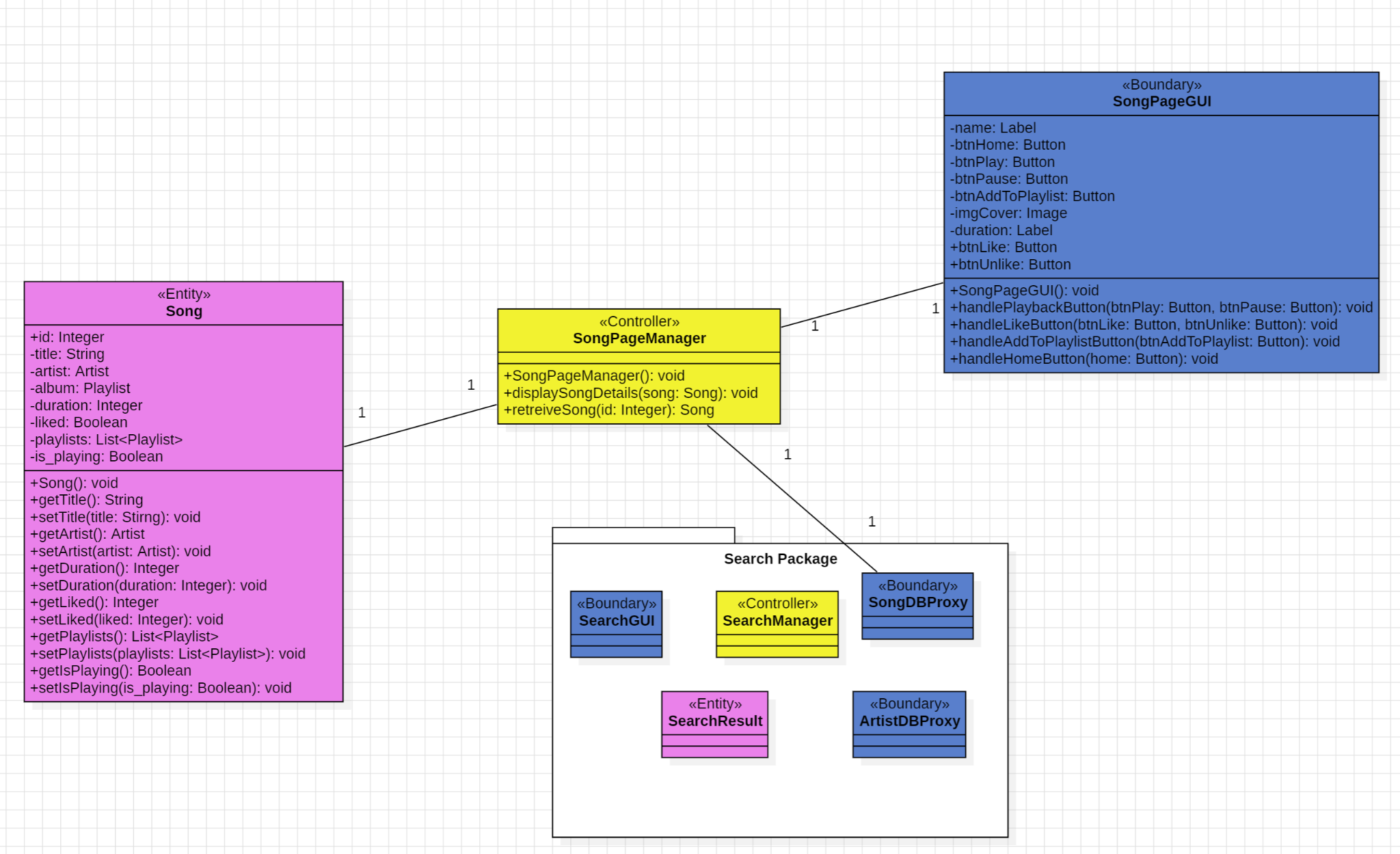
**Χαρακτηριστικα**

1. **id:** Αποτελεί τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του κάθε τραγουδιού
2. **title:** : Ο τιτλος / το ονομα του τραγουδιου
3. **artist:**  To ονομα του καλλιτεχνη που δημιουργησε το τραγουδι
4. **album:**  To oνομα του αλμπουμ που περιεχει το τραγουδι
5. **duration:** Η χρονικη διαρκεια του τραγουδιου
6. **cover:** Το εξωφυλλο του αλμπoυμ στο οποιο ανηκει το τραγουδι
7. **liked**: Ενα flag που καθοριζει αν ο χρηστης εχει κανει like στο τραγουδι
8. **playlists**: Μια λιστα απο αντικειμενα Playlist στα οποια εχει βαλει ο χρηστης το τραγουδι
9. **is\_playing:**  Ενα Boolean flag που καθοριζει αν το τραγουδι παιζει η οχι

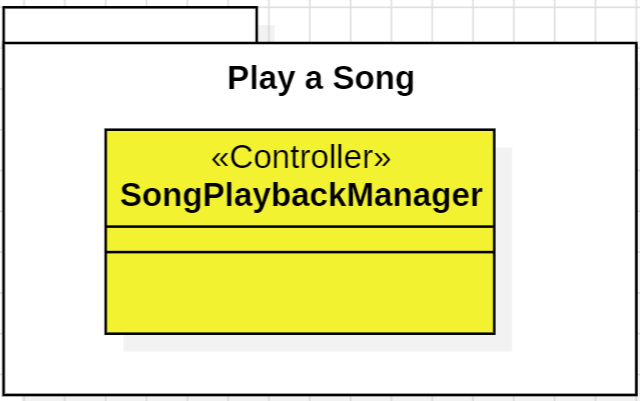
**Μεθοδοι**

1. **Song() :** Συναρτηση δομησης του αντικειμενου Song
2. **getTitle():** Συναρτηση που επιστρεφει τον τιτλο του τραγουδιου
3. **setTitle(title:String):** Θετει τον τιτλο του τραγουδιου
4. **getArtist():** Επιστρεφει το αντικειμενο του καλλιτεχνη στον οποιο ανηκει το τραγουδι
5. **setArtist(in artist:Artist):** Θετει τον καλλιτεχνη του τραγουδιου
6. **getDuration():** Επιστρεφει την διαρκεια του τραγουδιου
7. **setDuration(duration:Integer):** Θετει την διαρκεια του τραγουδιου
8. **getLiked():** Επιστρεφει τον αριθμο των <<likes>> που εχει το συγκεκριμενο τραγουδι
9. **setLiked( liked:Integer):** Ανανεωνει τον αριθμο των <<like>> του τραγουδιου
10. **getPlaylists():** Επιστρεφει μια λιστα απο αντικειμενα τυπου Playlist, στα οποια ο χρηστης εχει βαλει το συγκεκριμενο τραγουδι. (Αν το τραγουδι δεν ανηκει σε καποιο Playlist, τοτε επιστρεφεται μια κενη λιστα)
11. **setPlaylists(List<Playlist>):**Ανανεωνει τα Playlist στα οποια ανηκει το τραγουδι
12. **getIsPlaying():** Ενα flag που καθοριζει αν γινεται αναπαραγωγη του τραγουδιου η οχι
13. **setIsPlaylist(is\_playing: Boolean):** Ανανεωνει την τιμη της μεταβλητης is\_playing

#### 1.4.2 Διάγραμμα κλάσεων

****

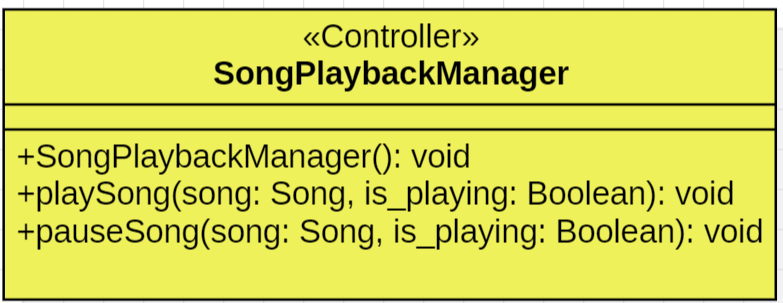
### 1.5 Play a Song: Πακέτο κλάσεων Gherkin Feature 5



### 1.5.1 Ορισμός Κλάσεων

##### 1.5.1.1. SongPlaybackManager

Η κλαση αυτη επικοινωνει με την SongPageGUI για τον συντονισμο της αναπαραγωγης ενος τραγουδιου



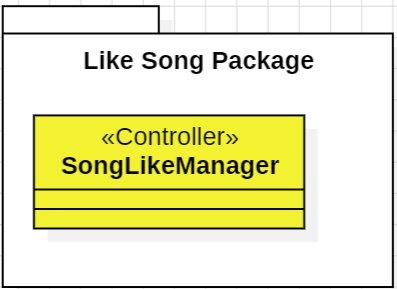
**Μεθοδοι**

1. **SongPlaybackManager():** Συναρτηση δομησης του ελεγκτη
2. **playSong(song, is\_playing):** Συναρτηση για την αναπαραγωγη του τραγουδιου εφοσον η Boolean μεταβλητη ειναι false
3. **pauseSong(song, is\_playing):** Συναρτηση για την παυση αναπαραγωγης του τραγουδιου εφοσον η Boolean μεταβλητη ειναι true

