



# UNIVERSITÀ DEGLI STUDI DI PALERMO

**TESINA DI PROGETTO:** SERVIZIO WEB DI STREAMING AUDIO

**CORSO DI LAUREA:** INGEGNERIA INFORMATICA E DELLE  
TELECOMUNICAZIONI

**CORSO:** PROGRAMMAZIONE WEB E MOBILE

**ANNO ACCADEMICO:** 2017/2018

**DOCENTE:** ROBERTO PIRRONE

**A CURA DI:**

GIANMARCO COPPOLA

GIUSEPPE MOSCARELLI

ANDREA VARA

# INDICE

<b>1. Obiettivi generali.....</b>	<b>3</b>
1.1 Definizioni.....	3
<b>2. Sistema corrente.....</b>	<b>4</b>
<b>3. Sistema proposto.....</b>	<b>4</b>
3.1 Panoramica del sistema.....	4
3.2 Requisiti funzionali.....	4
3.3 Requisiti non funzionali.....	5
<b>4. Modelli del sistema.....</b>	<b>6</b>
4.1 Modelli dei casi d'uso.....	6
4.1.1 Attori coinvolti.....	6
4.1.2 Visione d'insieme e macro-casi d'uso.....	7
4.1.3 Autenticazione.....	8
4.1.4 Gestione playlist.....	10
4.1.5 Gestione amicizia.....	13
4.1.6 Gestione account.....	14
4.1.7 Riproduzione.....	15
4.2 Diagrammi di sequenza.....	17
4.3 Diagrammi delle classi.....	25
<b>5. Architettura software proposta.....</b>	<b>30</b>
5.1 Panoramica generale.....	30
5.2 Database.....	30
5.2.1 Descrizione verbale del database.....	30
5.2.2 Diagramma entità-relazione.....	31
5.2.3 Descrizione delle tabelle.....	31
5.2.4 Vincoli di tupla.....	36
5.2.5 Diagramma entità-relazione MySQL Workbench.....	37
5.3 Decomposizione del sistema.....	38
5.3.1 Suddivisione dei ruoli all'interno del team di sviluppo.....	40
5.4 Diagramma di deployment.....	41

# 1. Obiettivi generali

Lo scopo del sistema proposto è quello di fornire un servizio di streaming audio agli utenti registrati tramite un browser web, o tramite l'apposita applicazione Android. Il sistema, chiamato SoundWave, permette inoltre agli utenti registrati di organizzare le proprie playlist, consentendo l'aggiunta, l'eliminazione o la modifica dei brani, e una funzione social che connette tra loro gli utenti.

## 1.1. Definizioni

<i>Autenticazione</i>	Procedura di accesso ad un sistema informatico mediante l'inserimento di opportuni dati di accesso.
<i>Dati utente</i>	Contengono tutti i dati associati ad un account.
<i>DBMS</i>	Database Management System, termine che si riferisce al database del sistema.
<i>Form</i>	Finestra contenente dei campi compilabili.
<i>Login</i>	Pagina che permette l'autenticazione di un utente.
<i>Logout</i>	Procedura per scollegarsi da un sistema informatico a cui si aveva avuto accesso tramite un'operazione di login.
<i>Notifica</i>	Notifica mostrata dal sistema a seguito dell'esecuzione di un'operazione da parte dell'utente.
<i>Pannello</i>	Finestra mostrata dal sistema a seguito dell'esecuzione di un'operazione da parte dell'utente.
<i>Registrazione</i>	Pagina contenente dei campi compilabili per la creazione di un account con il quale poter successivamente effettuare il login.
<i>Streaming audio</i>	Servizio per la trasmissione di segnali audio via Internet, che permette di ascoltare i segnali provenienti da un server via via che questi vengono ricevuti senza dover attendere il download completo e senza prima averli salvati sul proprio computer.
<i>Utente</i>	Generico utente che si interfaccia con il sistema
<i>Web player</i>	Pagina contenente tutte le funzionalità principali offerte dal sistema e visualizzata dagli utenti al seguito del login.

## **2. Sistema corrente**

Si suppone che non esista alcun software con le stesse funzionalità del sistema proposto attualmente adottato da un utente.

## **3. Sistema proposto**

### **3.1. Panoramica del sistema**

Il sistema si propone di fornire un servizio di riproduzione di file audio in streaming sul web. Deve tenere traccia di tutte le informazioni associate ad ogni singolo utente sia dal punto di vista social che musicale, consentendo all'utente la gestione della propria lista amici, delle proprie playlist e dei brani in riproduzione. Il sistema deve inoltre fornire agli utenti operazioni di ricerca di brani e album con conseguente riproduzione anche al di fuori di una playlist personale.

### **3.2. Requisiti funzionali**

I requisiti funzionali descrivono le interazioni tra il sistema e il suo ambiente indipendentemente dalla sua implementazione. L'ambiente include l'utente e qualsiasi altro sistema esterno con cui il sistema interagisce (DBMS). Le funzionalità che il sistema offre all'utente sono riportate di seguito.

- **Autenticazione e registrazione**

SoundWave offre i propri servizi solamente agli utenti registrati alla piattaforma e che hanno attivato il proprio account tramite il link di attivazione ricevuto via email in fase di registrazione. Un utente registrato e che ha effettuato l'accesso ha la possibilità di modificare i dati del suo profilo come il nome o la password. Un utente registrato ha anche accesso alla funzionalità di recupero della password via email. Il sistema offre a qualsiasi utente la possibilità di registrarsi con i propri dati personali e di attivare l'account.

- **Riproduzione audio in streaming**

Funzionalità principale del sistema. SoundWave permette ad un utente autenticato di riprodurre la musica in streaming selezionando un qualsiasi brano presente all'interno del sistema.

- **Gestione delle playlist**

Il sistema permette ad un utente autenticato di gestire le proprie playlist per avere un catalogo personale dal quale selezionare i brani da riprodurre. Un utente può creare una nuova playlist, eliminarne una creata in precedenza, aggiungere o rimuovere i brani da una di esse.

- **Gestione degli amici**

Il sistema permette ad un utente autenticato di gestire la propria lista amici, quindi cercare altri utenti, aggiungere un utente alla propria lista amici ed eliminare un utente dalla lista. Inoltre un utente può vedere gli amici attualmente connessi al sistema e il loro brano attualmente in riproduzione. L'amicizia è univoca: se un utente 1 aggiunge un utente 2 alla sua lista degli amici potrà vederlo online e visualizzare il brano dell'amico in ascolto, ma l'utente 2 non avrà l'utente 1 tra gli amici. Per poter accedere a tutte le funzionalità relative all'amicizia l'utente 2 dovrà aggiungere l'utente 1 alla sua lista amici.

### **3.3. Requisiti non funzionali**

- *Affidabilità*

Il sistema permette l'accesso agli utenti in qualsiasi momento e fornisce ad essi le informazioni richieste interrogando il DBMS tramite una consegna affidabile dei dati.

- *Controllo sulla validità dei dati*

Il sistema impedisce l'inserimento di dati errati o inconsistenti da parte dell'utente, come l'inserimento di una password troppo corta o un indirizzo email che non rispetta il formato corretto.

- *Implementazione*

Il lato client del software è realizzato in HTML, CSS e Javascript. Il lato server del software è realizzato in Javascript e fa uso di Node.js e dei relativi moduli necessari alla gestione di tutte le funzionalità.

- *Sicurezza*

I dati sensibili vengono custoditi all'interno del database evitando perdite o intromissioni di terze parti. Inoltre tutte le password degli utenti registrati alla piattaforma vengono cifrate prima di essere memorizzate nel database.

- *Scalabilità*

Tutte le richieste dal server al database vengono effettuate in parallelo su una connessione diversa tramite un pool di connessioni per consentire più richieste contemporaneamente ed evitando rallentamenti dovuti ad un alto numero di richieste provenienti da diversi utenti.

- *Usabilità*

Gli utenti possono accedere alle funzionalità del sistema in maniera chiara e intuitiva.

