

1. Software Requirement Specification

1.1 Introduzione

1.1.1 Scopo del documento

Lo scopo di questo documento è quello di esplicitare i vari passi della progettazione e realizzazione del software **PlaylistHub** come progetto per l'esame di Ingegneria del Software e progettazione Web. A cura di: Alessia Cinelli, Andrea Tozzi.

1.1.2 Panoramica del sistema definito

Il progetto proposto consiste nello sviluppo di un'applicazione web dedicata alla gestione e fruizione di playlist musicali.

La principale funzione dell'app è la possibilità per gli utenti di esportare playlist (essa verrà ascoltata sulla piattaforma musicale su cui sono state caricate). La ricerca delle playlist è facilitata dalla possibilità di inserire filtri tra generi musicali e tonalità emotive.

Gli utenti registrati avranno la possibilità di arricchire ulteriormente la collezione di playlist messa a disposizione dalla piattaforma, aggiungendone delle nuove.

Per garantire la qualità del servizio, l'app è dotata di un sistema di moderazione amministrativa, da parte di un utente supervisore, che potrà decidere se approvare o eliminare le playlist caricate dagli utenti, nel caso queste non dovessero rispettare le linee guida. A seguito della gestione della propria playlist, l'utente riceverà una notifica, in un menù apposito, con un messaggio che specifica la decisione presa dal supervisore (approvazione o eliminazione).

1.1.3 Requisiti HW e SW:

Software:

- Star UML per la creazione dei diagrammi relativi alla progettazione del sistema.
- Scene Builder per la generazione di file FXML destinati alla realizzazione dell'interfaccia grafica.
- IntelliJ IDEA come piattaforma di sviluppo.

Hardware:

- Sistemi operativi:
 - Windows 7 o versioni successive.
 - MacOS v10.7 o superiore.
- Processore: Minimo 1 GHz, consigliato 2 GHz.
- Memoria (RAM): Minimo 2 GB, consigliato 4 GB.
- Hard Disk: Minimo 3 GB, consigliato 5 GB.

1.1.4 Sistemi correlati:

Groovifi: Genera automaticamente delle playlist a partire da una canzone e da alcuni parametri musicali, entrambi a scelta dell'utente. La playlist generata contiene delle canzoni molto compatibili con quella fornita in fase di creazione.

Pro:

- È possibile accedere con il proprio account Spotify per esportare velocemente la playlist sul proprio profilo ed ascoltarla.
- Accedendo alla lista delle canzoni ascoltate di recente, Groovifi consiglia delle playlist ancora prima di iniziare una generazione basata su parametri personali.

Contro:

- Non è facile scoprire musica nuova perché i consigli sono basati sugli stessi parametri della canzone presa in input.
- Le generazioni restituiscono spesso gli stessi risultati.

Spotify: La piattaforma di streaming musicale più utilizzata.

Pro:

- Essendo la più famosa applicazione per lo streaming musicale, per gli utenti è molto facile e veloce accedere al menù per la scoperta di nuove playlist e canzoni.
- Vasta scelta di playlist catalogate per generi ed umore.

Contro:

- Molte playlist dello stesso genere ed umore sono simili tra di loro, essendo tutte costruite su parametri come la popolarità.

1.2 User stories

US-1:

Come utente registrato, vorrei salvare le *playlist* sul mio account, in modo che siano facilmente accessibili in futuro. (Non è stato implementato)

US-2:

Come utente, vorrei visualizzare il numero di like di una *playlist*, per valutarne la popolarità. (Non è stato implementato)

US-3:

Come utente registrato, vorrei proporre le mie *playlist*, per condividerle con altri utenti.

US-4:

Come supervisore, vorrei approvare o rifiutare le nuove *playlist*, per garantire la qualità del servizio.

US-5:

Come utente registrato, vorrei definire le mie *preferenze musicali*, per risparmiare tempo nelle ricerche. (Implementato in parte)

US-6:

Come utente, vorrei filtrare le *playlist* impostando uno o più generi musicali, per ascoltare la musica adatta alla situazione in cui mi trovo.

US-7:

Come utente registrato, vorrei ricevere una *notifica* quando una mia *playlist* viene *gestita*, per conoscerne l'esito.

Dizionario:

- *Playlist:* Titolo, link, generi musicali contenuti, stati emotivi, autore.
- *Preferenze musicali:* generi musicali preferiti, impostabili in fase di creazione dell'account.
- *Generi musicali:* Pop, Indie, Classic, Rock, Electronic, House, HipHop, Jazz, Acoustic, REB, Country, Alternative.
- *Stati Emotivi:* Sadness, Happiness, Danceable, Chill, Electronic, Acoustic, Speechiness, Instrumental.
- *Notifica:* Notifica permanente nel sistema.
- *Gestita:* Approvata o rifiutata da parte del supervisore.

1.3 Functional Requirements

FR-1

Il sistema invia una *notifica* all'autore della *playlist*, quando viene *gestita* dal supervisore.

FR-2

Il sistema verifica la *validità del link* inserito dall'utente.

FR-3

Il sistema garantisce l'assenza di duplicati tra i link delle *playlist*.

FR-4

Il sistema offre la ricerca di *playlist* filtrate per *generi musicali*.

FR-5

Il sistema imposta le playlist caricate in uno *stato di attesa*, per preservare la qualità del servizio.

FR-6

Il sistema controlla se l'utente è autorizzato a caricare una *playlist* al momento della richiesta.