#### 1.5.2 Διάγραμμα κλάσεων

### 

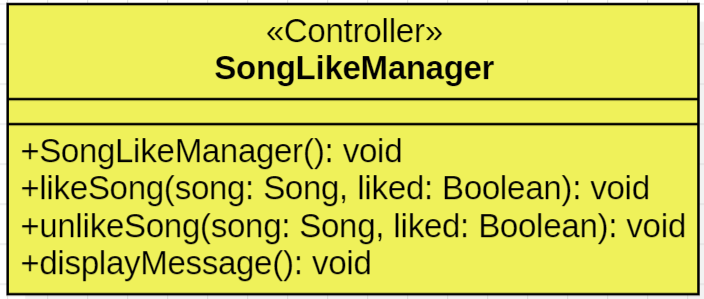
### 1. 6 Like a Song: Πακέτο κλάσεων Gherkin Feature 6



#### 1.6.1 Ορισμός Κλάσεων

##### 1.6.1.1 SongLikeManager

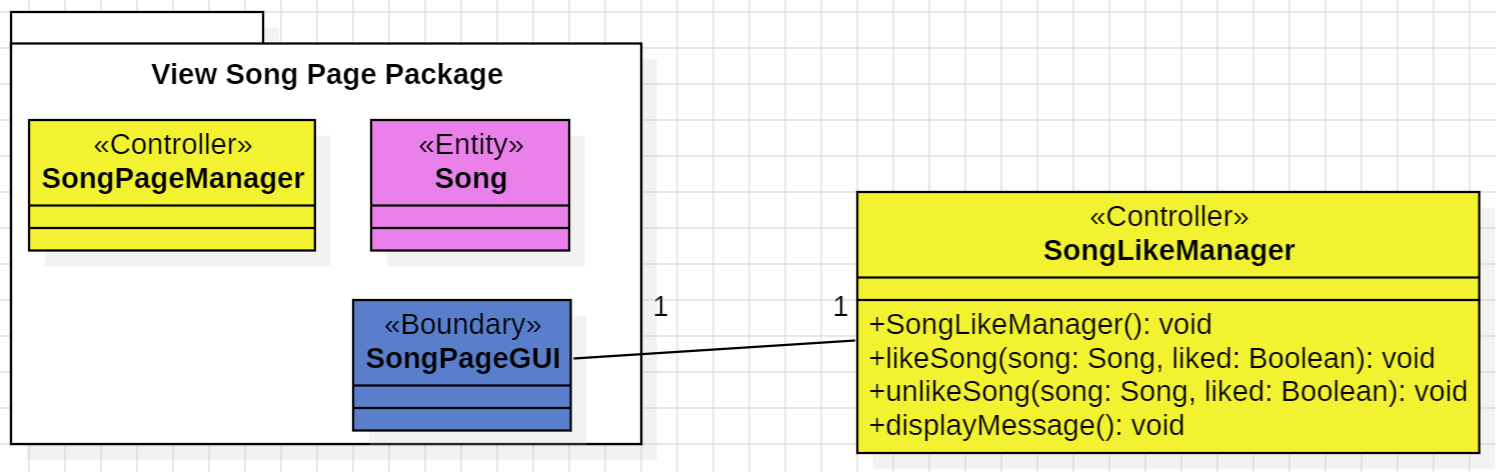
H κλαση αυτη επικοινωνει με την SongPageGUI για τον συντονισμο της διαδικασιας like ενος τραγουδιου απο τον χρηστη



**Μεθοδοι**

1. **SongLikeManager():** Συναρτηση δομησης της κλασης
2. **likeSong(song, liked):** Επισημαινει το τραγουδι ως liked απο τον χρηστη, εφοσον η μεταβλητη liked ειναι false και το προσθετει στην λιστα των liked τραγουδιων του χρηστη
3. **unlikeSong(sing, liked):** Επισημαινει το τραγουδι ως unliked απο τον χρηστη, εφοσον η μεταβλητη liked ειναι true και το αφαιρει απο την λιστα των liked τραγουδιων του χρηστη

#### 1.6.2 Διάγραμμα κλάσεων

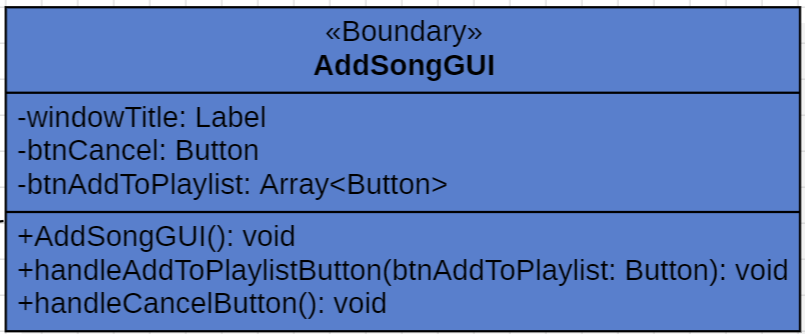
****

### 1.7 Add a Song to a Playlist: Πακέτο κλάσεων Gherkin Feature 7

#### 1.7.1 Ορισμός Κλάσεων

##### 1.7.1.1 AddSongGUI

H κλαση αυτη εκφραζει την διεπαφη του παραθυρου για την προσθηκη ενος τραγουδιου σε ενα απο τα playlist του χρηστη



**Χαρακτηριστικα**

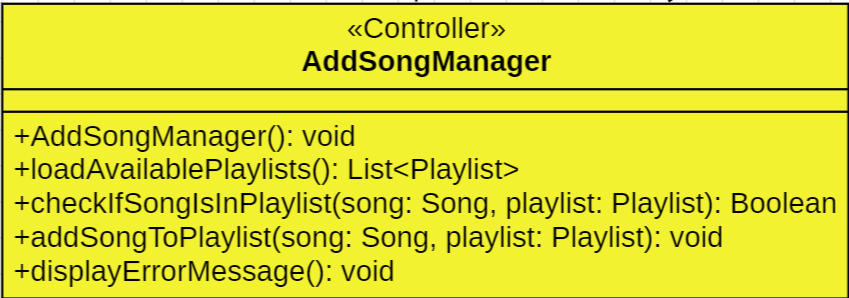
1. **windowTitle:** Μια ετικετα που γραφει το μηνυμα “Select playlist to add song into:”
2. **btnCancel:** Κουμπι για την επιστροφη στη σελιδα του τραγουδιου
3. **btnAddToPlaylist:** Λιστα απο κουμπια , ενα για καθε playlist του χρηστη, για να επιλεγει σε ποιο θα προσθεσει το τραγουδι

**Μεθοδοι**

1. **AddSongGUI:** Συναρτηση δομησης της κλασης
2. **handleAddToPlaylistButton(btnAddToPlaylist):** Καλει την συναρτηση checkIfSongIsInPlaylist() του AddSongManager και εφοσον επιστραφει False (το τραγουδι δεν βρισκεται ηδη στο playlist) καλειται η addSongToPlaylist(song, playlist) για να προστεθει το τραγουδι στο επιλεγμενο playlist, αλλιως εαν επιστραφει True, καλειται η displayErrorMessage()
3. **handleCancelButton():** Καλειται η συναρτηση displaySongDetails() του SongPageManager (Πακετο View Song Page) για να επιστρεψει ο χρηστης στην σελιδα του τραγουδιου