- *Visualizzazione responsive*

Il sistema adatta la propria interfaccia grafica in base allo schermo del dispositivo sul quale è visualizzato. La visualizzazione si adatta a qualsiasi tipo di display (smartphone, tablet, desktop).

## **4. Modelli del sistema**

### **4.1. Modelli dei casi d'uso**

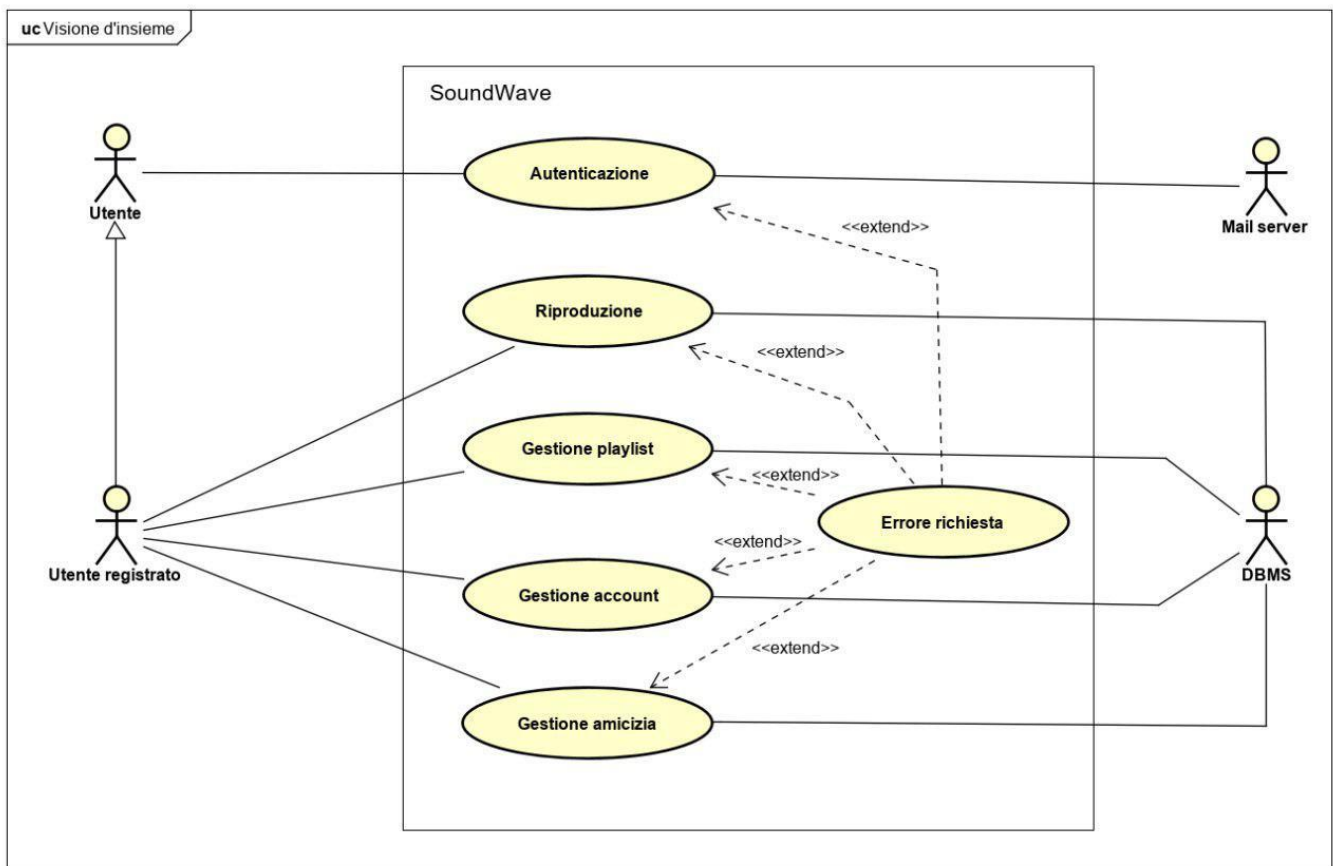
Nella trattazione dei casi d'uso, laddove l'utente abbia la possibilità di scegliere se andare avanti o meno nell'esecuzione di una operazione, assumeremo che l'utente scelga sempre di andare avanti, in quanto se scegliesse di non proseguire il sistema ritornerebbe ad una condizione precedente che non è utile ai fini dell'illustrazione del modello per il nostro sistema (perché possibilmente già descritta in precedenza).

#### **4.1.1. Attori coinvolti**

Gli attori coinvolti all'interno del sistema sono i seguenti:

<i>DBMS:</i>	attore che gestisce la creazione e la manipolazione dei dati e l'interazione efficiente con il database.
<i>Mail server:</i>	attore che gestisce l'invio delle mail ai client per l'attivazione degli account e il recupero delle password.
<i>Utente:</i>	generico attore che interagisce col sistema senza essere ancora registrato.
<i>Utente registrato:</i>	attore registrato a SoundWave che può usufruire dei servizi offerti dal sistema.

#### 4.1.2. Visione d'insieme e macro-casi d'uso

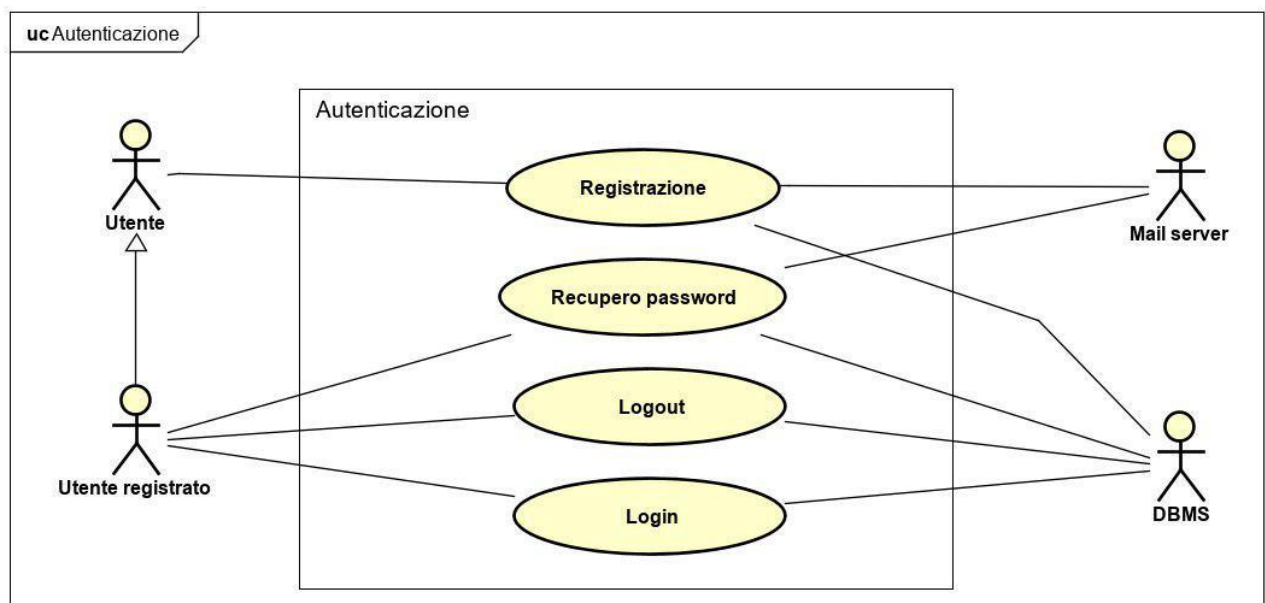


La visione d'insieme del sistema racchiude cinque macro casi d'uso (Autenticazione, Riproduzione, Gestione playlist, Gestione account e Gestione amicizia); comprende inoltre il caso d'uso Caduta connessione, che estende tutti i macro-casi d'uso.