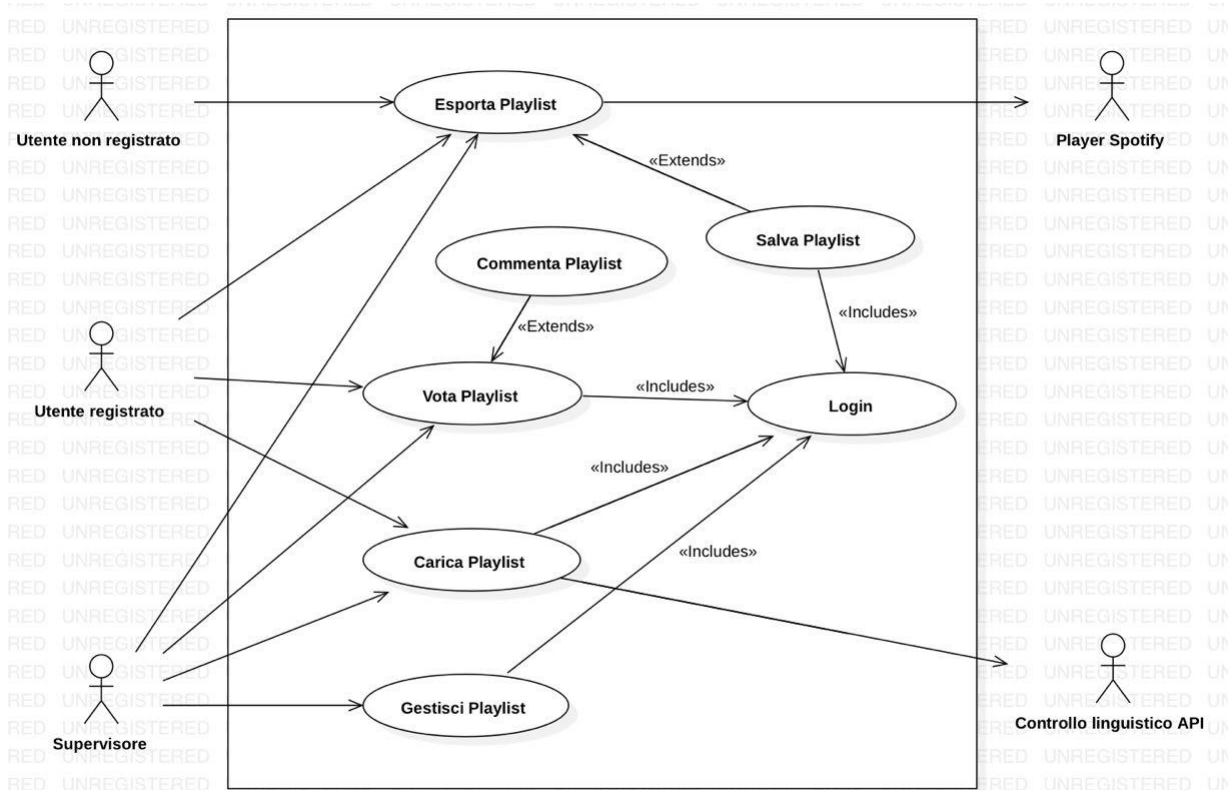
(Non è stato implementato)

Dizionario:

- *Notifica*: un messaggio visibile a schermo composto da: Titolo della playlist gestita e decisione effettuata.
- *Playlist*: Titolo, link, generi musicali contenuti, stati emotivi, autore.
- *Generi musicali*: Pop, Indie, Classic, Rock, Electronic, House, HipHop, Jazz, Acoustic, REB, Country, Alternative
- *Validità del link*: consiste nel verificare se il link fornito corrisponde a un indirizzo web reale.
- *Esito*: Approvata o in Attesa di approvazione.
- *Stato di attesa*: Non visibili finché non saranno gestite dal supervisore.
- *Gestita*: Approvata o rifiutata da parte del supervisore.

1.4 Use Cases

OVERVIEW DIAGRAM



HMW - UC Internal Steps

Carica playlist (Utente registrato):

1. L'utente seleziona l'opzione: carica playlist.
2. Il sistema verifica se l'utente è *abilitato* per l'operazione. (non è stato implementato)
3. Il sistema prepara una scheda playlist vuota.
4. L'utente inserisce i campi richiesti.
5. L'utente indica di aver terminato la compilazione della playlist.
6. Il sistema verifica la *correttezza dei campi* inseriti.
7. Il sistema contrassegna la playlist: "In attesa di approvazione".
8. Il sistema salva la playlist.

Extensions:

- 2a. L'utente non ha i permessi per caricare la playlist: Il sistema notifica l'utente e termina il caso d'uso.
- 6a. I campi inseriti dall'utente non sono corretti: Il sistema evidenzia i campi errati e indica la motivazione.
- 7a. Il database non risponde e la playlist non viene contrassegnata: Il sistema notifica il fallimento e termina il caso d'uso.

Dizionario:

- **Abilitato:** Gli utenti registrati possono essere privati della possibilità di caricare playlist se vengono rifiutate per tre volte consecutive. (non è stato implementato)
- **Correttezza dei campi:**
 - Il link deve essere presente una sola volta nel database.
 - Deve essere selezionato almeno un genere musicale.

Gestisci Playlist (Supervisore):

1. Il supervisore seleziona l'opzione: Gestisci Playlist.
2. Il sistema recupera le playlist in attesa di approvazione.
3. Il sistema mostra una tabella con le playlist recuperate.
4. Il supervisore seleziona una playlist di cui vuole visualizzare le informazioni. (non è stato implementato)
5. Il sistema mostra una scheda contenente le informazioni della playlist selezionata. (non è stato implementato)
6. Il supervisore verifica se il link è associato ad una playlist musicale.
7. Il supervisore seleziona una *scelta* sulla playlist.
8. Il sistema modifica il campo approvazione della playlist, con la scelta effettuata.
9. Il sistema rimuove dalla tabella la playlist gestita.

Extensions:

2a. *Il database non risponde e le playlist non vengono recuperate*: Il sistema notifica il fallimento e termina il caso d'uso.

