

Swetify

Niccolò Malgeri, Filippo Viti, Alessio Delli Colli

July 2024

Contents

1	Motivazioni	3
2	Requisiti Funzionali	3
2.1	Funzionalità proposte al cliente	3
2.2	Funzionalità proposte all'artista	3
2.3	Diagramma dei casi d'uso	4
2.4	Template dei casi d'uso	4
2.5	Mockups	6
2.5.1	Pagina di accesso	6
2.5.2	Pagina di ricerca	6
2.5.3	Suggerimenti	7
2.5.4	Coda di riproduzione	7
2.6	Sequence diagram	8
3	Scelte di progetto	8
4	Documentazione	9
4.1	Domain model	9
4.1.1	Entity	9
4.1.2	Customer	9
4.1.3	Artist	9
4.1.4	Track	10
4.1.5	Song	10
4.1.6	Podcast	10
4.1.7	Playlist	10
4.1.8	Album	10
4.1.9	Suggestion	10
4.1.10	PlaybackQueue	10
4.2	Business logic	11
4.2.1	Handler	11
4.2.2	State	11
4.2.3	NavigationManager	11
4.2.4	AlbumsHandler	11

4.2.5	ArtistInfoHandler	11
4.2.6	HomeHandler	11
4.2.7	LoginHandler	12
4.2.8	PlaybackHandler	12
4.2.9	PlaylistHandler	12
4.2.10	RegistrationHandler	12
4.2.11	SearchHandler	12
4.2.12	SuggestionsHandler	12
4.2.13	UserPlaylistsHandler	12
4.3	Dao	12
4.3.1	Dao	12
4.3.2	CustomerDao	13
4.3.3	ArtistDao	13
4.3.4	SongDao	13
4.3.5	PodcastDao	13
4.3.6	PlaylistDao	13
5	Test effettuati	13

1 Motivazioni

Il nostro intensivo utilizzo di piattaforme di streaming musicali ha suscitato in noi un interesse riguardo la loro struttura e il desiderio di replicarne il funzionamento.

Abbiamo deciso quindi di realizzare un'applicazione che simuli le loro funzionalità da noi denominata Swetify.

2 Requisiti Funzionali

Swetify prevede la partecipazione di due tipologie di utenti: cliente e artista.