##### 1.7.1.2 ΑddSongManager

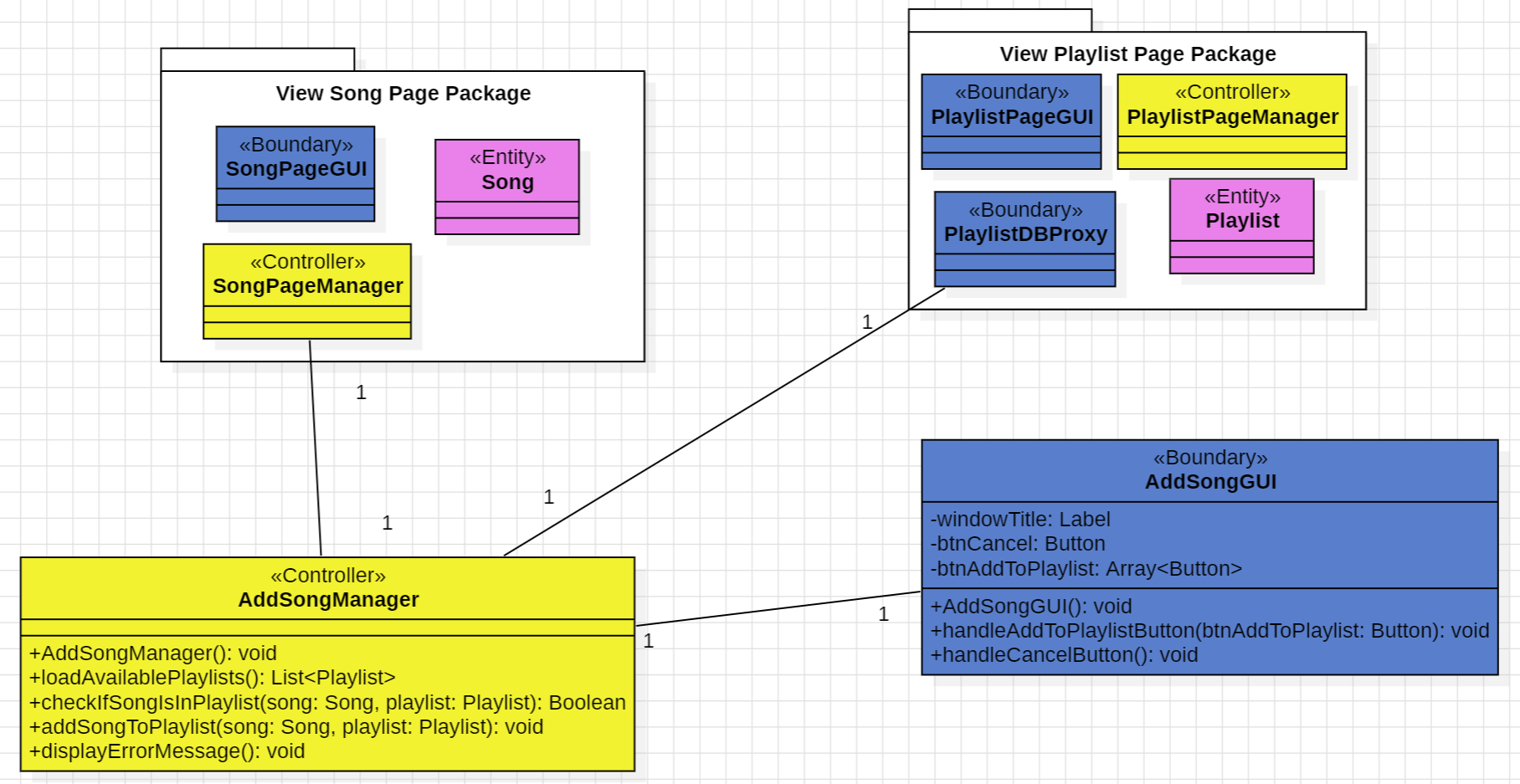
Η κλαση αυτη επικοινωνει με τις AddSongGUI, SongPageManager, PlaylistDBProxy και συντονιζει την διαδικασια προσθηκης ενος τραγουδιου σε playlist,



**Μεθοδοι**

1. **AddSongManager():** Συναρτηση δομησης της κλασης
2. **loadAvailablePlaylists():** Προβαλει στο παραθυρο προσθηκης τραγουδιου σε playlist τα διαθεσιμα playlist που εχει δημιουργησει ο χρηστης
3. **checkIfSongIsInPlaylist(song, playlist):** Ελέγχει αν το τραγουδι που της εχει δοθει βρισκεται μεσα στο playlist και επιστρεφει True η False αντιστοιχα
4. **addSongToPlaylist(song, playlist):** Προσθετει το τραγουδι στο playlist που εχει δοθει και το αποθηκευει στην βαση με την κληση της modifyExistingPlaylist(id, modifiedPlaylist) της PlaylistDBProxy (Πακετο View Playlist Page)
5. **displayErrorMessage():** Προβαλλει στην οθονη το μηνυμα “Song is already in this Playlist”

#### 1.7.2 Διάγραμμα κλάσεων

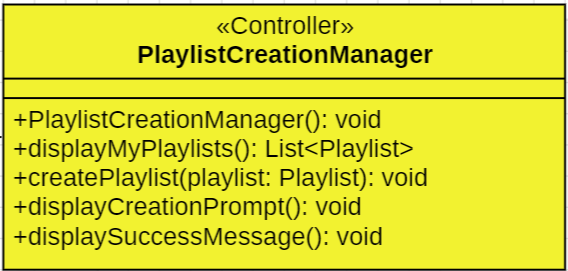
****

### 1.8 Create a Playlist: Πακέτο κλάσεων Gherkin Feature 8

#### 1.8.1 Ορισμός Κλάσεων

##### 1.8.1.1 PlaylistCreationManager

H κλαση αυτη επικοινωνει με τις MyPlaylistsPageGUI, PlaylistCreationToolGUI, PlaylistDBProxy και συντονιζει την διαδικασια δημιουργιας ενος playlist

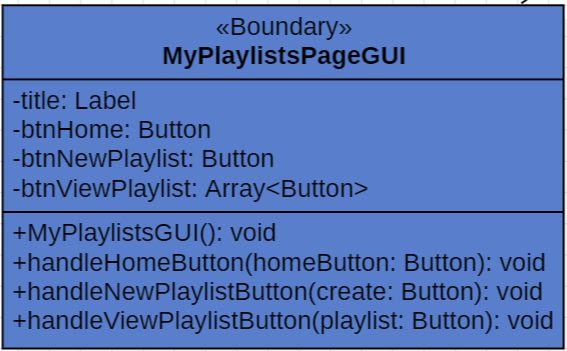


**Μεθοδοι**

1. **PlaylistCreationManager():** Συναρτηση δομησης του ελεγκτη
2. **displayMyPlaylists():** Επιστεφει απο την βαση και προβαλλει το παραθυρο MyPlaylists, με τα playlist που εχει δημιουργησει ο χρηστης
3. **createPlaylist(playlist):** Δημιουργει ενα αντικειμενο Playlist και το αποθηκευει στην βαση
4. **displayCreationPrompt():** Προβαλλει στον οθονη το παραθυρο PlaylistCreationTool
5. **displaySuccessMessage():** Προβαλλει το μηνυμα <<New Playlist Created>> στην οθονη

##### 1.8.1.2 MyPlaylistsPageGUI

H κλαση αυτη υλοποιει την διεπαφη της σελιδας My Playlists, που περιεχει τα δημιουργημενα playlist του χρηστη



**Χαρακτηριστικα**

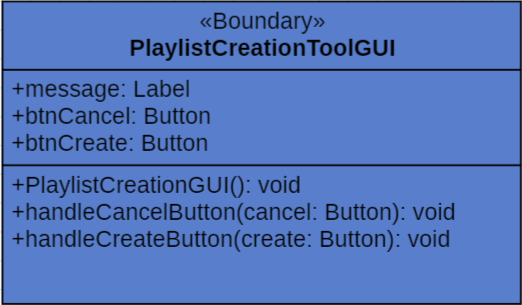
1. **title:** Ετικετα με τον τιτλο της σελιδας
2. **btnHome:** Κουμπι για την επιστροφη στην αρχικη οθονη
3. **btnNewPlaylist:** Κουμπι για την δημιουργια ενος καινουριου playlist
4. **btnViewPlaylist:** Μια λιστα απο κουμπια, ενα για καθε δημιουργημενο playlist του χρηστη, για την προβολη της σελιδας του playlist που επιλεγει ο χρηστης

**Μεθοδοι**

1. **MyPlaylistsGUI():** Συναρτηση δομησης της κλασης
2. **handleHomeButton(homeButton):** Συναρτηση διαχειρησης της χρησης του κουμπιου για επιστροφη στην αρχικη οθονη
3. **handleNewPlaylistButton(playlist):** Καλει την συναρτηση displayCreationPrompt() του PlaylistCreationManager()
4. **handleViewPlaylistButton():** Καλει την συναρτηση displayPlaylistDetails(id) του PlaylistPageManager (Πακετο View Playlist Page) για το playlist που εχει επιλεξει ο χρηστης

##### 1.8.1.3 PlaylistCreationToolGUI

H κλαση αυτη υλοποιει την διεπαφη του παραθυρου δημιουργιας ενος playlist



**Χαρακτηριστικα**

1. **message:** Ετικετα με το μηνυμα “Enter Playlist’s Name:”
2. **btnCancel:** Κουμπι ακυρωσης δημιουργιας playlist
3. **btnCreate:** Κουμπι επιβεβαιωσης δημιουργιας playlist

**Μεθοδοι**

1. **PlaylistCreationToolGUI():** Συναρτηση δομησης της κλασης
2. **handleCancelButton(cancel):** Καλει την συναρτηση displayMyPlaylists() του PlaylistCreationManager για την προβολη του παραθυρου MyPlaylists
3. **handleCreateButton():** Καλει την συναρτηση createPlaylist(playlist) για την δημιουργια ενος playlist με τα στοιχεια που εχουν δοθει ως input απο τον χρηστη, και επειτα καλει την displaySuccessMessage()

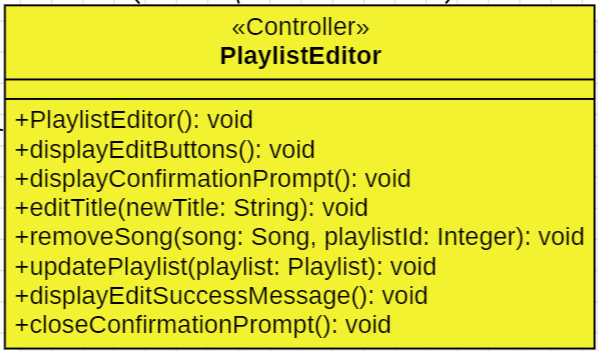
#### 1.8.2 Διάγραμμα κλάσεων

### 1.9 Edit a Playlist: Πακέτο κλάσεων Gherkin Feature 9

#### 1.9.1 Ορισμός Κλάσεων

##### 1.9.1.1 PlaylistEditor

H κλαση αυτη επικοινωνει με τις PlaylistPageGUI, PlaylistDBProxy, PlaylistEditGUI, ConfirmationPrompt και συντονιζει την διαδικασια επεξεργασιας ενος playlist