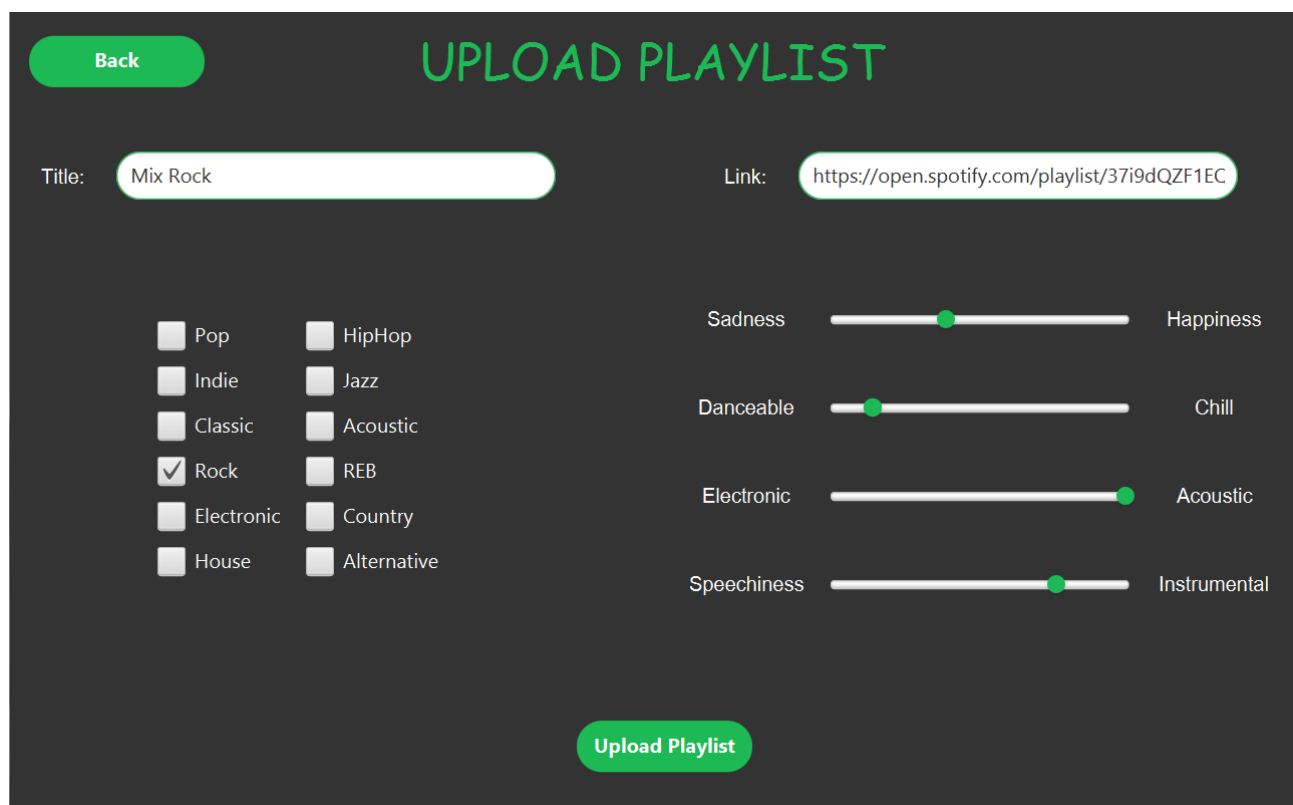
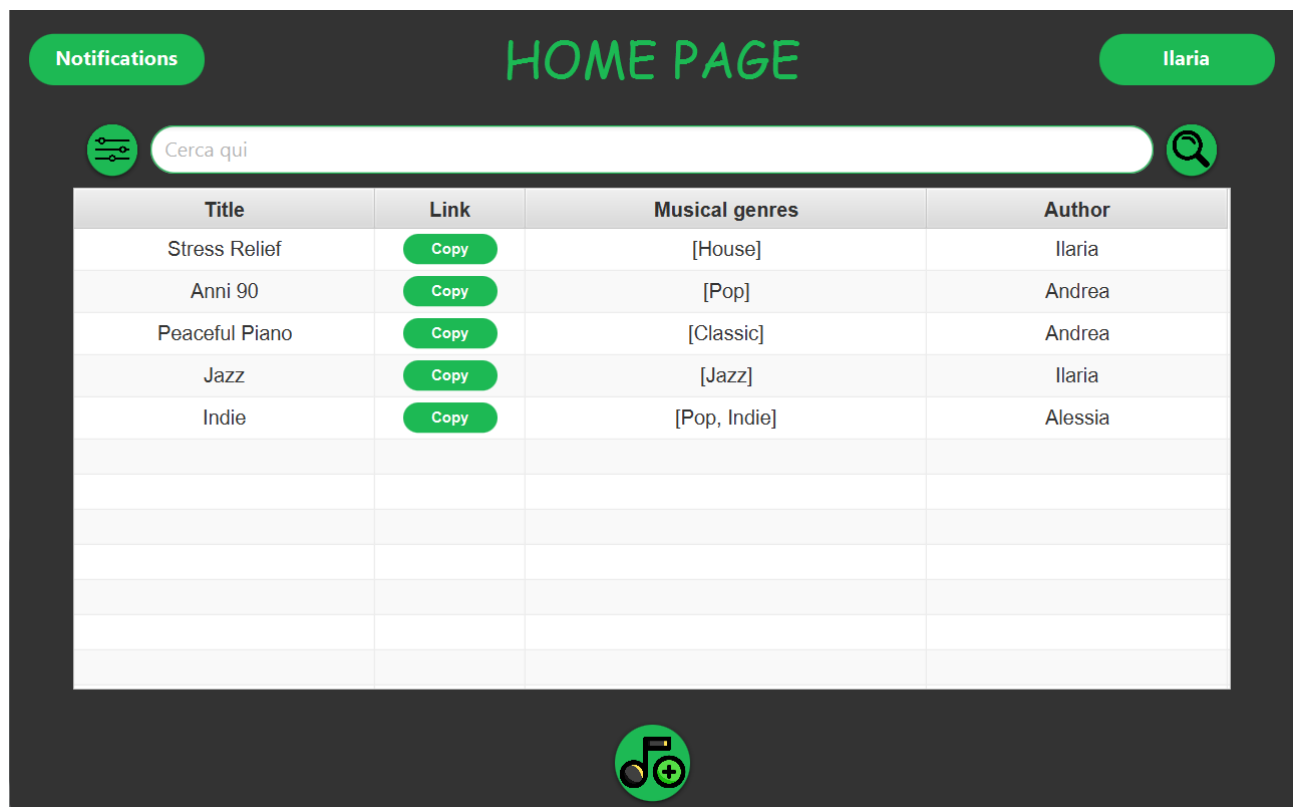
8a. *Playlist non trovata, l'autore ha eliminato la playlist durante la fase di gestione*: Il sistema notifica il fallimento al supervisore e termina il caso d'uso.

8b. *La playlist è stata gestita da un altro supervisore con un esito differente*: Il sistema notifica l'incongruenza al supervisore e chiede di effettuare una nuova scelta.

Dizionario:

- *Scelta*: Approva o rifiuta.

2. Storyboards



[Back](#)

MANAGE PLAYLISTS

Title	Link	Musical genres	Author	Approve or Reject
Stress Relief	Copy	[House]	Ilaria	V X
Anni 90	Copy	[Pop]	Andrea	V X
Peaceful Piano	Copy	[Classic]	Andrea	V X
Chill Hits	Copy	[Pop, Classic]	Ilaria	V X
Rock Classic	Copy	[Classic, Rock]	Andrea	V X
Jazz	Copy	[Jazz]	Ilaria	V X

[Back](#)

ACCOUNT

Username: Lollogiga

Email: lollogiga@gmail.com

Musical Genres

☒ Pop

☐ Indie

☐ Classic

☒ Rock

☐ Electronic

☐ House

☒ HipHop

☒ Jazz

☒ Acoustic

☐ REB

☐ Country

☐ Alternative

[Save](#)