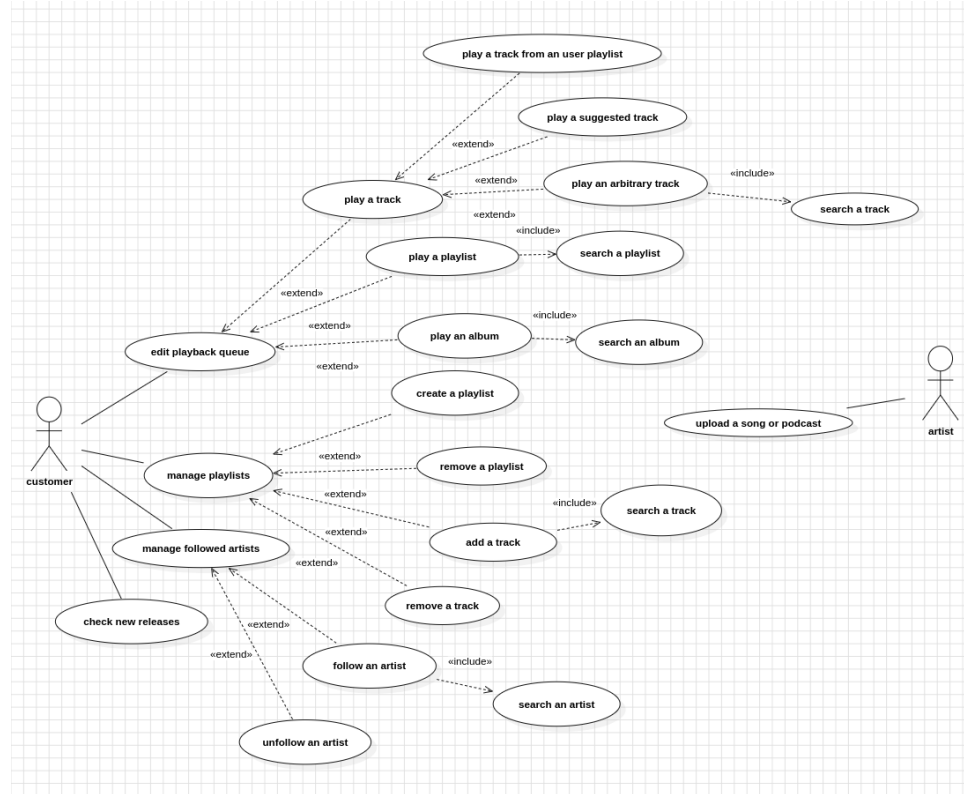
2.1 Funzionalità proposte al cliente

- visualizzare un catalogo musicale che consenta agli utenti di cercare brani, album e artisti tramite una barra di ricerca.
- visualizzare le informazioni dettagliate di un brano, inclusi titolo, artista, album e durata.
- riprodurre, mettere in pausa e saltare i brani.
- creare, modificare ed eliminare le proprie playlist, aggiungere e rimuovere brani da queste playlist.
- ricevere raccomandazioni di brani basate sulla cronologia di ascolto dell'utente
- seguire gli artisti per ricevere aggiornamenti sui nuovi rilasci.

2.2 Funzionalità proposte all'artista

- caricare album contenenti canzoni o podcast.

2.3 Diagramma dei casi d'uso



2.4 Template dei casi d'uso

UC1	riproduzione canzone
livello	user goal
descrizione	l'utente cerca e riproduce una canzone
attori	cliente
pre-condizioni	l'utente deve avere le credenziali per effettuare l'accesso