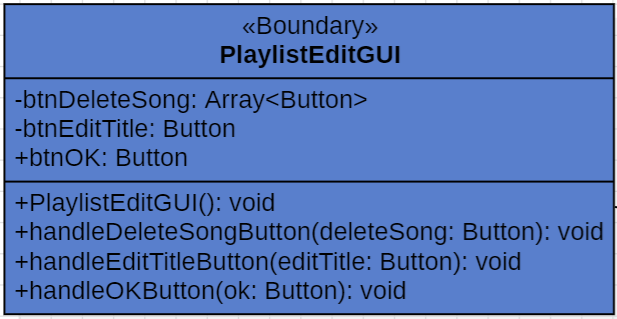


**Μεθοδοι**

1. **PlaylistEditor:** Συναρτηση δομησης της κλασης
2. **editTitle(newTitle):** Τροποποιει τον τιτλο του playlist και το αποθηκευει στην βαση με τη χρηση της updatePlaylist
3. **displayEditButtons():** Θετει την μεταβλητη editMode = True για το playlist, και προβαλλει στην οθονη τα κουμπια επεξεργασιας του playlist
4. **displayConfirmationPrompt():** Προβαλλει στην οθονη το παραθυρο Confirmation Prompt για την επιβεβαιωση των αλλαγων
5. **removeSong(song, playlistId):** Αφαιρει το δοσμενο τραγουδι απο το playlist που επεξεργαζεται ο χρηστης.
6. **updatePlaylist(playlist):** Aποθηκευει το τροποποιημενο playlist στη βαση με τη χρηση της playlistDBProxy (Πακετο View Playlist Page) και καλει την συναρτηση displayEditSuccessMessage()
7. **displayEditSuccessMessage():** Προβαλλει στην οθονη το μηνυμα “Playlist modified” μετα την επιτυχη επεξεργασια του playlist
8. **closeConfirmationPrompt():** Κλεινει το παραθυρο επιβεβαιωσης αλλαγων και επιστρεφει στο παραθυρο Playlist Page με την κληση της displayPlaylistDetails(id) του PlaylistPageManager

##### 1.9.1.2 PlaylistEditlGUI

H κλαση αυτη υλοποιει την διεπαφη του παραθυρου επεξεργασιας ενος playlist



**Χαρακτηριστικα**

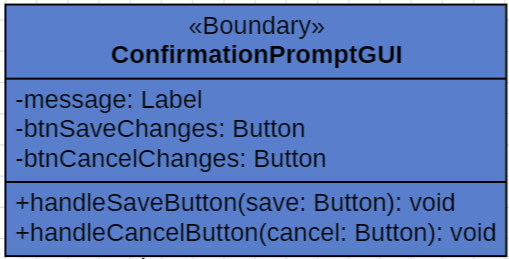
1. **btnDeleteSong:** Λιστα απο κουμπια, ενα για καθε τραγουδι του playlist, για την διαγραφη τους απο αυτο
2. **btnEditTitle:** Κουμπι για την επεξεργασια του τιτλου του playlist
3. **btnOK:** Κουμπι για την επιβεβαιωση των αλλαγων που εχουν γινει στο playist

**Μεθοδοι**

1. **PlaylistEditGUI():** Συναρτηση δομησης της κλασης
2. **handleDeleteSongButton(deleteSong):** Καλει την συναρτηση removeSong(song, playlistId του PlaylistEditor
3. **handleEditTitleButton(editTitle):** Καλει την συναρτηση editTitle(newTitle) του PlaylistEditor
4. **handleOKButton(ok):** Καλει την συναρτηση displayConfirmationPrompt() του PlaylistEditor

##### 1.9.1.3 ConfirmationPromptGUI

H κλαση αυτη υλοποιει την διεπαφη του παραθυρου επιβεβαιωσης αλλαγων που εχουν γινει στο playlist

****

**Χαρακτηριστικα**

1. **message:** Μια ετικετα με το μηνυμα “Save Changes?”
2. **btnSaveChanges:** Κουμπι για την αποθηκευση των αλλαγων που εχουν γινει στο playlist
3. **btnCancelChanges:** Κουμπι για την απορριψη των αλλαγων που εχουν γινει στο playlist

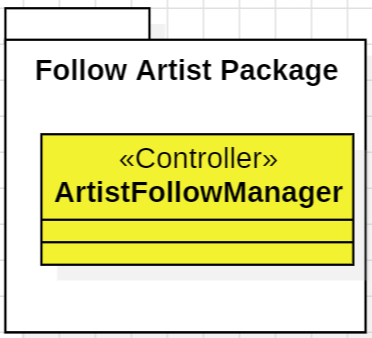
**Μεθοδοι**

1. **handleSaveButton(save):** Καλει την συναρτηση updatePlaylist() του PlaylistEditor για να αποθηκευτουν οι αλλαγες στη βαση
2. **handleCancelButton(cancel):** Καλει την συναρτηση closeConfirmationPrompt()

#### 1.9.2 Διάγραμμα κλάσεων

### 

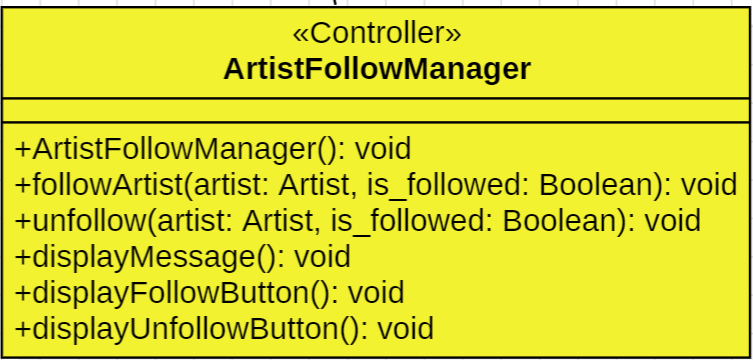
### 1.10 Follow Artists: Πακέτο κλάσεων Gherkin Feature 10



#### 1.10.1 Ορισμός Κλάσεων

##### 1.10.1.1 ArtistFollowManager

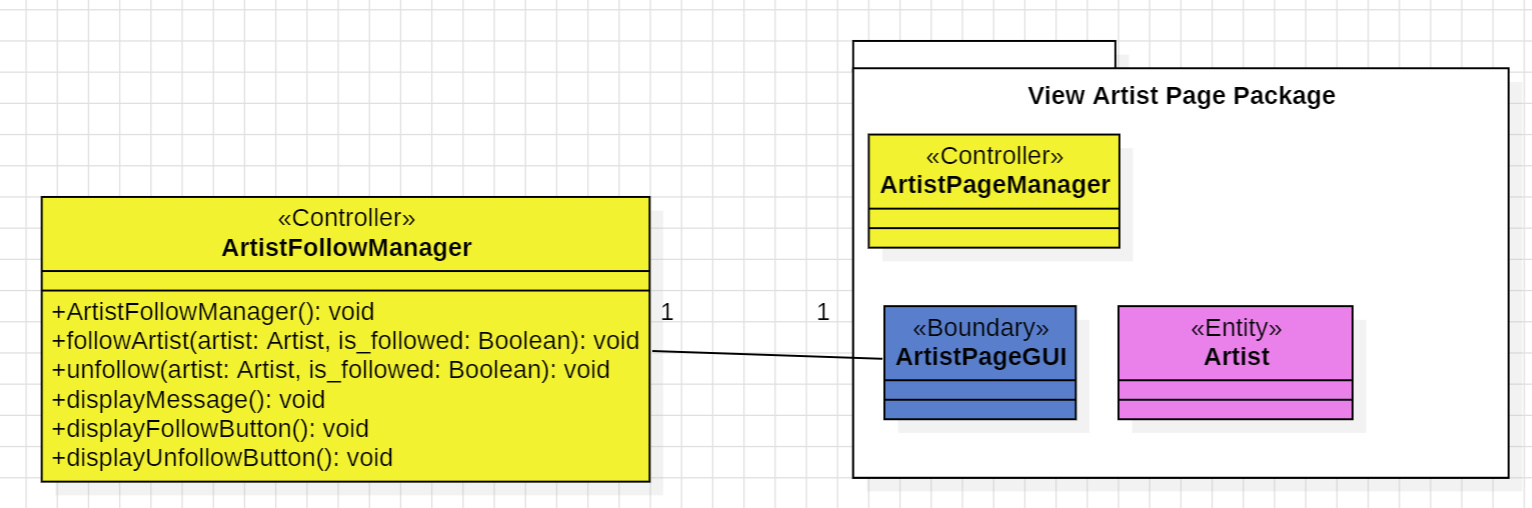
H κλαση αυτη επικοινωνει με την ArtistPageGUI και συντονιζει την διαδικασια follow ενος καλλιτεχνη απο τον χρηστη



**Μεθοδοι**

1. **ArtistFollowManager():** Συναρτηση δομησης της κλασης
2. **followArtist(artist):** Θετει την μεταβλητη του καλλιτεχνη is\_followed = True αυξανει κατα 1 την μεταβλητη followers και καλει την συναρτηση displayMessage(message = “Started Following Artist”)
3. **unfollowArtis(artist, is\_followed):** Θετει την μεταβλητη του καλλιτεχνη is\_followed = False, μειωνει κατα 1 την μεταβλητη followers και καλει την συναρτηση displayMessage(message = “Stopped Following Artist”)
4. **displayFollowButton(is\_followed):** Ελεγχει εαν ο καλλιτεχνης ακολουθειται ηδη απο τον χρηστη με την χρηση της Boolean μεταβλητης is\_followed. Αν οχι εμφανιζει στην οθονη το κουμπι Follow
5. **displayUnFollowButton(is\_followed):** Ελεγχει εαν ο καλλιτεχνης ακολουθειται ηδη απο τον χρηστη με την χρηση της Boolean μεταβλητης is\_followed. Αν ναι, εμφανιζει στην οθονη το κουμπι Unfollow
6. **displayMessage(message):** Εμφανιζει το μηνυμα του input στην οθονη.