<i><b>Autenticazione:</b></i>	racchiude i casi d'uso relativi all'accesso degli utenti al sistema, inclusa la registrazione.
-------------------------------	--

<b>Riproduzione:</b>	racchiude i casi d'uso relativi alla riproduzione dei brani.
<b>Gestione playlist:</b>	racchiude le funzionalità relative alla gestione delle playlist.
<b>Gestione account:</b>	racchiude le funzionalità relative alla gestione dei dati utente.
<b>Gestione amicizia:</b>	racchiude le funzionalità relative alla gestione degli amici.
<b>Errore richiesta:</b>	restituisce la pagina di errore nel caso in cui il client richieda un percorso inesistente.

### 4.1.3. Autenticazione



#### Registrazione

<b>Caso d'uso:</b> Registrazione
<b>ID:</b> RGR
<b>Attori:</b> Utente, DBMS, Mail server.
<b>Precondizioni:</b> 1. La pagina web di login al sito è stata visualizzata sul browser dell'utente.
<b>Sequenza degli eventi:</b> 1. L'utente clicca sul link "Registrati subito" presente all'interno del form di login. 2. Il sistema mostra la pagina web di registrazione, richiedendo l'inserimento di tutti i campi necessari alla creazione di un nuovo account. 3. L'utente compila il form di registrazione e clicca il tasto "Registrati".



<ol style="list-style-type: none"> <li>Il sistema invia i dati al DBMS.</li> <li>Il DBMS invia al sistema la conferma di avvenuta registrazione.</li> <li>Il mail server invia una e-mail all'utente contenente un link di attivazione dell'account.</li> <li>Il sistema mostra a schermo un messaggio che notifica il corretto invio della e-mail.</li> <li>L'utente clicca sul link di attivazione ricevuto tramite e-mail.</li> <li>Il sistema completa la creazione dell'account, attivandolo.</li> </ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"> <li>La registrazione dell'utente è stata effettuata con successo e il sistema mostra la pagina di login.</li> </ol>

## Recupero password

<b>Caso d'uso:</b> Recupero password
<b>ID:</b> RCRPSW
<b>Attori:</b> Utente registrato, DBMS, Mail server.
<b>Precondizioni:</b> <ol style="list-style-type: none"> <li>La pagina web di login al sito è stata visualizzata sul browser dell'utente registrato.</li> </ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"> <li>L'utente registrato clicca sul link "Clicca qui" per effettuare l'operazione di recupero password.</li> <li>Il sistema mostra un pannello contenente un campo in cui inserire l'indirizzo e-mail associato all'account dell'utente registrato.</li> <li>L'utente registrato compila il campo e preme il tasto "Recupera password".</li> <li>Il sistema richiede al DBMS la corrispondenza dell'indirizzo e-mail.</li> <li>Il DBMS conferma al sistema la corrispondenza dell'e-mail.</li> <li>Il mail server invia una e-mail all'utente registrato contenente una password generata casualmente con la quale può effettuare il prossimo accesso.</li> <li>Il sistema mostra a schermo un messaggio che notifica il corretto invio della e-mail.</li> </ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"> <li>Il sistema ha reimpostato la password dell'utente registrato.</li> </ol>

## Logout

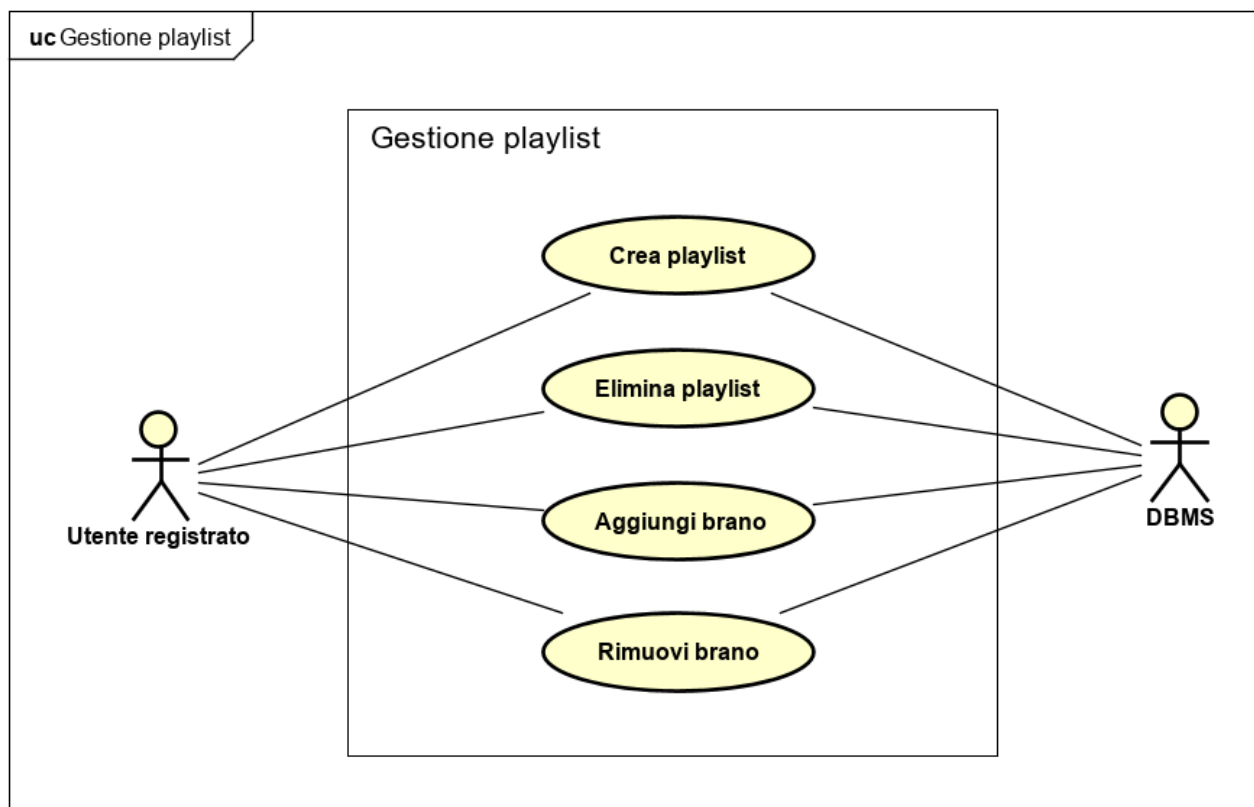
<b>Caso d'uso:</b> Logout
<b>ID:</b> LGL
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"> <li>La pagina web del web player è stata visualizzata sul browser dell'utente registrato dopo che questo si è autenticato.</li> </ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"> <li>L'utente registrato clicca sul tasto "Logout".</li> <li>Il sistema mostra a schermo una notifica in cui chiede conferma per il logout.</li> <li>L'utente registrato conferma di voler effettuare logout.</li> <li>Il sistema comunica al DBMS la disconnessione dell'utente.</li> <li>Il DBMS aggiorna i dati.</li> </ol>
<b>Postcondizioni:</b>

1. Il sistema ha mostrato la pagina di login.

## Login

<b>Caso d'uso:</b> Login
<b>ID:</b> LGI
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. La pagina web di login è stata visualizzata sul browser dell'utente registrato.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato compila il form di login e clicca sul tasto "Accedi".</li><li>2. Il sistema invia i dati inseriti dall'utente registrato al DBMS.</li><li>3. Il DBMS verifica la correttezza dei dati inseriti.</li><li>4. Il sistema mostra la schermata principale dell'applicazione (pagina del web player).</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente è stato autenticato dal sistema.</li></ol>

### 4.1.4. Gestione playlist



## Crea playlist

<b>Caso d'uso:</b> Crea playlist
<b>ID:</b> CPL
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente registrato ha cliccato il tasto "Le mie playlist".</li><li>2. Il sistema ha mostrato a schermo la schermata contenente le playlist dell'utente registrato.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato clicca il tasto "Crea playlist".</li><li>2. Il sistema mostra a schermo un pannello contenente un form dove inserire il nome della playlist da creare.</li><li>3. L'utente registrato inserisce il nome della playlist e clicca il tasto "Crea playlist".</li><li>4. Il sistema richiede al DBMS la creazione della nuova playlist.</li><li>5. Il DBMS crea la playlist e comunica al sistema che l'operazione è avvenuta con successo.</li><li>6. Il sistema aggiunge la nuova playlist alla lista delle playlist dell'utente registrato.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. La nuova playlist è stata creata con successo.</li></ol>