post-condizioni	la canzone selezionata viene aggiunta alla coda
normale svolgimento	1) l'utente apre la pagina di ricerca 2) inserisce il nome di una canzone 3) seleziona una voce dall'elenco proposto 4) seleziona l'opzione "aggiungi in coda"
svolgimenti alternativi	4b) l'utente seleziona l'opzione "aggiungi in testa"

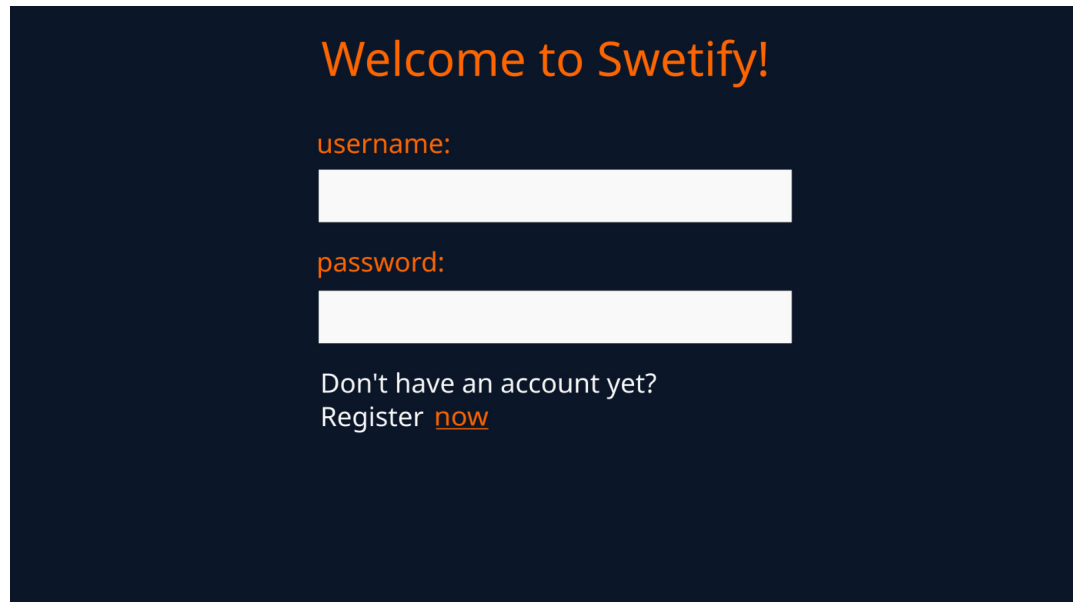
UC2	modifica playlist
livello	user goal
descrizione	l'utente modifica una delle sue playlist personali
attori	cliente
pre-condizioni	l'utente deve avere le credenziali per effettuare l'accesso ed avere una playlist salvata
post-condizioni	la playlist presenta i cambiamenti apportati dall'utente
normale svolgimento	1) l'utente apre la pagina "le mie playlist" 2) l'utente seleziona la playlist da modificare 3) l'utente cerca una canzone da aggiungere 4) l'utente termina salvando le modifiche
svolgimenti alternativi	3b) l'utente seleziona una canzone da rimuovere 4b) l'utente annulla le modifiche alla playlist