#### 1.10.2 Διάγραμμα κλάσεων

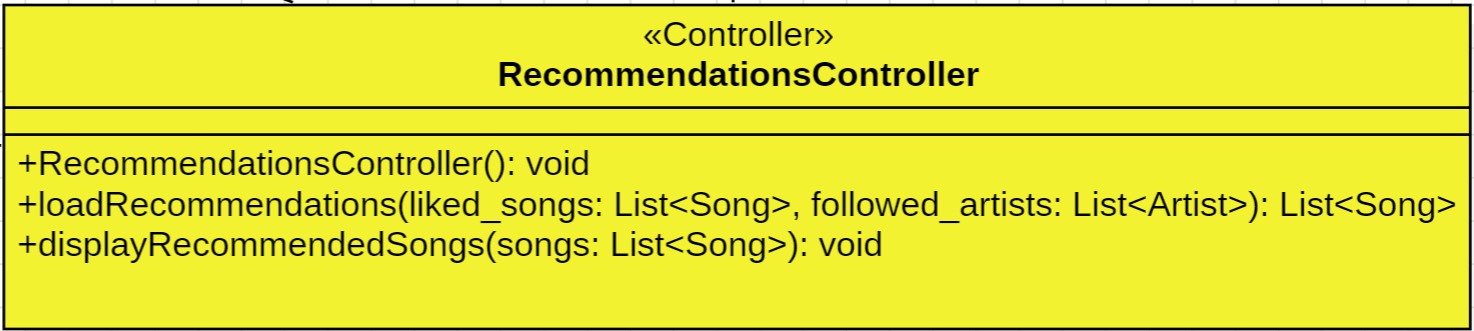
****

### 1.11 Get Song Recommendations: Πακέτο κλάσεων Gherkin Feature 11

#### 1.11.1 Ορισμός Κλάσεων

##### 1.11.1.1 RecommendationsController

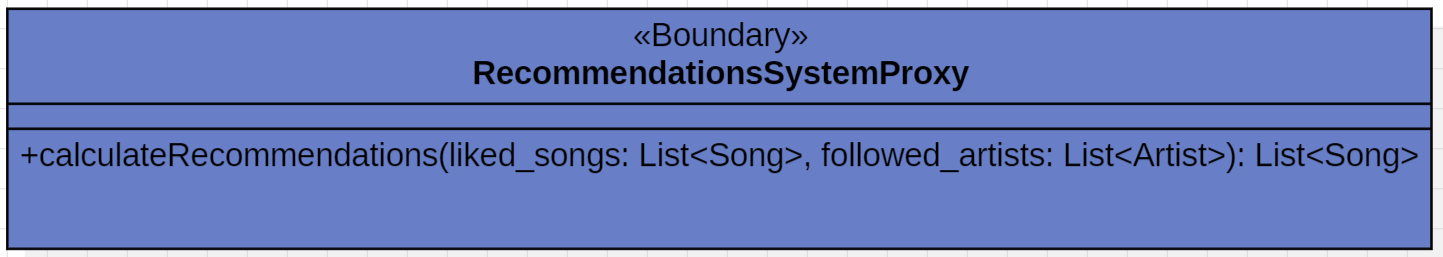
H κλαση αυτη επικοινωνει με την ΜyRecommendationsGUI, RecommendationsSystemProxy, Song και συντονιζει την διαδικασια υπολογισμου και εμφανισης στο παραθυρο Recommended Songs, των προτεινομενων τραγουδιων του χρηστη

****

**Μεθοδοι**

1. **RecommendationsController():** Συναρτηση δομησης του ελεγκτη
2. **loadRecomendations(liked\_songs, followed\_artists):** Συναρτηση που φορτωνει απο το συστημα προτιμησεων, τραγουδια για τον χρηστη, με βαση τα τραγουδια που εχει κανει like, και τους καλλιτεχνες που ακολουθει, με κληση της calculateRecommendations() του RecommendationsSystemProxy
3. **displayRecommendedSongs(songs):** Προβαλλει στην οθονη My Recommendations τα προτεινομενα τραγουδια του χρηστη

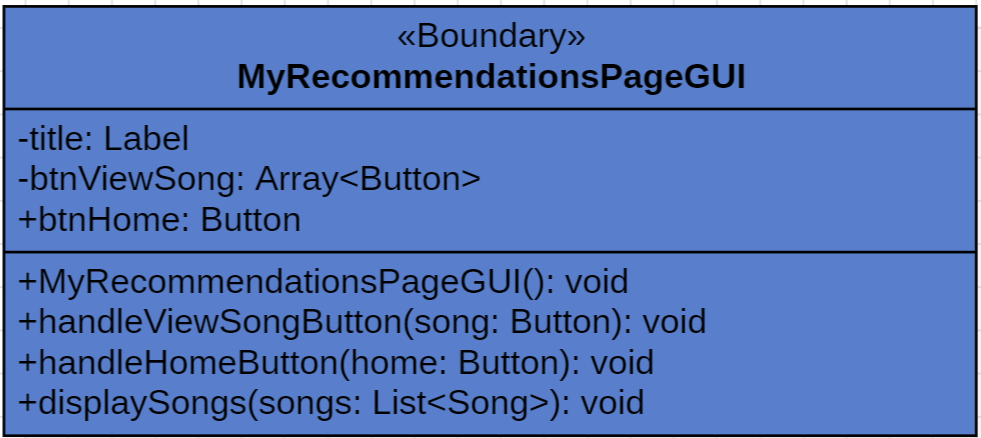
##### 1.11.1.2 RecommendationsSystemProxy

H κλαση αυτη υλοποιει την διεπαφη αναμεσα στο συστημα της εφαρμογης και στο συστημα υπολογισμου των προτεινομενων τραγουδιων****

**Μεθοδοι**

1. **calculateRecommendations(liked\_songs, followed\_artists):** Υπολογιζειτις προτιμησεις τραγουδιων για τον χρηστη με βαση τα input και επιστρεφει μια λιστα με προτεινομενα τραγουδια

##### 1.11.1.3 ΜyRecommendationsPageGUI

H κλαση αυτη υλοποιει την διεπαφη του παραθυρου της σελιδας Recommended Songs****

**Χαρακτηριστικα**

1. **title:** Μια ετικετα με τον τιτλο
2. **btnViewSong:** Μια λιστα απο κουμπια που αντιπροσωπευουν προτεινομενα τραγουδια για τον χρηστη
3. **btnHome:**  Κουμπι επιστροφης στην αρχικη οθονη

**Μεθοδοι**

1. **MyRecommendationsPageGUI():** Συναρτηση δομησης της κλασης
2. **handleViewSongButton(viewSong):** Καλειται η συναρτηση retreiveSong(id) του SongPageManager (Πακετο View Song Page) για την ανακτηση του τραγουδιου με το συγκεκριμενο id απο την βαση και επειτα η displaySongDeatails(Song) για την προβολη της σελιδας του τραγουδιου που επιλεχθηκε.
3. **handleHomeButton(home: Button):** Συναρτηση διαχειρησης της χρησης του κουμπιου για επιστροφη στην αρχικη οθονη
4. **displaySongs(songs):** Προβαλλει τα προτεινομενα τραγουδια στην οθονη