## Elimina playlist

<b>Caso d'uso:</b> Elimina playlist
<b>ID:</b> EPL
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente registrato ha cliccato il tasto "Le mie playlist".</li><li>2. Il sistema ha mostrato a schermo la schermata contenente le playlist dell'utente registrato.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato seleziona una delle sue playlist.</li><li>2. Il sistema mostra a schermo la lista dei brani all'interno della playlist ed un'icona "Elimina playlist".</li><li>3. L'utente registrato clicca sull'icona "Elimina playlist".</li><li>4. Il sistema invia al DBMS la richiesta di eliminazione della playlist.</li><li>5. Il DBMS elimina la playlist e comunica al sistema che l'operazione è avvenuta con successo.</li><li>6. Il sistema rimuove la playlist dalla lista delle playlist dell'utente registrato.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. La playlist è stata eliminata con successo.</li></ol>

## Aggiungi brano

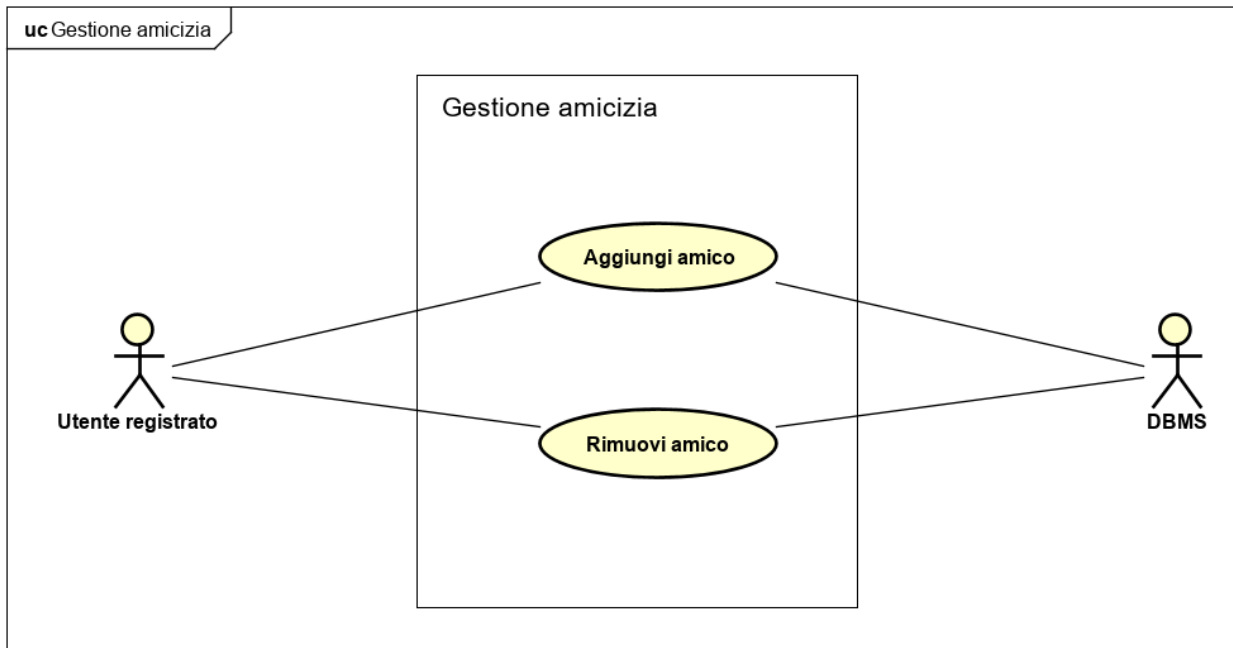
<b>Caso d'uso:</b> Aggiungi brano
<b>ID:</b> ABR
<b>Attori:</b> Utente registrato, DBMS.

<b>Precondizioni:</b>
1. Il sistema ha mostrato una lista di brani.
<b>Sequenza degli eventi:</b>
1. L'utente clicca l'icona "Aggiungi ad una playlist".
2. Il sistema mostra un pannello contenente la lista delle playlist dell'utente registrato.
3. L'utente seleziona una playlist tra quelle mostrate.
4. Il sistema richiede al DBMS di aggiornare la lista dei brani all'interno della playlist selezionata.
5. Il DBMS aggiunge il brano alla playlist e notifica al sistema che l'operazione avvenuta con successo.
6. Il sistema aggiunge il brano alla playlist.
<b>Postcondizioni:</b>
1. La playlist è stata aggiornata con successo.

## Rimuovi brano

<b>Caso d'uso:</b> Rimuovi brano
<b>ID:</b> RBR
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b>
1. L'utente registrato ha cliccato il tasto "Le mie playlist".
2. Il sistema ha mostrato a schermo la schermata contenente le playlist dell'utente registrato.
<b>Sequenza degli eventi:</b>
1. L'utente registrato seleziona una delle sue playlist.
2. Il sistema mostra a schermo la lista dei brani all'interno della playlist e l'icona "Rimuovi brano".
3. L'utente registrato clicca sull'icona "Rimuovi brano".
4. Il sistema invia al DBMS la richiesta di rimozione del brano dalla playlist.
5. Il DBMS rimuove il brano dalla playlist e comunica al sistema che l'operazione è avvenuta con successo.
6. Il sistema rimuove il brano dalla lista dei brani della playlist dell'utente registrato.
<b>Postcondizioni:</b>
1. La playlist è stata aggiornata con successo.

### 4.1.5. Gestione amicizia



#### Aggiungi amico

<b>Caso d'uso:</b> Aggiungi amico
<b>ID:</b> AAM
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente registrato ha cliccato il tasto "Gestione amicizie".</li><li>2. Il sistema ha mostrato a schermo la lista degli amici dell'utente registrato.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato clicca sul tasto "Aggiungi amici".</li><li>2. Il sistema mostra a schermo un pannello contenente un form di ricerca degli utenti registrati alla piattaforma.</li><li>3. L'utente registrato inserisce il nome, il cognome o il nome utente dell'utente da aggiungere.</li><li>4. Il sistema richiede al DBMS la lista degli utenti che soddisfano i criteri di ricerca.</li><li>5. Il DBMS invia la lista degli utenti.</li><li>6. Il sistema mostra a schermo la lista degli utenti trovati.</li><li>7. L'utente registrato clicca sull'icona "Aggiungi amico" relativa all'utente da aggiungere alla lista amici.</li><li>8. Il sistema chiede al DBMS di aggiornare la lista degli amici dell'utente registrato.</li><li>9. Il DBMS aggiunge l'utente cercato alla lista degli amici dell'utente registrato e comunica al sistema che l'operazione è avvenuta con successo.</li><li>10. Il sistema aggiunge l'utente alla lista degli amici.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente cercato è stato aggiunto alla lista degli amici.</li></ol>