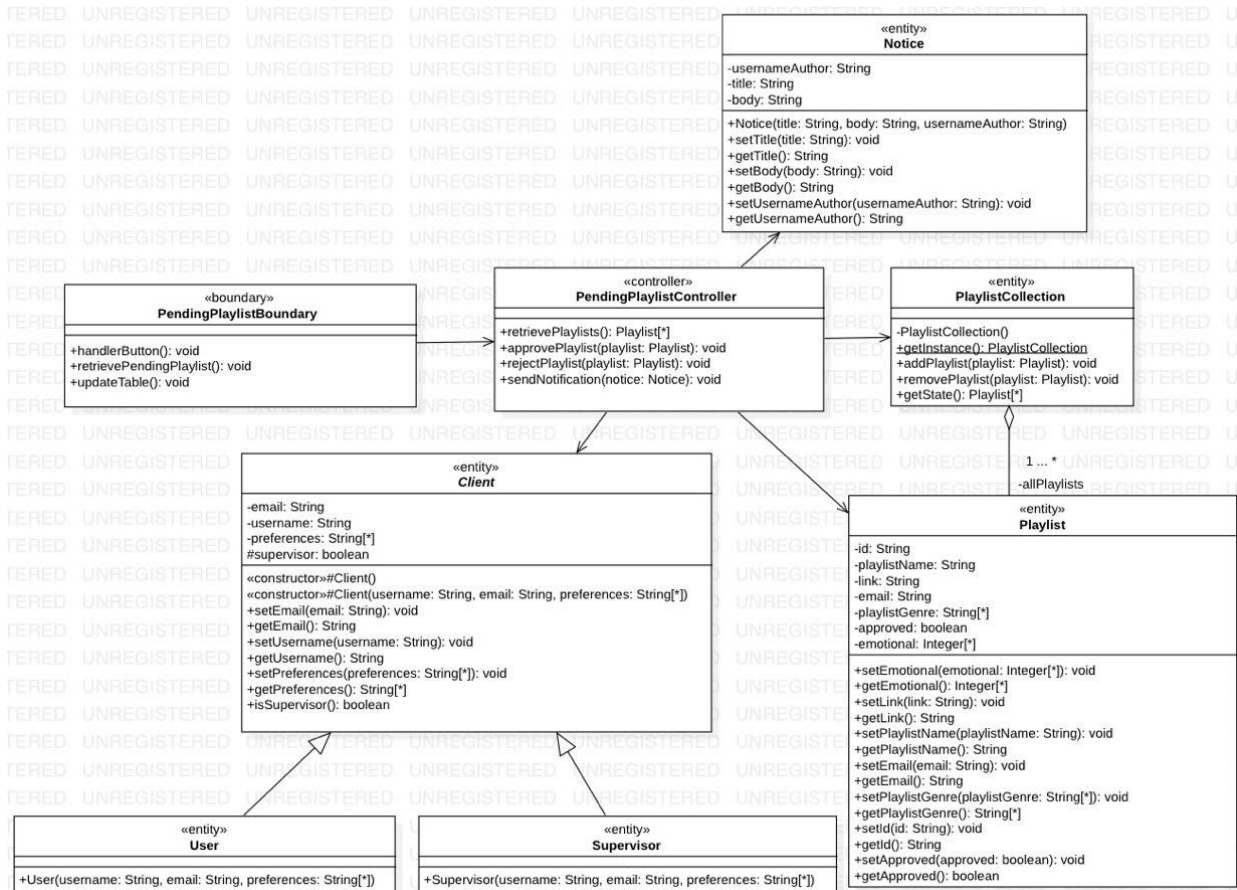
Approved	Title	Link	Musical genres
	Onde Alfa	Copy	[Classic]
	Mix House	Copy	[House]