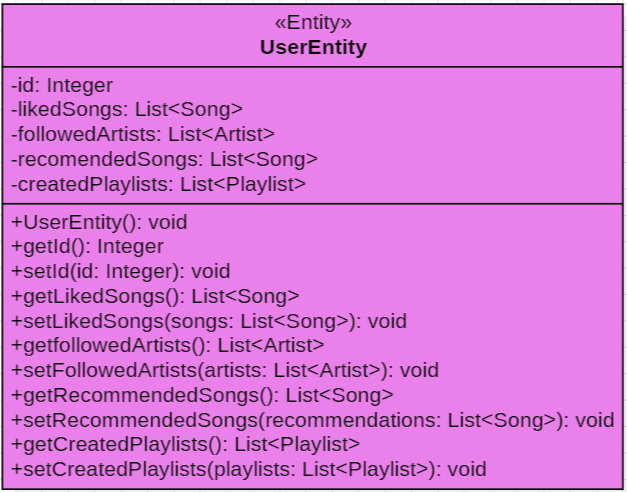
#### 1.11.2 Διάγραμμα κλάσεων

### 1.12 Navigate to Homepage: Πακέτο κλάσεων Gherkin Feature 12\*

#### 1.12.1 Ορισμός Κλάσεων

##### 1.12.1.1 UserEntity

Η κλαση αυτη περιεχει τα δεδομενα της οντοτητας ενος χρηστη της εφαρμογης



**Χαρακτηριστικα**

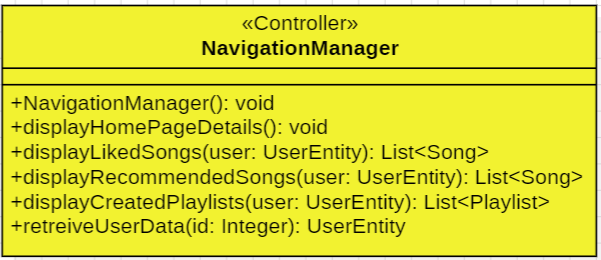
1. **id:** Αποτελεί τον μοναδικό χαρακτηριστικό ακέραιο αριθμό του κάθε χρηστη
2. **likedSongs:** Μια λιστα με τα τραγουδια που εχει κανει like ο χρηστης
3. **followedArtists:** Μια λιστα με τους καλλιτεχνες που εχει κανει follow ο χρηστης
4. **recommendedSongs:** Μια λιστα με τα προτεινομενα τραγουδια για τον χρηστη
5. **createdPlaylists:** Μια λιστα με τα playlists που εχει δημιουργησει ο χρηστης

**Μεθοδοι**

1. **UserEntity():** Αποτελει συναρτηση δομησης της κλασης
2. **getId():** Επιστρεφει το id του χρηστη
3. **setId(id):** Τροποποιει το id του χρηστη
4. **getLikedSongs():** Επιστρεφει την λιστα με τα τραγουδια που εχει κανει like ο χρηστης
5. **setLikedSongs(List<Song>):** Τροποποιει την λιστα με τα τραγουδια που εχει κανει like ο χρηστης
6. **getFollowedArtists():** Επιστρεφει την λιστα με τους καλλιτεχνες που εχει κανει follow ο χρηστης
7. **setFollowedArtists(List<Artist>):** Τροποποιει την λιστα με τους καλλιτεχνες που εχει κανει follow ο χρηστης
8. **getRecommendedSongs():** Επιστρεφει την λιστα με τα προτεινομενα τραγουδια για τον χρηστη
9. **setRecommendedSongs(List<Song>):** Τροποποιει την λιστα με τα προτεινομενα τραγουδια για τον χρηστη
10. **getCreatedPlaylists():** Επιστρεφει την λιστα με τα playlists που εχει δημιουργησει ο χρηστης
11. **setCreatedPlaylist(List<Playlist>):** Τροποποιει την λιστα με τα playlists που εχει δημιουργησει ο χρηστης

##### 1.12.1.2 NavigationManager

Η κλαση αυτη συντονιζει την πλοηγηση του χρηστη στην εφαρμογη



**Μεθοδοι**

1. **NavigationManager():** Συναρτηση δομησης της κλασης
2. **displayHomePageDetails():** Εμφανιζει την σελιδα της αρχικης οθονης
3. **displayLikedSongsPage(user):** Εμφανιζει την σελιδα Liked Songs
4. **displayRecommendedSongsPage(user):** Εμφανιζει την σελιδα Recommended Song
5. **displayMyPlaylistsPage(user):** Εμφανιζει την σελιδα My Playlists
6. **retreiveUserData(id):** Επικοινωνει με την UserDBProxy για να ανακτησει τα δεδομενα του χρηστη με βαση το id του

##### 1.12.1.3 ΗοmePageGUI

Η κλαση αυτη εκφραζει την διεπαφη του παραθυρου της αρχικης οθονης



**Χαρακτηριστικα**

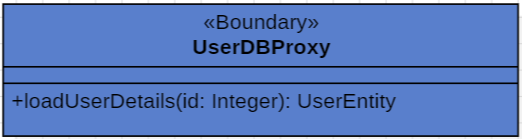
1. **title:** Μια ετικετα με τον τιτλο της εφαρμογης
2. **logo:** Μια εικονα με το λογοτυπο της εφαρμογης
3. **btnSearch:** Κουμπι για την πλοηγηση στο παραθυρο Search
4. **btnLikeSongs:** Κουμπι για την πλοηγηση στο παραθυρο Liked Songs
5. **btnMyPlaylists:** Κουμπι για την πλοηγηση στο παραθυρο My Playlists
6. **btnRecommendedSongs:** Κουμπι για την πλοηγηση στο παραθυρο Recommended Songs

**Μεθοδοι**

1. **HomePageGUI():** Συναρτηση δομησης της κλασης
2. **handleSearchButton(search):** Καλει πρωτα την συναρτηση retreiveUserData(id) και μετα την displayHomePageDetails()
3. **handleLikedSongsButton(liked):** Καλει πρωτα την συναρτηση retreiveUserData(id) και μετα την displayLikedSongsPage(user)
4. **handleMyPlaylistsButton(myPlaylists):** Καλει πρωτα την συναρτηση retreiveUserData(id) και μετα την displayMyPlaylistsPage(user)
5. **handleRecommendedSongsButton(recommended):** Καλει πρωτα την συναρτηση retreiveUserData(id) και μετα την displayRecommendedSongsPage(user)

##### 1.12.1.4 UserDBProxy

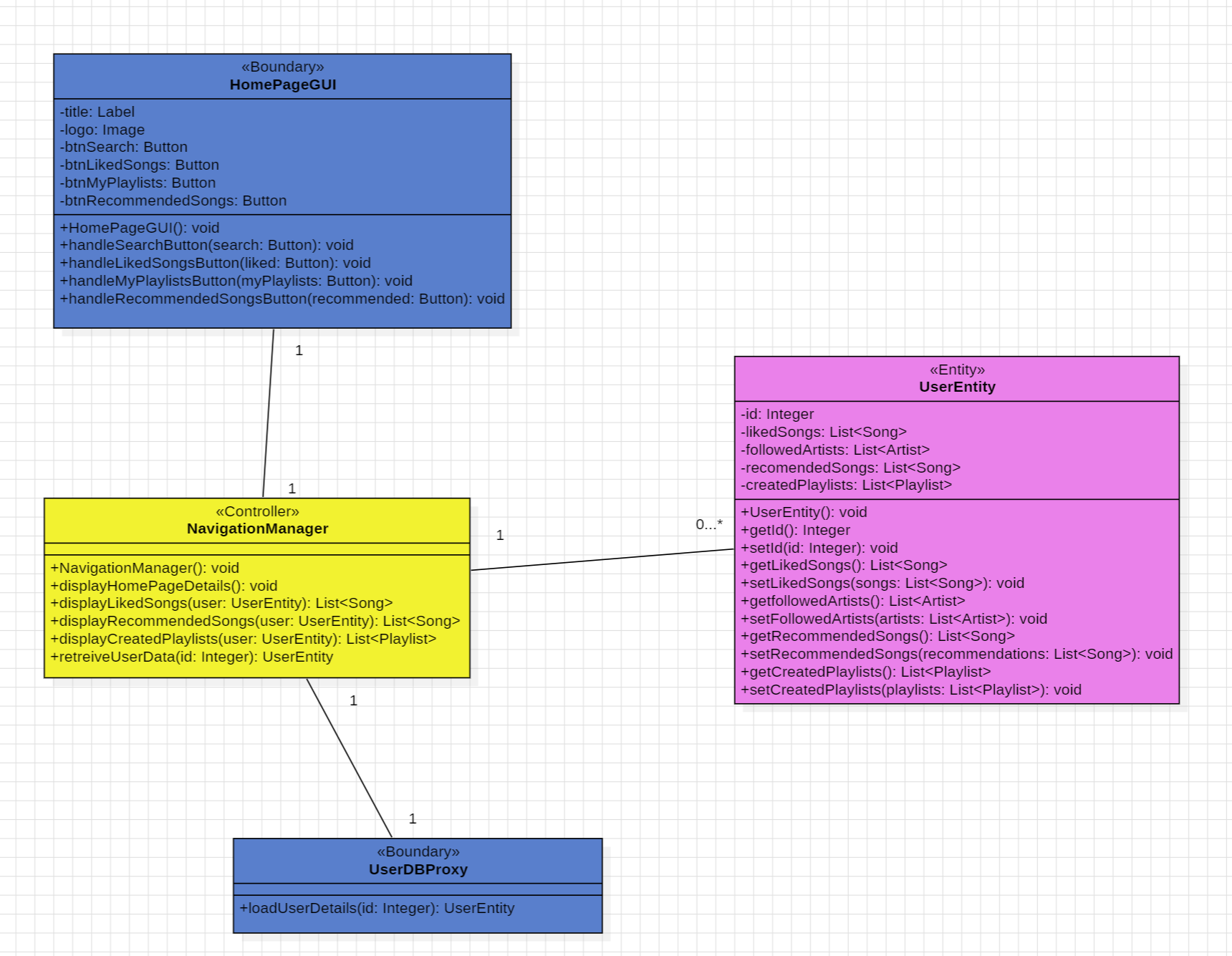
Η κλαση αυτη επικοινωνει με την βαση δεδομενων για την αναζητηση λεπτομερειων των χρηστων