UC3	aggiunta album
livello	user goal
descrizione	l'artista carica un nuovo album sul suo profilo
attori	artista
pre-condizioni	l'artista deve avere le credenziali per effettuare l'accesso
post-condizioni	il nuovo album è visibile se cercato dagli utenti
normale svolgimento	1) l'artista seleziona l'opzione carica album 2) inserisce i nomi delle canzoni ed i rispettivi dati 3) l'artista termina l'inserimento salvando l'album
svolgimenti alternativi	3b) l'artista annulla il caricamento dell'album

UC4	iscrizione ad un artista
livello	user goal
descrizione	l'utente aggiunge un artista agli artisti seguiti
attori	cliente
pre-condizioni	l'utente deve avere le credenziali per effettuare l'accesso
post-condizioni	nel momento in cui l'artista carica un nuovo album l'utente può visualizzarlo nella sezione "nuovi rilasci"
normale svolgimento	1) l'utente apre la pagina di ricerca 2) inserisce il nome di una canzone 3) seleziona una voce dall'elenco proposto 4) seleziona l'opzione "aggiungi in coda"

2.5 Mockups

2.5.1 Pagina di accesso



Welcome to Swetify!

username:

password:

Don't have an account yet?
Register [now](#)

2.5.2 Pagina di ricerca

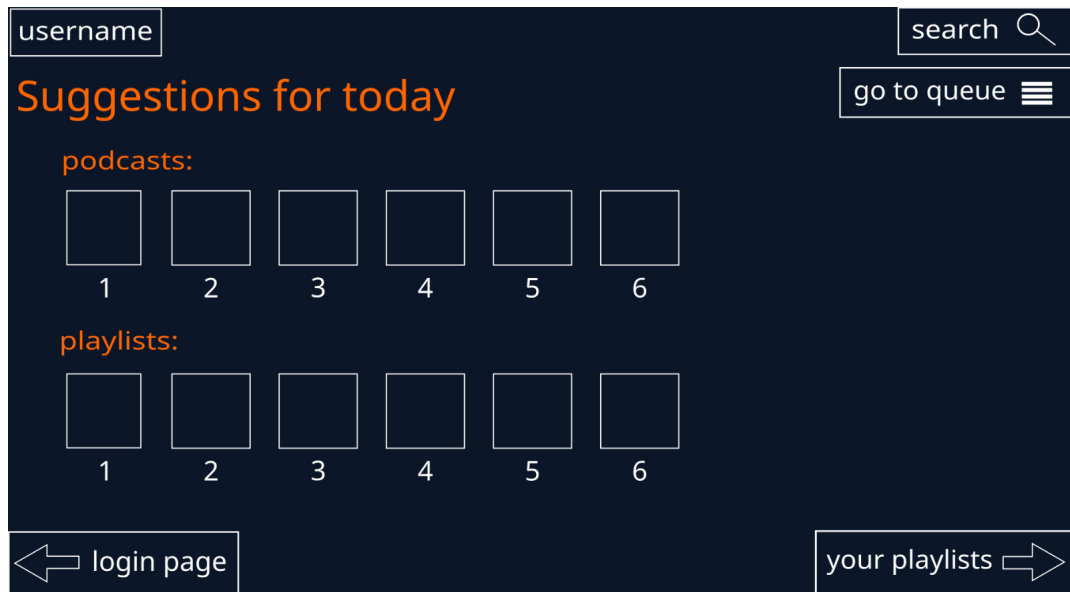


searchName X

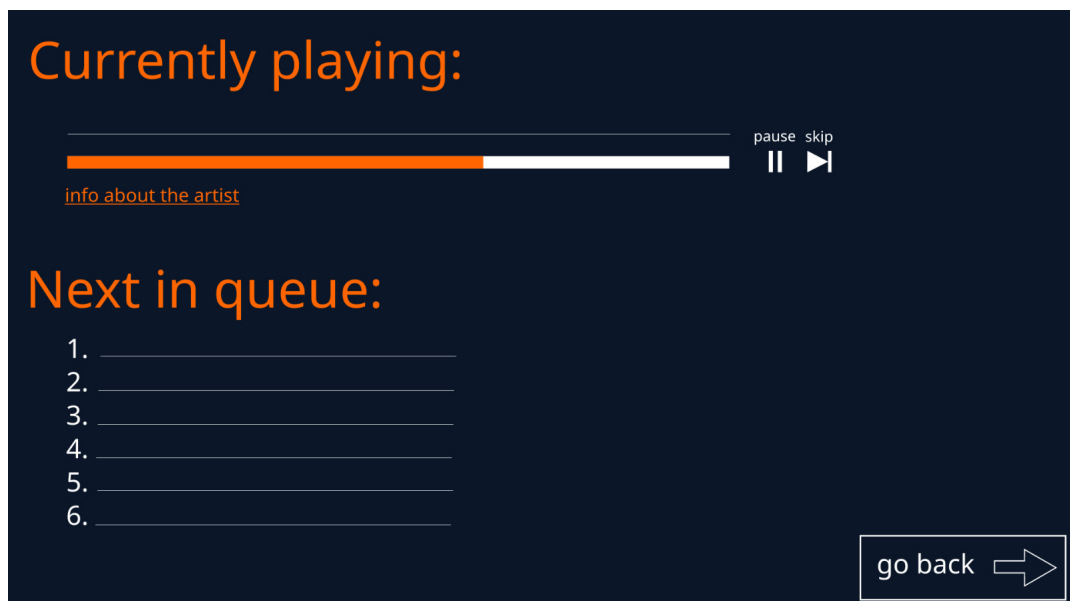
Songs:	Podcasts:	Artists:
1. _____	1. _____	1. _____
2. _____	2. _____	2. _____
3. _____	3. _____	3. _____
4. _____	4. _____	4. _____
5. _____	5. _____	5. _____
6. _____	6. _____	6. _____
7. _____	7. _____	7. _____
8. _____	8. _____	8. _____
9. _____	9. _____	9. _____