## Rimuovi amico

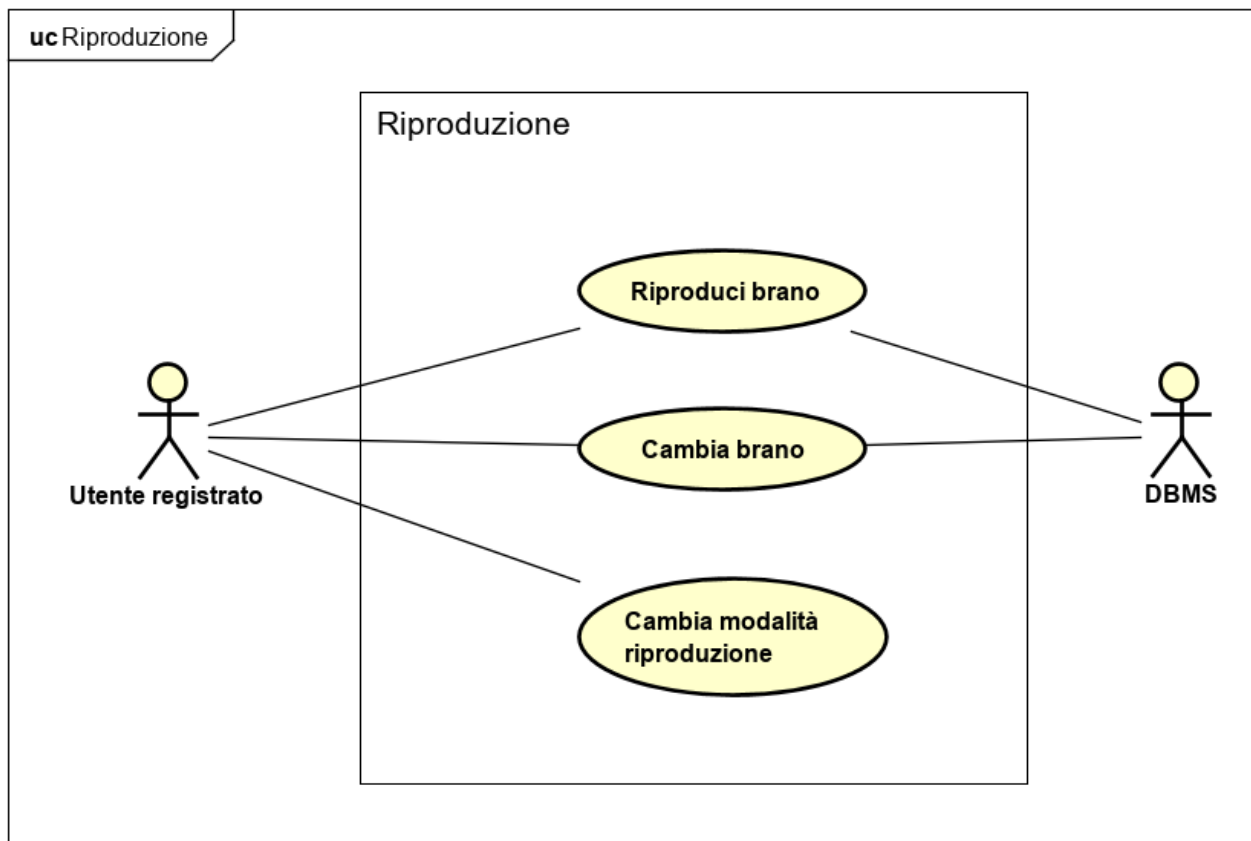
<b>Caso d'uso:</b> Rimuovi amico
<b>ID:</b> RAM
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente registrato ha cliccato il tasto "Gestione amicizie".</li><li>2. Il sistema ha mostrato a schermo la lista degli amici dell'utente registrato.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato clicca sul tasto "Rimuovi amico".</li><li>2. Il sistema mostra a schermo una notifica di conferma rimozione dell'amico.</li><li>3. L'utente registrato conferma la rimozione dell'amico dalla lista degli amici.</li><li>4. Il sistema chiede al DBMS di aggiornare la lista degli amici dell'utente registrato.</li><li>5. Il DBMS rimuove l'utente selezionato dalla lista degli amici dell'utente registrato e comunica al sistema che l'operazione è avvenuta con successo.</li><li>6. Il sistema rimuove l'utente dalla lista degli amici.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente cercato è stato rimosso dalla lista degli amici.</li></ol>

### 4.1.6. Gestione account

#### Gestione account

<b>Caso d'uso:</b> Gestione account
<b>ID:</b> GSA
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. L'utente registrato ha cliccato il tasto "Visualizza profilo".</li><li>2. Il sistema ha mostrato a schermo la un pannello contenente i dati utente dell'utente registrato.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato clicca sul tasto "Modifica" per rendere modificabili i campi, ne modifica il contenuto e clicca il tasto "Conferma modifiche".</li><li>2. Il sistema chiede al DBMS di aggiornare i dati utente dell'utente registrato.</li><li>3. Il DBMS aggiorna i dati e comunica al sistema che l'operazione è avvenuta con successo.</li><li>4. Il sistema mostra a schermo una notifica di avvenuta modifica dei dati utente.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. I dati dell'utente sono stati modificati.</li></ol>

#### 4.1.7. Riproduzione



#### Riproduci brano

<b>Caso d'uso:</b> Riproduci brano
<b>ID:</b> RBR
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. La pagina web del web player è stata visualizzata sul browser dell'utente registrato dopo che questo si è autenticato.</li><li>2. Il sistema ha mostrato una lista di brani.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato clicca sul tasto "Play" relativo ad un brano della lista.</li><li>2. Il sistema richiede al DBMS l'URL del brano da riprodurre.</li><li>3. Il DBMS invia l'URL relativo al brano al sistema.</li><li>4. Il sistema avvia lo streaming del brano.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. La riproduzione è stata inviata correttamente.</li></ol>

## Cambia brano

<b>Caso d'uso:</b> Cambia brano
<b>ID:</b> CBR
<b>Attori:</b> Utente registrato, DBMS.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. La pagina web del web player è stata visualizzata sul browser dell'utente registrato dopo che questo si è autenticato.</li><li>2. Il sistema ha mostrato una lista di brani.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato seleziona un nuovo brano da riprodurre dalla lista di brani oppure clicca sul tasto "Brano successivo" o "Brano precedente".</li><li>2. Il sistema manda il brano in riproduzione il brano richiesto, e se il brano fa parte di una particolare lista di brani imposta quella lista come sequenza di riproduzione.</li><li>3. Il sistema avvia lo streaming del nuovo brano, avviando il caso d'uso "Riproduci brano".</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. Il brano in riproduzione è stato cambiato correttamente.</li></ol>