3. Design

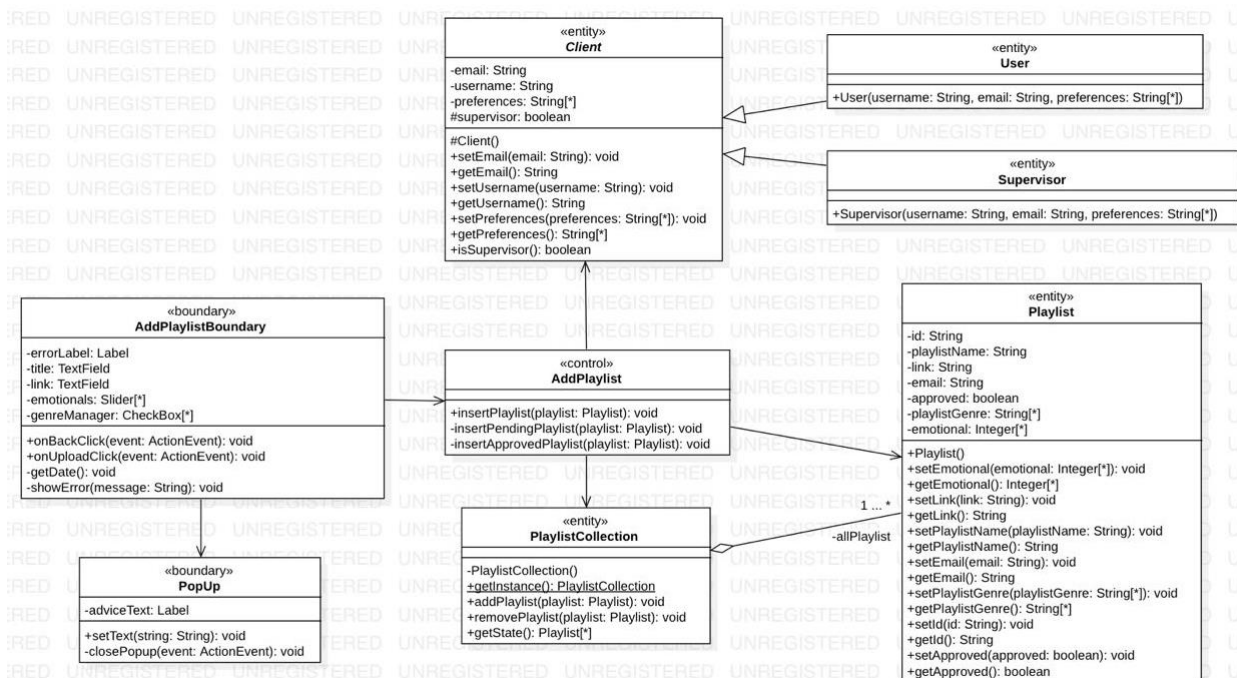
3.1 Class Diagram

3.1.1 VOPC (Analysis)

VOPC relativo al caso d'uso: *Gestisci Playlist*

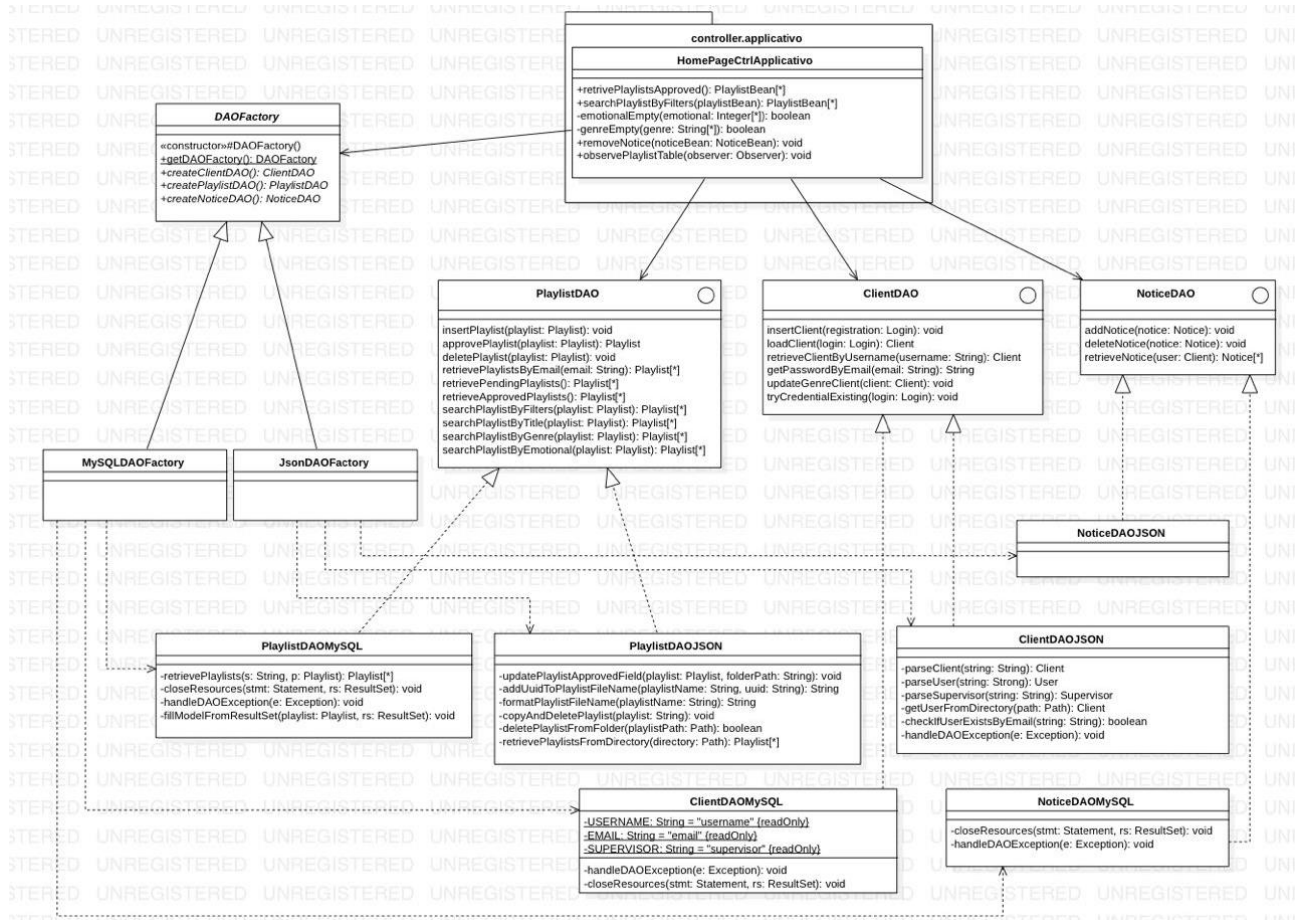


VOPC relativo al caso d'uso: *Carica Playlist*