go back →

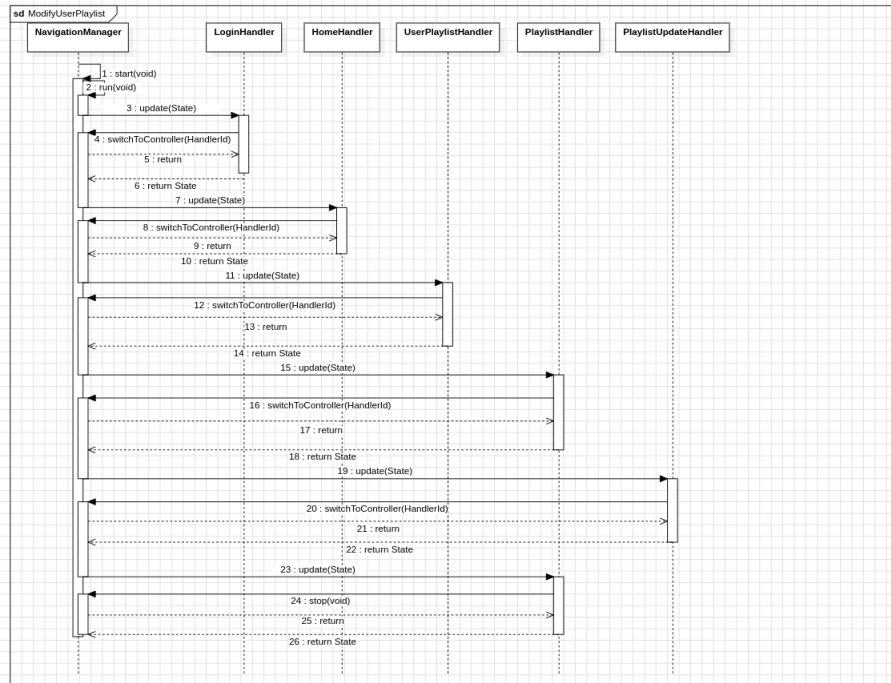
2.5.3 Suggerimenti



2.5.4 Coda di riproduzione



2.6 Sequence diagram



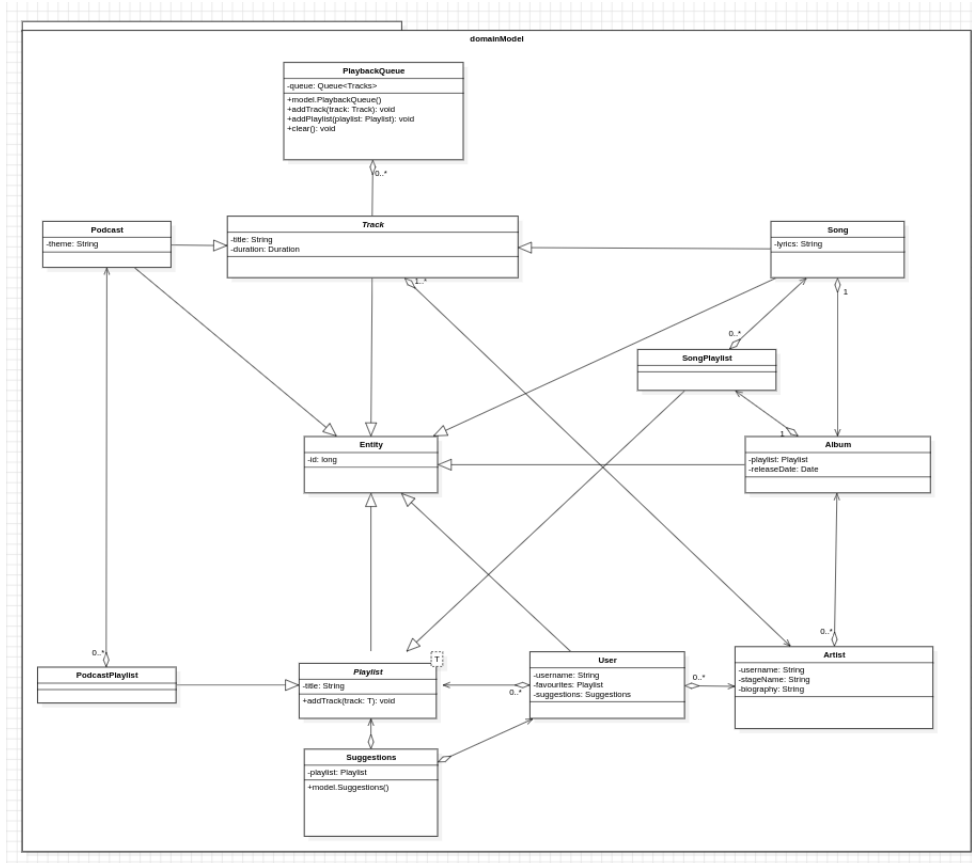
3 Scelte di progetto

Abbiamo scelto di strutturare il progetto creando una divisione tra domain model, business logic e data access objects.

- Il package **domainmodel** definisce le classi che simboleggiano gli elementi con cui l'utente interagisce.
- il package **businesslogic** descrive le modalità e le possibilità di interazione che l'utente ha con tali oggetti.
- il **dao** garantisce la persistenza di alcuni oggetti all'interno dell'applicazione.

4 Documentazione

4.1 Domain model



4.1.1 Entity

Classe base astratta che garantisce la presenza di un ID al fine di localizzare gli oggetti nel database.

4.1.2 Customer

Rappresenta il cliente e contiene le sue credenziali di accesso oltre alle playlist salvate.

4.1.3 Artist

Rappresenta l'artista e contiene le sue credenziali d'accesso e gli album da lui caricati.

4.1.4 Track

Rappresenta un qualsiasi oggetto che può essere aggiunto alla coda di riproduzione.

4.1.5 Song

Concretizzazione di Track che rappresenta una canzone.

4.1.6 Podcast

Concretizzazione di Track che rappresenta un podcast.

4.1.7 Playlist

Rappresenta una playlist ed espone i metodi per aggiungere o rimuovere delle track in testa oppure in coda.

4.1.8 Album

Permette all'utente di riprodurre le canzoni al suo interno ma è immutabile.