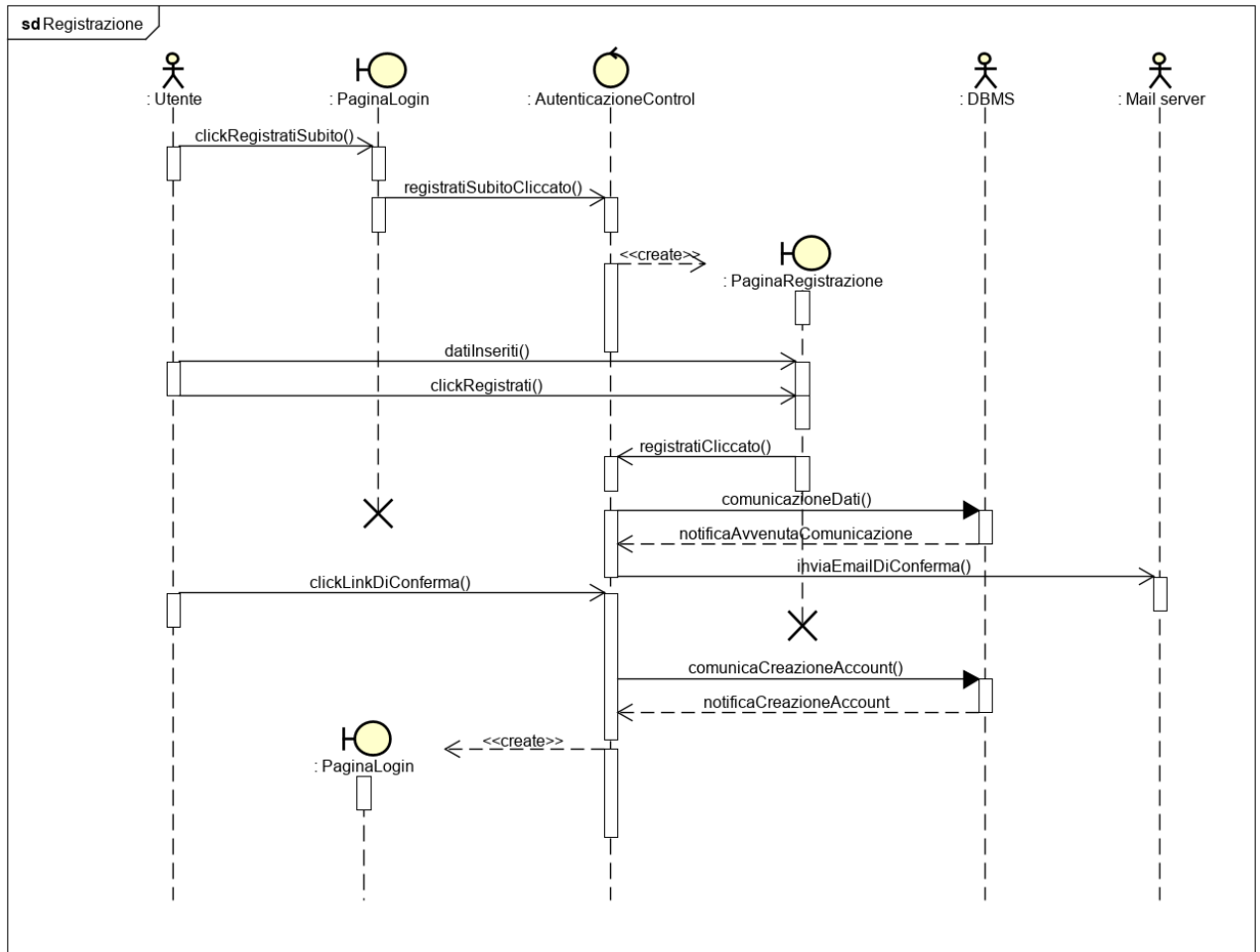
## Cambia modalità riproduzione

<b>Caso d'uso:</b> Cambia modalità riproduzione
<b>ID:</b> CMR
<b>Attori:</b> Utente registrato.
<b>Precondizioni:</b> <ol style="list-style-type: none"><li>1. La pagina web del web player è stata visualizzata sul browser dell'utente registrato dopo che questo si è autenticato.</li><li>2. Il sistema ha avviato la riproduzione di un brano.</li></ol>
<b>Sequenza degli eventi:</b> <ol style="list-style-type: none"><li>1. L'utente registrato cambia la modalità di riproduzione cliccando su "Shuffle" o "Ripeti sequenza".</li><li>2. Il sistema modifica la modalità di riproduzione.</li></ol>
<b>Postcondizioni:</b> <ol style="list-style-type: none"><li>1. La modalità di riproduzione è stata modificata correttamente.</li></ol>