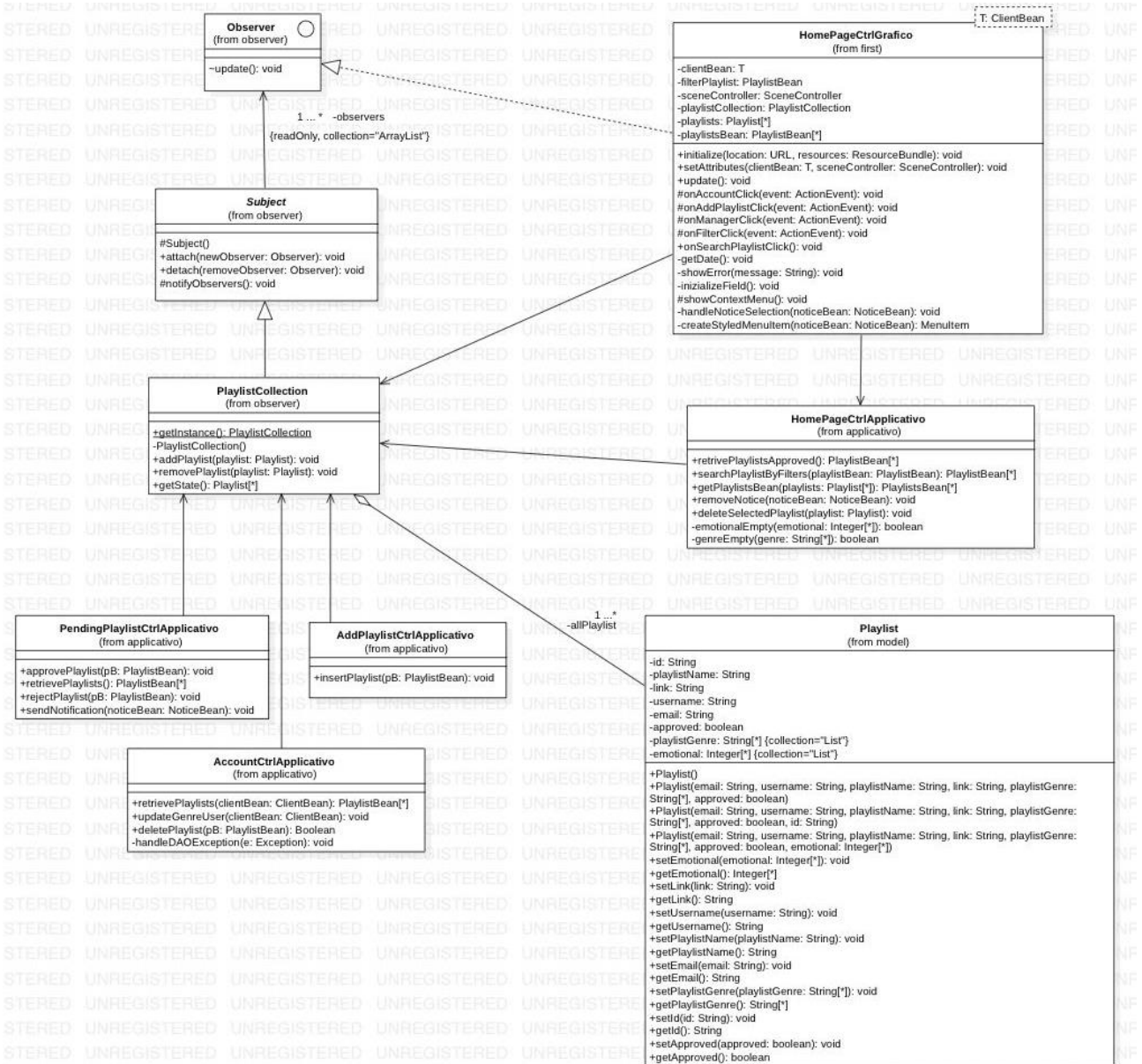


3.1.2 Design-Level Diagram

Il Design-Level Diagram riportato è relativo ad un Pattern Abstract Factory.

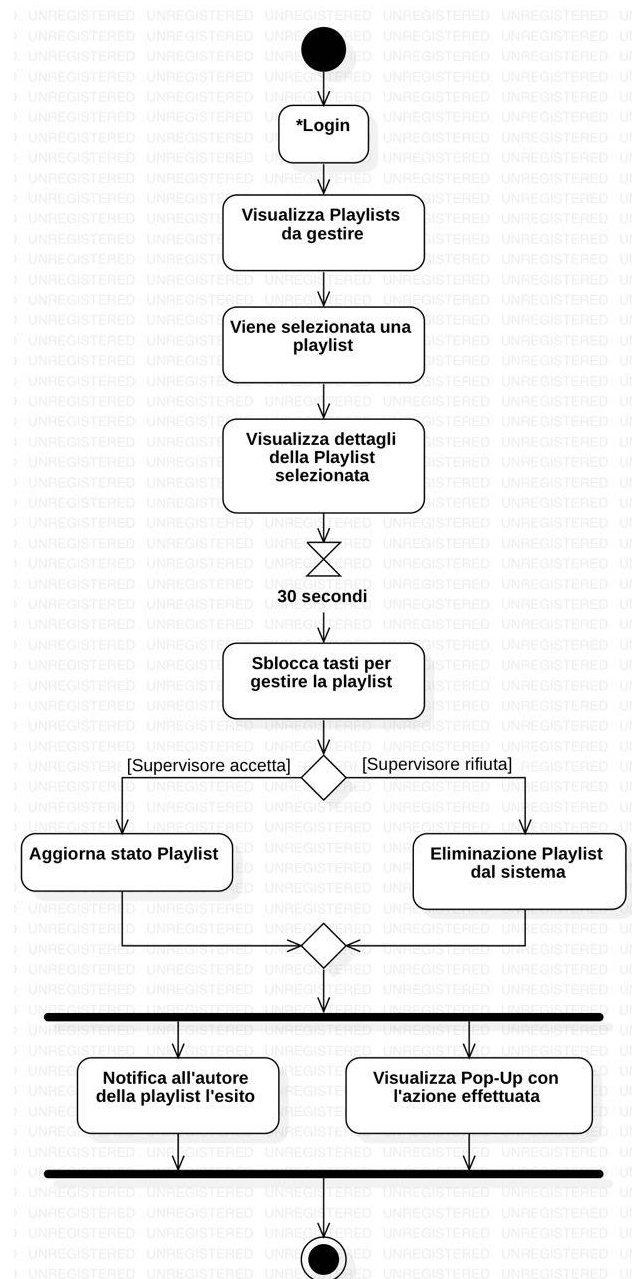


Il Design-Level Diagram riportato è relativo ad un Pattern Observer.