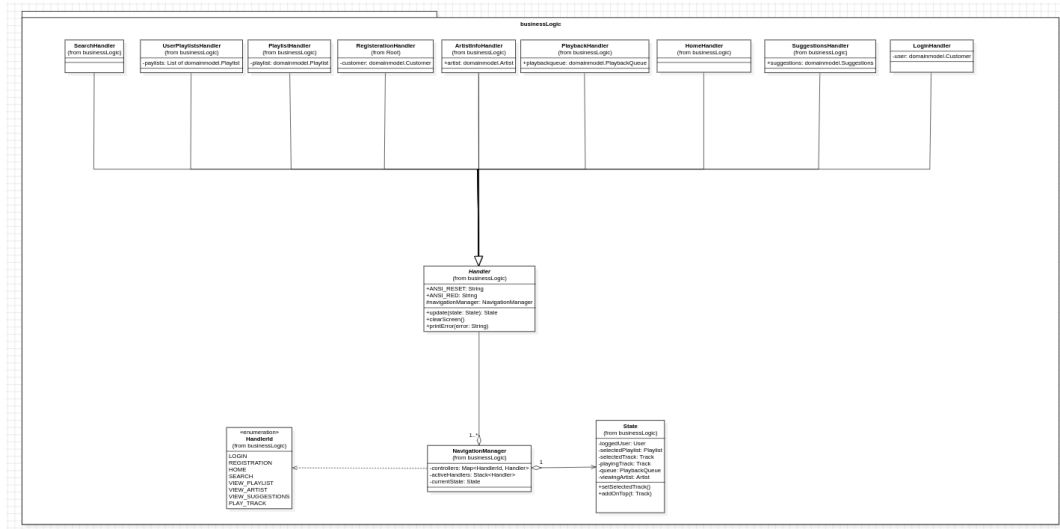
4.1.9 Suggestion

Rappresenta il possibile interesse del cliente verso una track.

4.1.10 PlaybackQueue

Rappresenta la coda di riproduzione e permette di aggiungere in testa ed in coda oltre ad eliminare in testa.

4.2 Business logic



4.2.1 Handler

Classe base astratta che rappresenta un handler generico.

Essa garantisce la presenza di un metodo update in ogni handler che contiene la logica di quest'ultimo.

4.2.2 State

Classe che permette ai vari handler di comunicare tra loro dati come l'utente loggato, la canzone o la playlist selezionata

4.2.3 NavigationManager

Gestisce la navigazione tra le pagine passando il controllo ai vari handler.

4.2.4 AlbumsHandler

Permette di visualizzare o riprodurre un album.

4.2.5 ArtistInfoHandler

Mostra le informazioni salienti di un artista.

4.2.6 HomeHandler

Contiene la schermata di ingresso e instrada l'utente verso le varie pagine.

4.2.7 LoginHandler

Permette all'utente di effettuare l'accesso con un nome utente e una password o eventualmente passare alla schermata di registrazione.

4.2.8 PlaybackHandler

Gestisce la coda di riproduzione e permette all'utente di visualizzare i brani contenuti in essa.

4.2.9 PlaylistHandler

Mostra all'utente i brani contenuti in una data playlist e permette ad esso di aggiungerla alla coda di riproduzione.

4.2.10 RegistrationHandler

Permette all'utente di registrarsi all'interno dell'applicazione con un nome utente ed una password.

4.2.11 SearchHandler

Gestisce la ricerca all'interno delle canzoni, dei podcast e degli artisti disponibili.

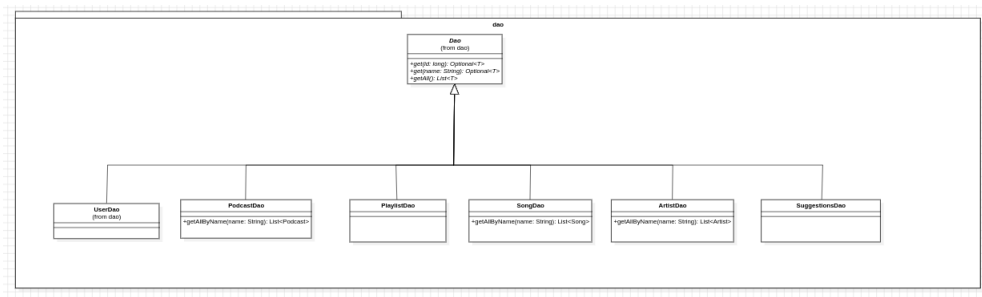
4.2.12 SuggestionsHandler

Mostra all'utente le track suggerite dall'applicazione.

4.2.13 UserPlaylistsHandler

Mostra l'elenco delle playlist create dall'utente.

4.3 Dao



4.3.1 Dao

Classe base astratta del data access object che garantisce la capacità di leggere o scrivere oggetti dal database.

4.3.2 CustomerDao

Permette di ottenere oggetti Customer dal database a patire dal loro nome utente o ID.

4.3.3 ArtistDao

Permette di ottenere oggetti Artist dal database a patire dal loro nome utente o ID.

4.3.4 SongDao

Permette di ottenere oggetti Song dal database in base ad una parola chiave.

4.3.5 PodcastDao

Permette di ottenere oggetti Podcast dal database in base ad una parola chiave.

4.3.6 PlaylistDao

Permette di leggere e scrivere le playlist dell'utente dal database.

5 Test effettuati