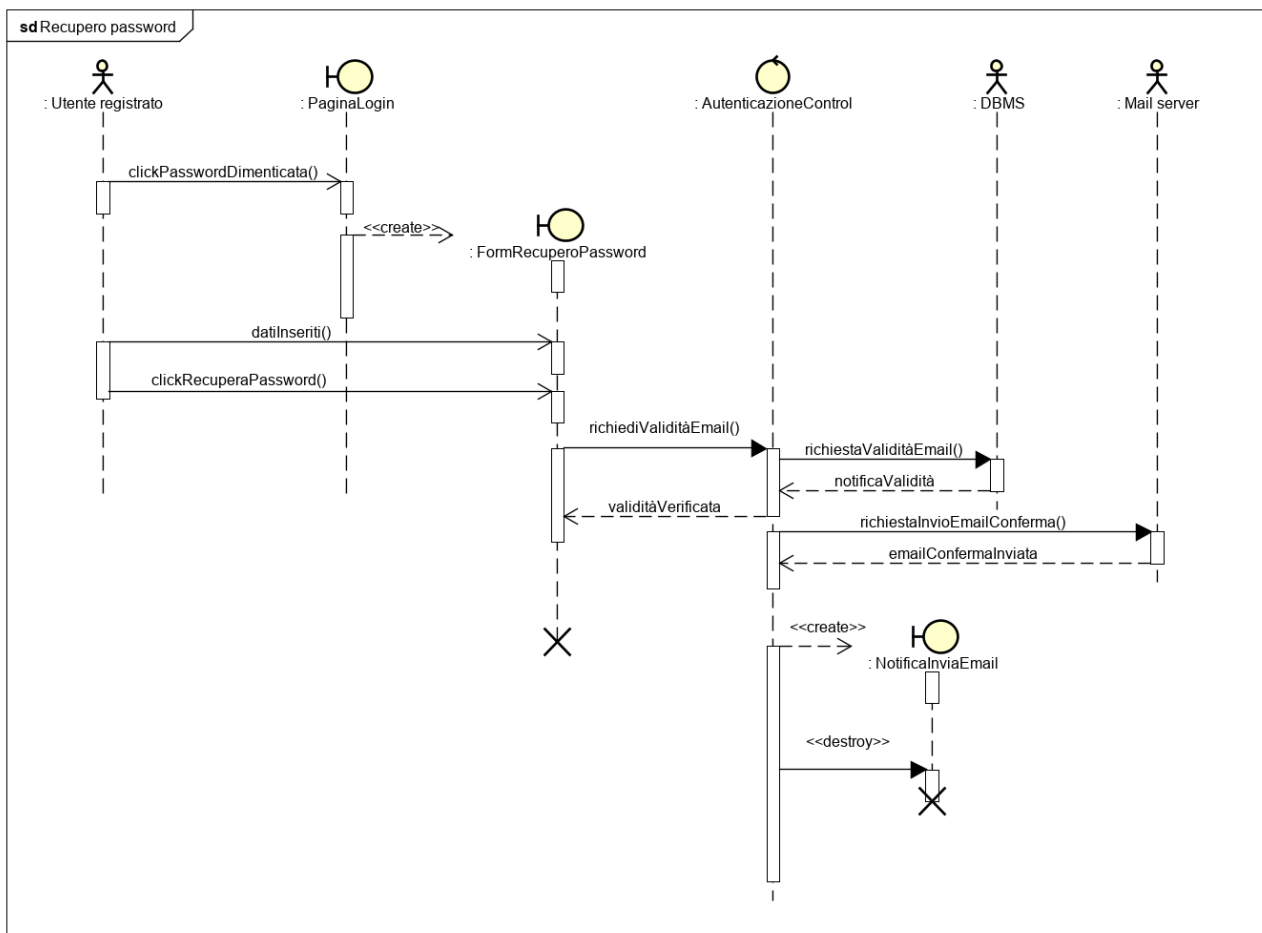


## 4.2. Diagrammi di sequenza

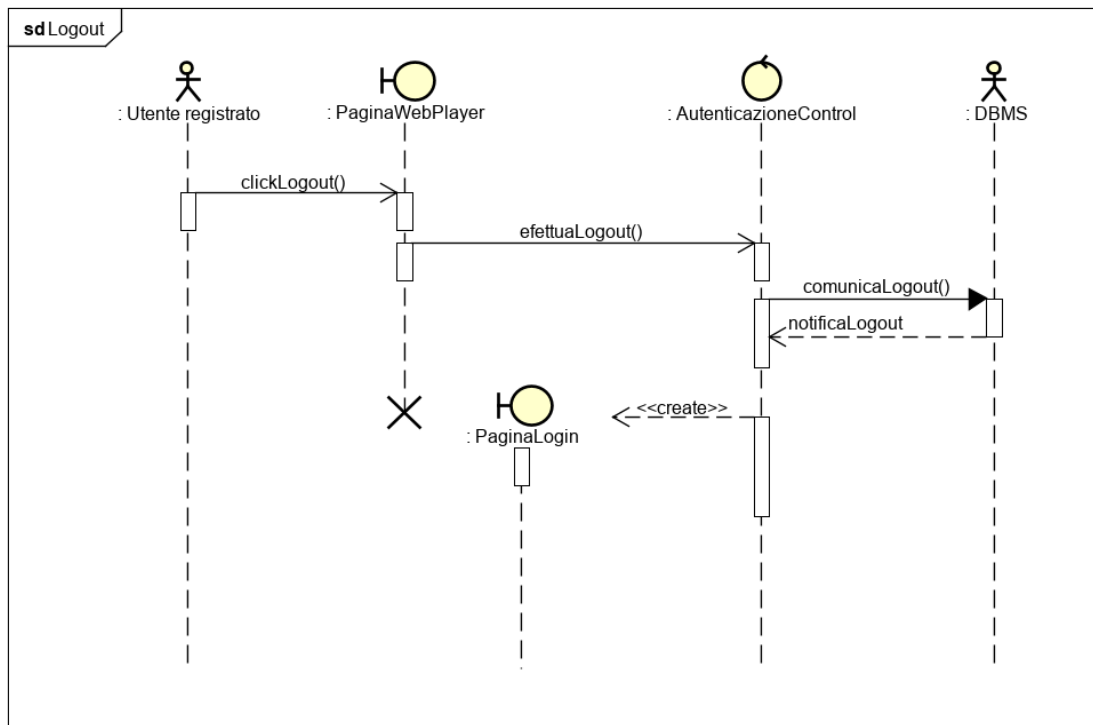
### Registrazione



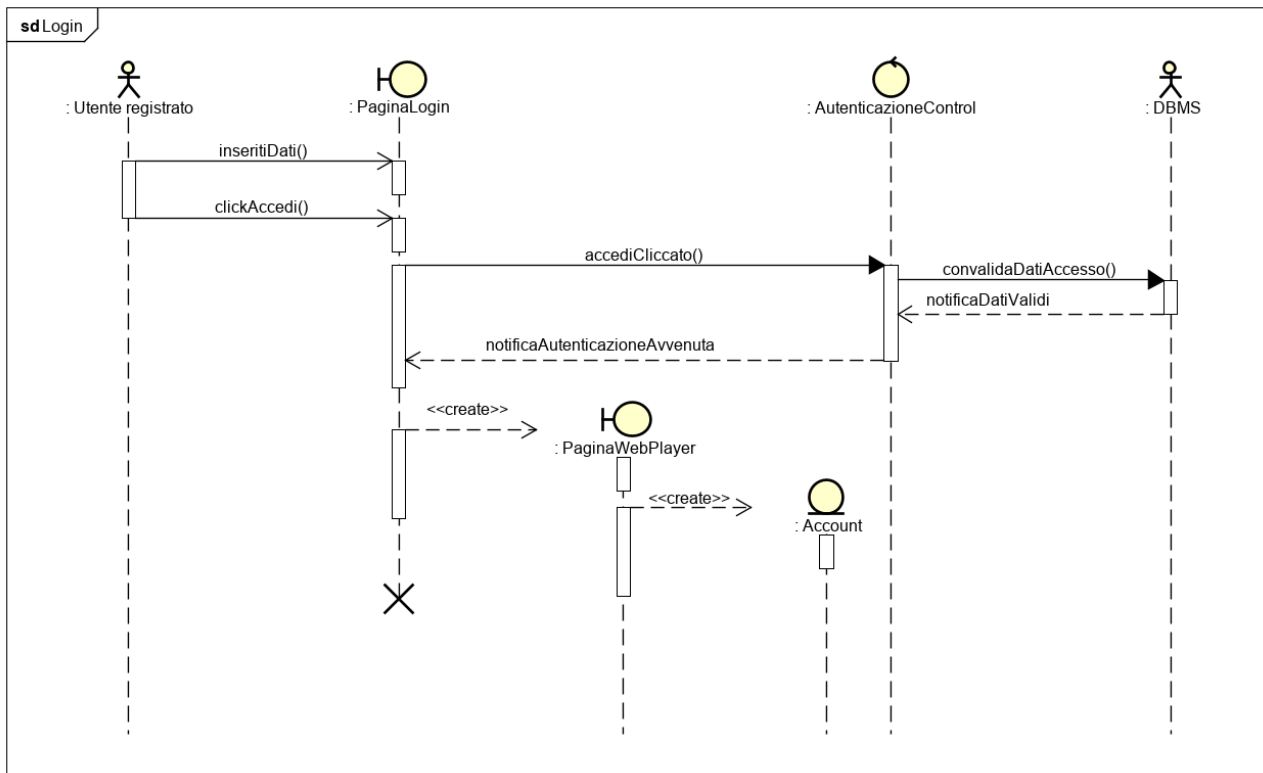
## Recupero password



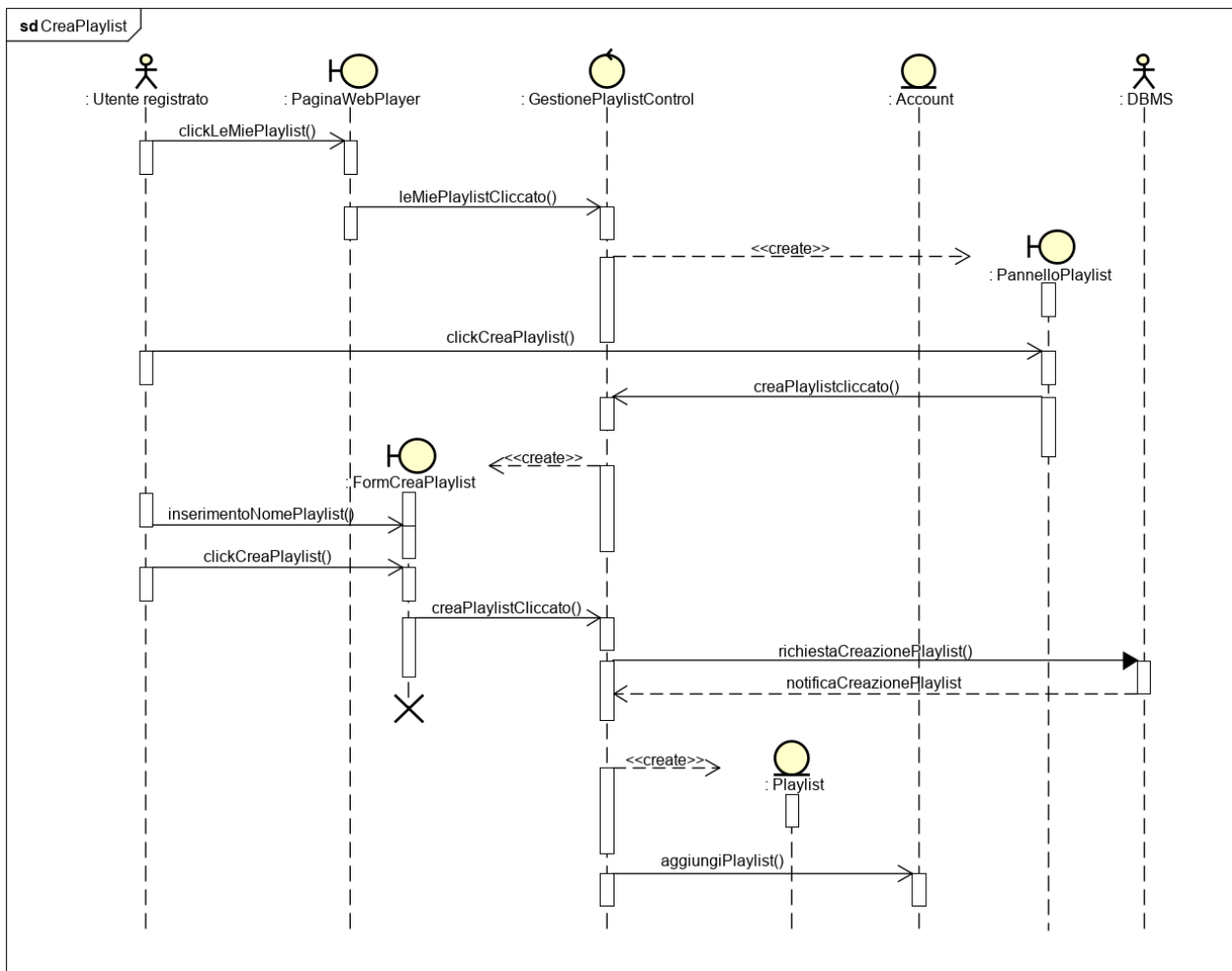