****

**Μεθοδοι**

1. **loadUserDetails(id):** Αναζητα στη βαση και επιστρεφει το αντικειμενο UserEntity με το δοσμενο id

#### 1.12.2 Διάγραμμα κλάσεων

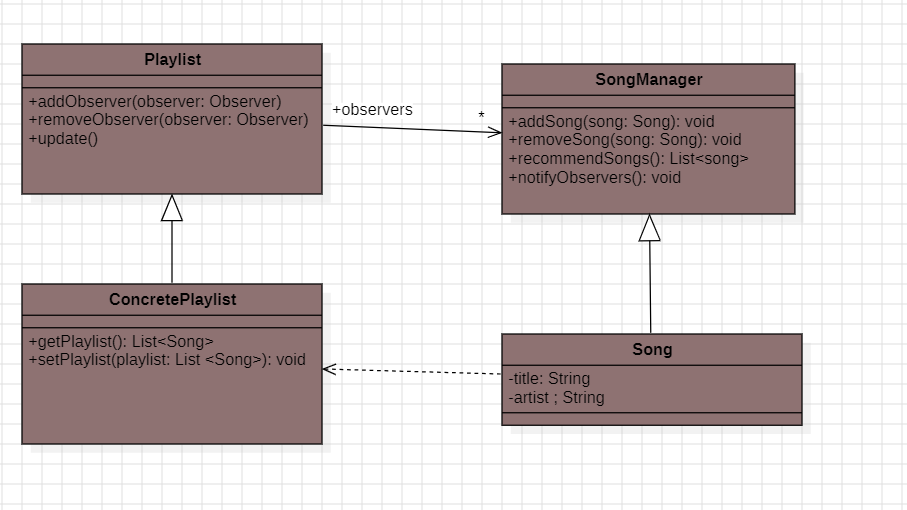


# 2 Πρότυπα Σχεδιασμού που υιοθετήθηκαν

## 2.1 Πρότυπο Observer

Το **πρότυπο σχεδίασης Observer (συμπεριφορικό πρότυπο)** χρησιμοποιείται για να επιτρέπει τη συνεργασία μεταξύ αντικειμένων και πιο συγκεκριμένα την αλληλεπίδραση ένα-προς-πολλά αντικείμενα, με τέτοιο τρόπο ώστε όταν ένα αντικείμενο αλλάζει κατάσταση, τα υπόλοιπα στοιχεία της σύνδεσης να “ενημερώνονται” αυτόματα και ταυτόχρονα.

Mε βάση τη *Μη Λειτουργική Απαίτηση 4 (το σύστημα* θα πρέπει να μπορεί να παραμένει αποδοτικό, παρά το μεγάλο αριθμό χρηστών και τραγουδιών*)* το πρότυπο Observer μπορεί να αξιοποιηθεί για την ενημέρωση και τη διαχείριση διαφόρων στοιχείων της εφαρμογής ως απόκριση στις αλλαγές σε μια σειρά δεδομένα π.χ προτεινόμενα τραγούδια με βάση προσθήκες ή αφαιρέσεις σε playlists.

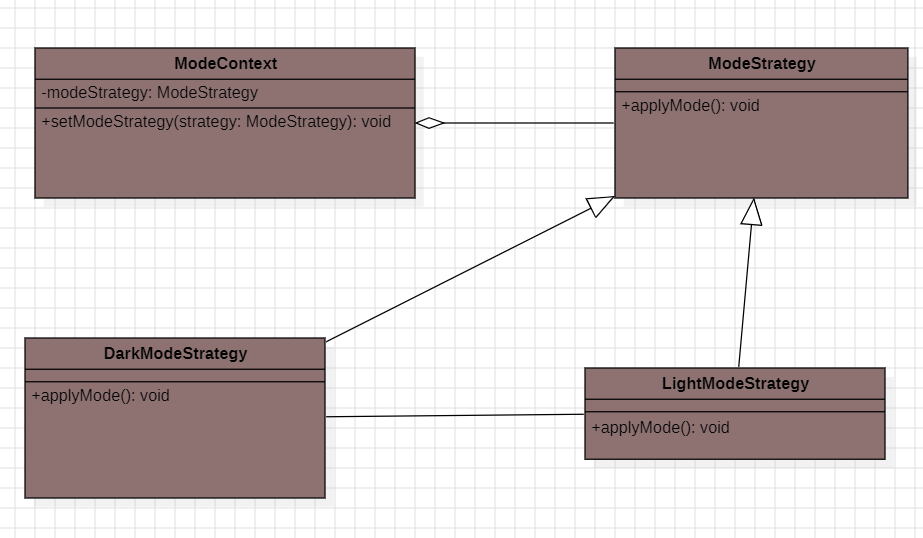


*Εφαρμογή προτύπου Observer*

## 2.2 Πρότυπο Strategy

Το **πρότυπο σχεδίασης Strategy (συμπεριφορικό πρότυπο)** ορίζει μια συλλογή αλγορίθμων, αξιοποιώντας τον καθένα, με εναλλάξιμη χρήση. Αυτό επιτρέπει στην εφαρμογή να διαφοροποιεί λειτουργίες του, βάσει και των προτιμήσεων των χρηστών.

Mε βάση τη *Μη Λειτουργική Απαίτηση 5 (Το σύστημα πρέπει να έχει εύχρηστο και εύληπτο περιβάλλον διεπαφής)* και πιο συγκεκριμένα τη δυνατότητα εναλλαγής dark/light mode,το πρότυπο Strategy μπορεί να αξιοποιηθεί για αυτό το σκοπό, επιτρέποντας την εύκολη εναλλαγή μεταξύ τους.