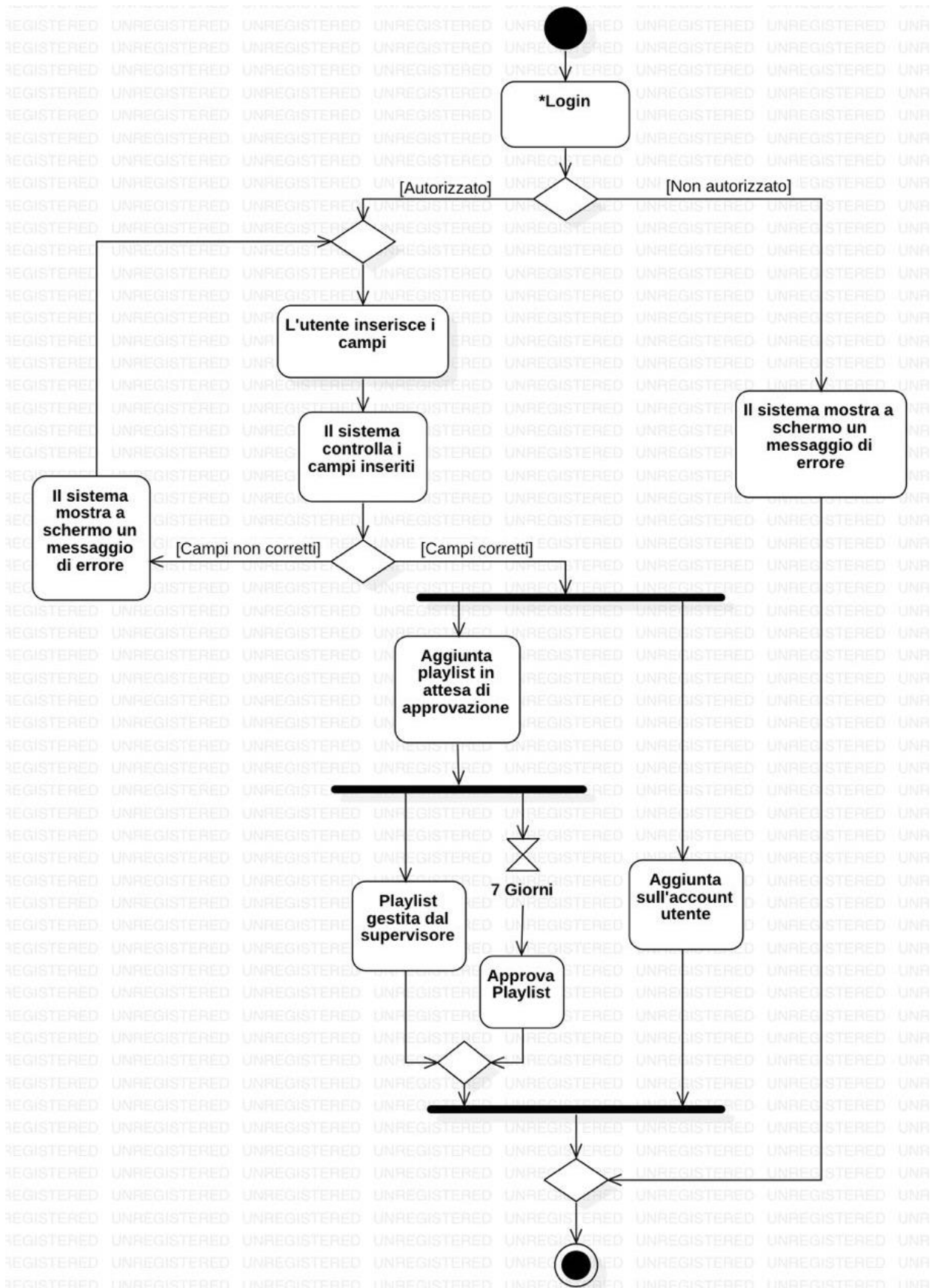
3.2 Activity Diagram

Activity Diagram relativo al caso d'uso: *Gestisci Playlist*



Il diagramma non rispetta quanto implementato:

- Non vengono visualizzati ulteriori dettagli relativi alla playlist selezionata.
- Non vengono bloccati i tasti di gestione.
- Non viene visualizzato un pop-up a seguito dell'azione.

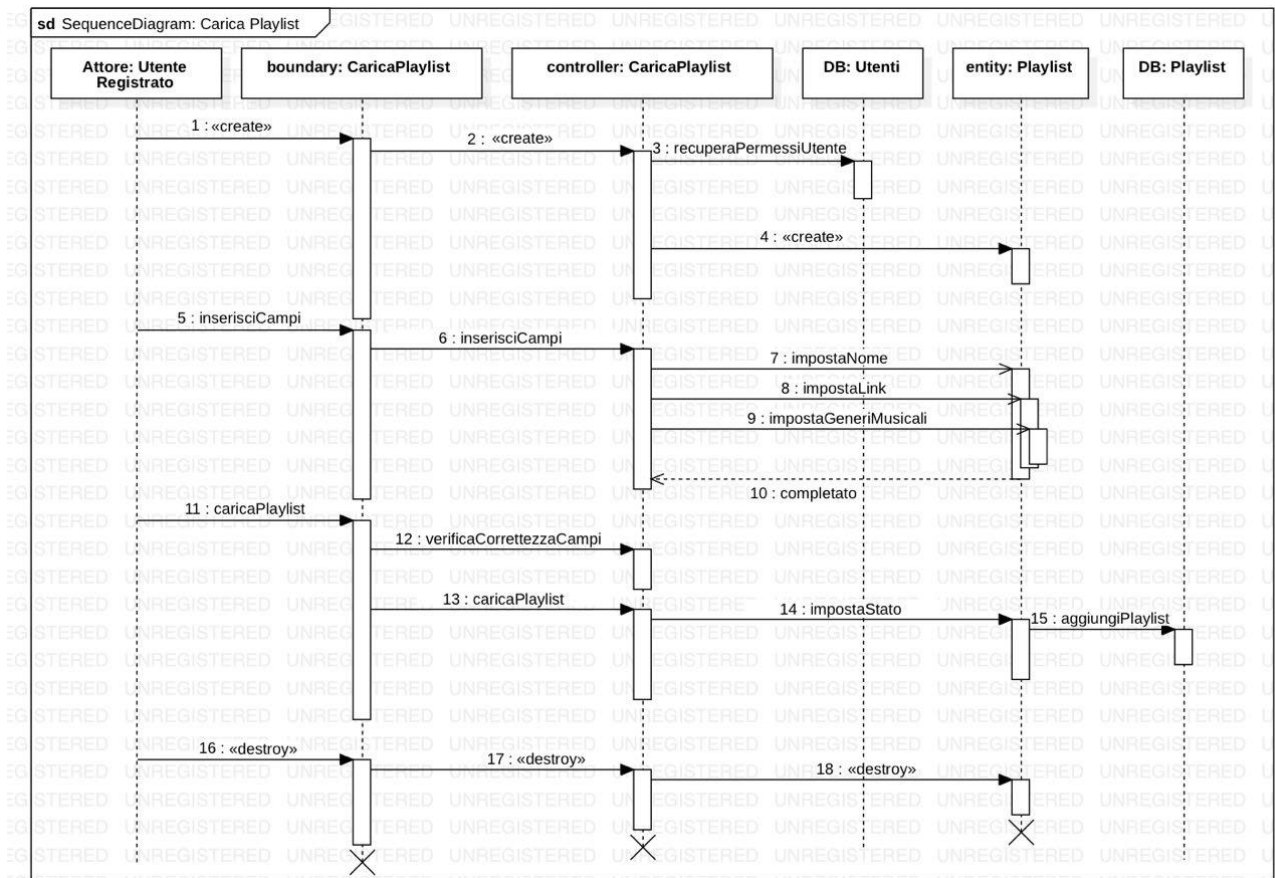
Activity Diagram riguardante il caso d'uso: Aggiungi Playlist

Il diagramma non rispetta quanto implementato:

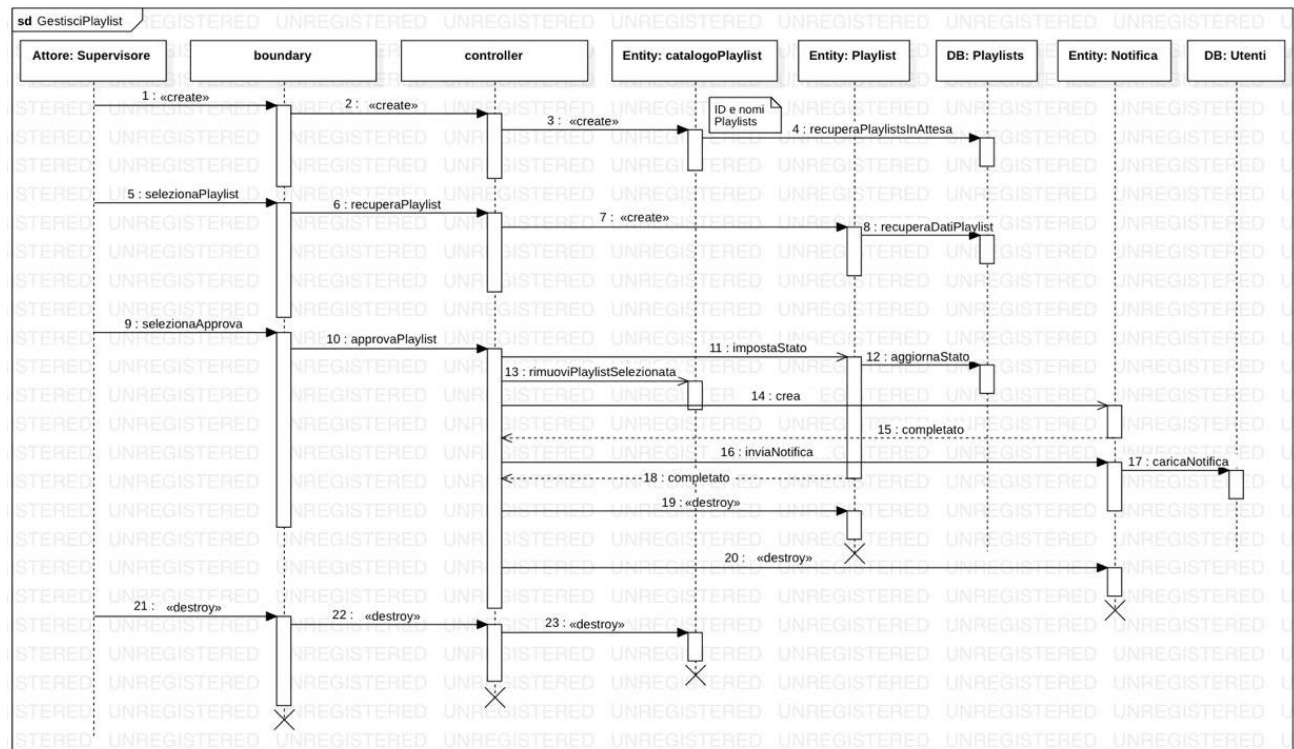
- Non viene effettuata l'approvazione automatica dopo sette giorni.

3.3 Sequence Diagram

Sequence Diagram relativo al caso d'uso: *Carica Playlist*



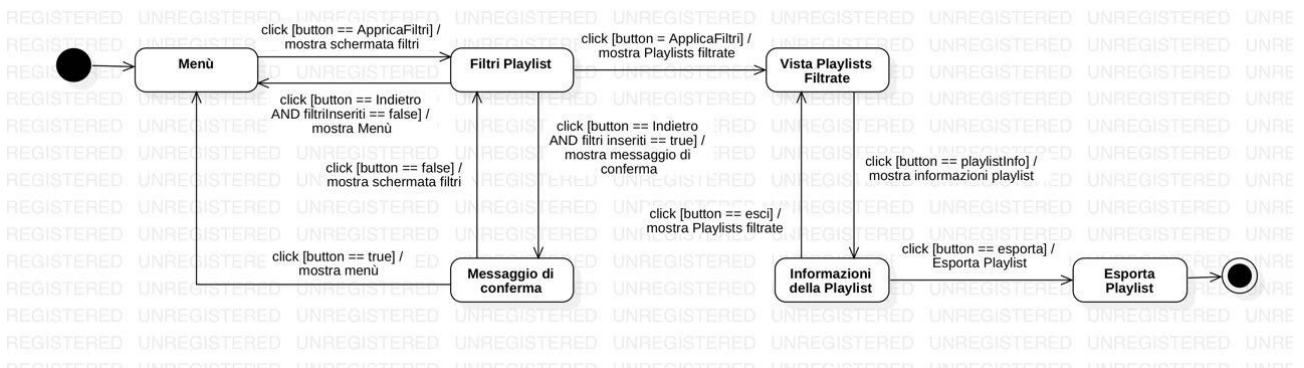
Sequence Diagram relativo al caso d'uso: *Gestisci Playlist*



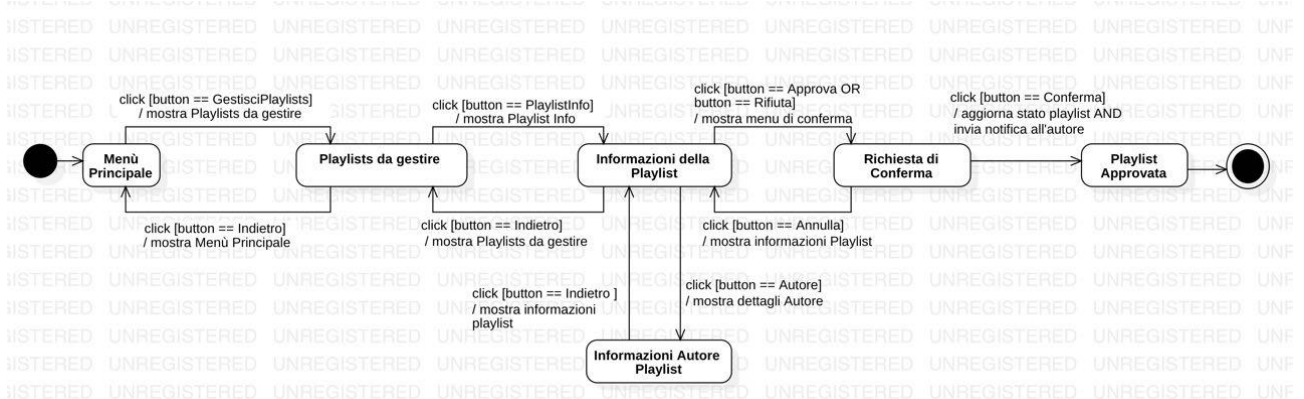
I diagrammi non rispettano del tutto quanto è stato implementato.

3.4 State Diagram

State Diagram relativo al caso d'uso: Esporta Playlist



State Diagram relativo al caso d'uso: Gestisci Playlist



I diagrammi non rispettano del tutto quanto è stato implementato.

4. Testing

I test effettuati da Alessia Cinelli riguardano:

1. Il corretto caricamento di una playlist in persistenza.
2. La corretta eliminazione di una playlist in persistenza.
3. Il corretto funzionamento del filtraggio per genere musicale delle playlist approvate.

I test effettuati da Andrea Tozzi riguardano:

1. La corretta registrazione di un utente nel caso vengano inseriti i dati correttamente.
2. La ricezione di una eccezione nel caso in cui in fase di registrazione un utente inserisca un'e-mail già registrata.
3. La ricezione di una eccezione nel caso in cui un utente in fase di registrazione inserisca un username già utilizzato.

5. SonarCloud

Il link di SonarCloud associato al progetto è:

https://sonarcloud.io/summary/overall?id=mzlala_PROG_ISPW