*Εφαρμογή Προτύπου Strategy*

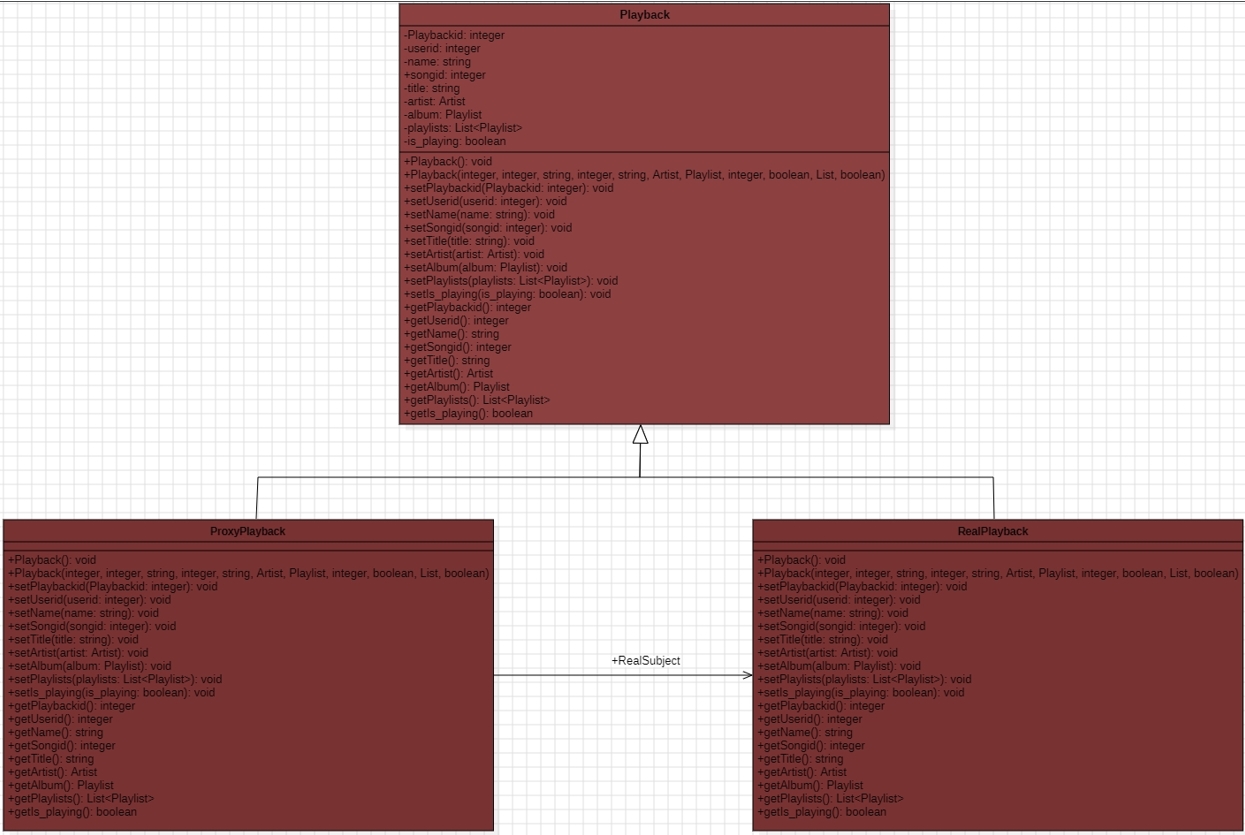
## 

## 

## 

## 2.3 Πρότυπο Proxy

Το **πρότυπο σχεδίασης Proxy** είναι ένα δομικό πρότυπο. Στην εφαρμογή μας, AudioSync, είναι σημαντική η προστασία των χρηστών της βάσης από υποκλοπή των δεδομένων καθώς και η ομαλή λειτουργία της εφαρμογής. Με την χρήση αυτου το προτύπου, καλύπτουμε τις **ΜΛΑ-3** και **ΜΛΑ-4.** Ειδικότερα, για την **ΜΛΑ-3,** θα πρέπει το σύστημα να δέχεται εντολές μόνο από ταυτοποιημένους χρήστες ή καλλιτέχνες έτσι ώστε να υπακούει στους περιορισμούς ασφαλείας κατα GDPR. Επίσης για την αποδοτικότητα του συστήματος , για την **ΜΛΑ-4**, όταν υπάρχουν πολλοί χρήστες είναι απαραίτητο να εξυπηρετούνται χωρίς καθυστερήσεις. Συνεπώς, τα υλοποιούμε μέσω ελέγχου. Γι΄αυτό χρησιμοποιούμε στην κλάση Playback ένα **protection** proxy για την ασφάλεια του χρήστη το οποίο μπορεί να υλοποιηθεί και ως **virtual** proxy για να φορτώνει ένα προσωρινό αντικείμενο (song), μέχρι να δοθεί πρόσβαση στο πραγματικό περιεχόμενο του αντικειμένου. Με αυτόν τον τρόπο προστατεύουμε τους χρήστες και τους καλλιτέχνες όπως επίσης και το σύστημα από overflow.



*Εφαρμογή Προτύπου Proxy*

# 3 Δυναμική Μοντελοποίηση

**3.1 Πακέτο ακρόασης ενός τραγουδιού από τον χρήστη**

Το πακέτο αυτό περιλαμβάνει την αρχική οθόνη που βλέπει ο χρήστης αφού εισέλθει στην εφαρμογή. Ο χρήστης μετά μπορεί να επιλέξει “Search” για να μπει στην σελίδα αναζήτησης και να πληκτρολογήσει το όνομα του τραγουδιού που θέλει να ακούσει.

Οι λειτουργικές απαιτήσεις για αυτό το πακέτο χρήσης είναι οι εξής:

* <ΛΑ-1 > Ο χρήστης πρέπει να μπορεί να κάνει αναζήτηση τραγουδιών και καλλιτεχνών.
* <ΛΑ-2> Ο χρήστης πρέπει να μπορεί να ακούει τραγούδια που επιλέγει.

**3.1.1 Αφήγηση Σεναρίου**

Ο χρήστης έχει ανοίξει την εφαρμογή. Του εμφανίζεται η αρχική οθόνη.Ο χρήστης επιλέγει “Enter” και εισέρχεται στην κεντρική οθόνη της εφαρμογής HomePageGUI. Σε αυτο το σημείο , ο χρήστης θα πρέπει να επιλέξει το κουμπί “Search”. Αφου το επιλέξει, εισέρχεται στο μενού αναζήτησης. Αν ο χρήστης επιλέξει “Songs” ή “Artists” χωρίς να έχει εισάγει τίποτα, ή αν δεν υπάρχει το συγκεκριμένο τραγούδι ή ο καλλιτέχνης, θα καλεστεί η συνάρτηση displayErrorMessage(). Σε περίπτωση ορθής συμπλήρωσης, ο χρήστης αφού πατήσει μια απο τις δύο επιλογές θα καλεστεί είτε η loadSongByName() είτε η loadArtistByName(). Στο δικό μας σενάριο ο χρήστης επιλέγει να ακούσει ένα τραγούδι που αναζήτησε. Στη συνέχεια αφού του εμφανίστηκε μια λίστα με τα ονόματα των αντικειμένων τύπου songs: List[Song] αυτός, επιλέγει ένα. Τώρα, καλείται η songPageManager() και βλέπει ο χρήστης το SongPageGUI. Σε αυτό το σημείο ο χρήστης μπορεί να ακούσει το τραγούδι, να το βάλει στα αγαπημένα, να το προσθέσει σε κάποια playlist ή να επιστρέψει στην αρχική οθόνη. Στο σενάριο μας, ο χρήστης επιλέγει να ακούσει το τραγούδι, ελέγχεται αν το τραγούδι ακούγεται ή όχι, μέσω της is\_playing(), και αφού ακούσει το τραγούδι επιλέγει να πάει στην αρχική οθόνη HomePageGUI μέσω της συνάρτησης HandleHomeButton().

**3.2 Πακέτο δημιουργίας playlist απο τον χρήστη**

Το πακέτο αυτό περιλαμβάνει τη σελίδα δημιουργίας και τη σελίδα τροποποίησης των playlist που βλέπει ο χρήστης. Ο χρήστης μπορεί να επιλέξει να δημιουργήσει μια καινούργια playlist ή να ακούσει/τροποποιήσει μια υπάρχουσα playlist.

Οι λειτουργικές απαιτήσεις για αυτό το πακέτο χρήσης είναι οι εξής:

* <ΛΑ-5 > Ο χρήστης πρέπει να μπορεί να δημιουργεί δικές του playlists.
* <ΛΑ-6> Το σύστημα πρέπει να παρέχει πληροφορίες σχετικά με playlists που επιλέγει ο χρήστης.
* <ΛΑ-7> Ο χρήστης πρέπει να μπορεί να προσθέτει τραγούδια σε playlist που έχει δημιουργήσει.
* <ΛΑ-8> Ο χρήστης πρέπει να μπορεί να τροποποιεί τις playlists του.

**3.2.1 Αφήγηση Σεναρίου**

Ο χρήστης βρίσκεται στο HomePageGUI και έχει επιλέξει “My Playlists”. Εμφανίζεται η οθόνη PlaylistPageGUI.Ο χρήστης επιλέγει “Create” και εμφανίζεται η οθόνη PlaylistCreationToolGUI και καλείται η συνάρτηση displayCreationPrompt() όπου πρέπει να συμπληρώσει το όνομα που θέλει να δώσει στην playlist.Αν είναι σωστο το input του χρήστη τότε εμφανίζεται το μήνυμα της συνάρτησης displaySuccessMessage().Και έτσι έχει δημιουργήσει μια playlist.Αφού το κάνει αυτό , επιστρέφει στο PlaylistPageGUI όπου επιλέγει τώρα την playlist που έχει δημιουργήσει.Με βάση το σενάριο 3.1.1 υποθέτουμε ότι έχει ακούσει τραγούδια και έχει επιλέξει μέσω του ελεγκτή AddSongManager να προσθέσει κάποια τραγούδια στην playlist που έφτιαξε.Τώρα ο χρήστης βρίσκεται στη σελίδα PlaylistPageGUI και θέλει να τροποποιήσει μια playlist που έφτιαξε.Αφού πατήσει πάνω στην playlist που τον ενδιαφέρει , εμφανίζεται η οθόνη MyPlaylistPageGUI της playlist που επέλεξε.Πατώντας το κουμπί “Edit” του εμφανίζεται η οθόνη PlaylistEditGUI.Σε αυτό το σημείο ο χρήστης επιλέγει να αλλάξει το όνομα της playlist και να διαγράψει κάποια από τα τραγούδια που πρόσθεσε.Σε αυτό το σημείο αφού τελειώσει με τις αλλαγές επιλέγει το “OK” button και μέσω της handleOKButton(ok) πηγαίναι στο παράθυρο ConfimationPromptGUI όπου του εμφανίζεται το μήνυμα “Save Changes?” και επιλέγει το btnSaveChanges και οδηγείται πάλι στο PlaylistPageGUI.

**3.3 Πακέτο επισκόπησης προτεινόμενων τραγουδιών από τον χρήστη**

Το πακέτο αυτό περιλαμβάνει τη σελίδα με τα τραγούδια που το σύστημα προτείνει στο χρήστη βάσει των μουσικών επιλογών του. Ο χρήστης θα μπορεί να βλέπει τα τραγούδια που προτείνονται έτσι ώστε να επιλέγει αν θα τα κάνει “like”.

Οι λειτουργικές απαιτήσεις για αυτό το πακέτο χρήσης είναι οι εξής:

* <ΛΑ-4 > Ο χρήστης πρέπει να μπορεί να κάνει «like» σε τραγούδια.
* <ΛΑ-10> Το σύστημα πρέπει να παρέχει στο χρήστη προτεινόμενα τραγούδια.

**3.3.1 Αφήγηση Σεναρίου**

Ο χρήστης βρίσκεται στην κεντρική οθόνη του συστήματος. Επιλέγει το κουμπί «Recommended Songs» και μεταβαίνει στην ΜyRecommendationsGUI οθόνη. Έπειτα ανάλογα με τις προτιμήσεις του μπορεί να επιλέξει να αναπαράγει ένα τραγούδι από τις προτάσεις, χρησιμοποιώντας την playSong() συνάρτηση, ενώ επιπλέον μπορεί να αποθηκεύσει ως αγαπημένο ένα τραγούδι από τα προτεινόμενα με τη χρήση της likeSong() συνάρτησης.