## Logout



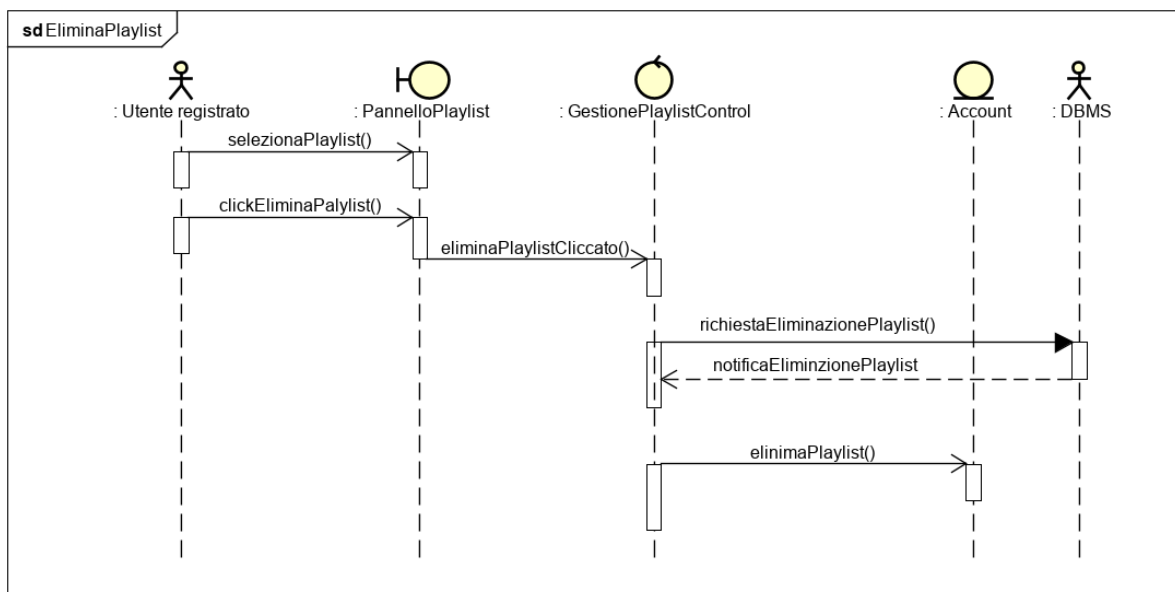
## Login



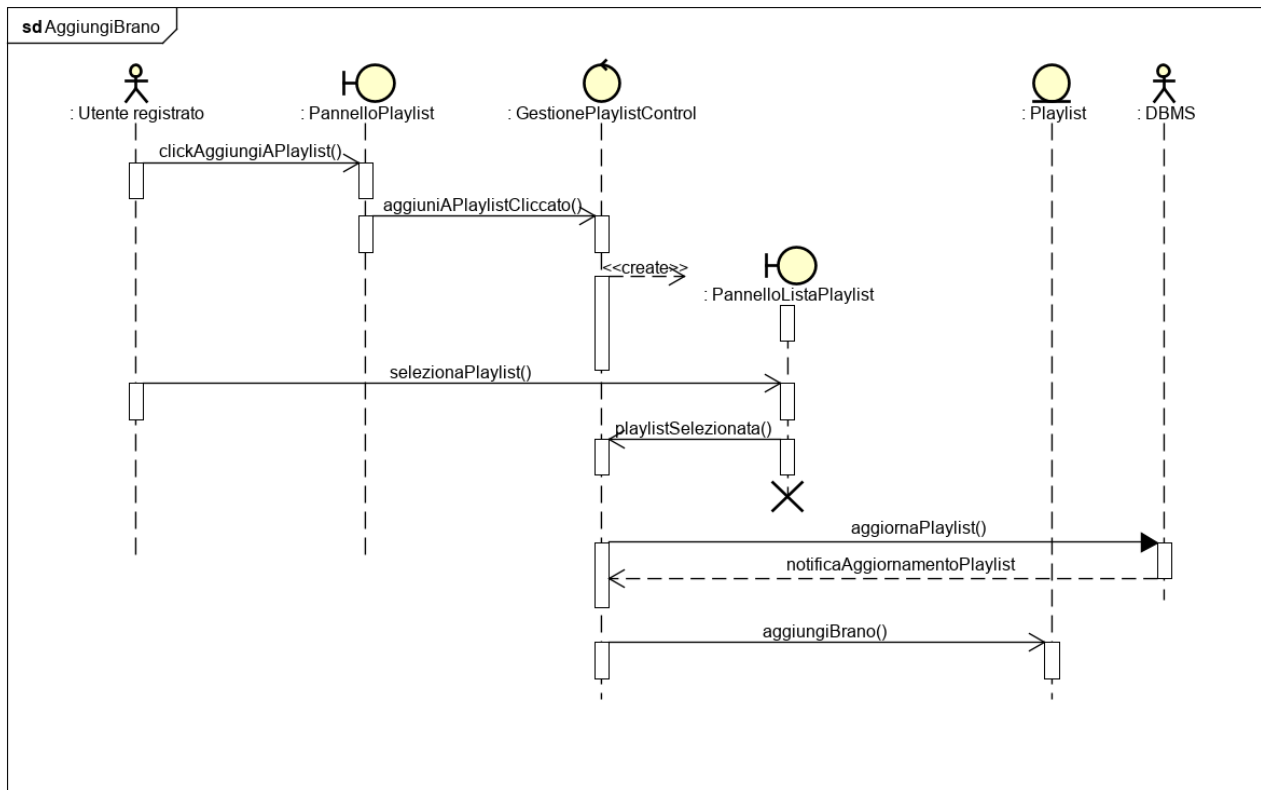
## Crea playlist



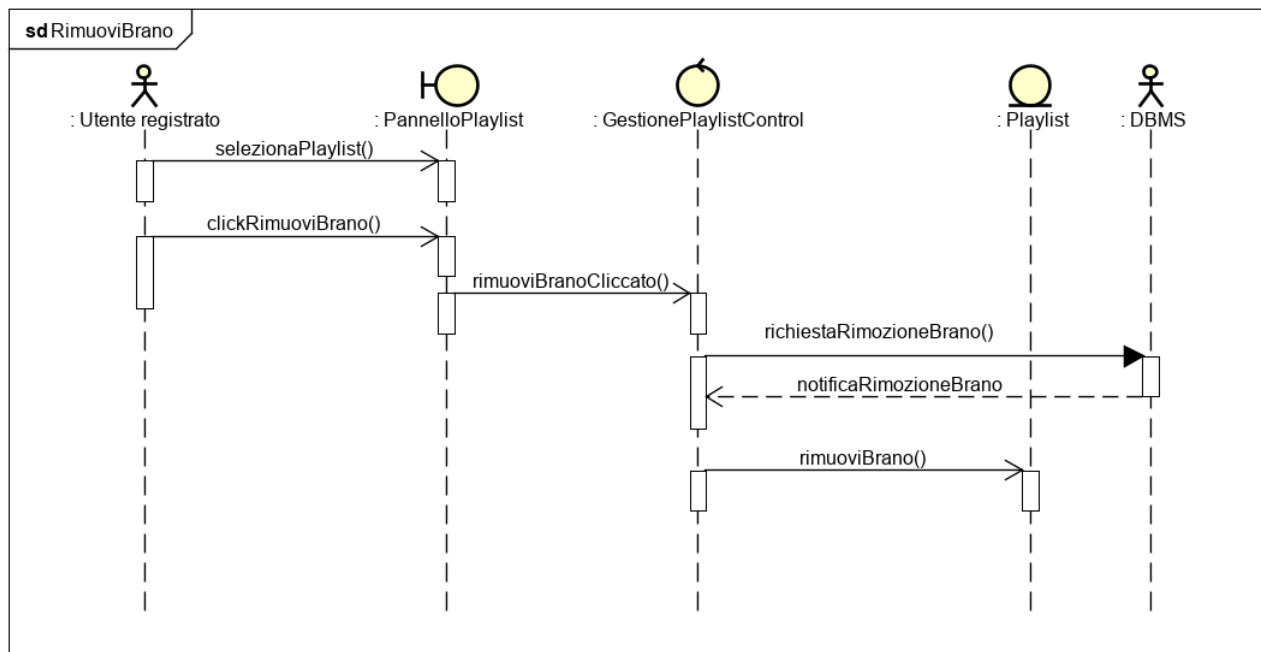
## Elimina playlist



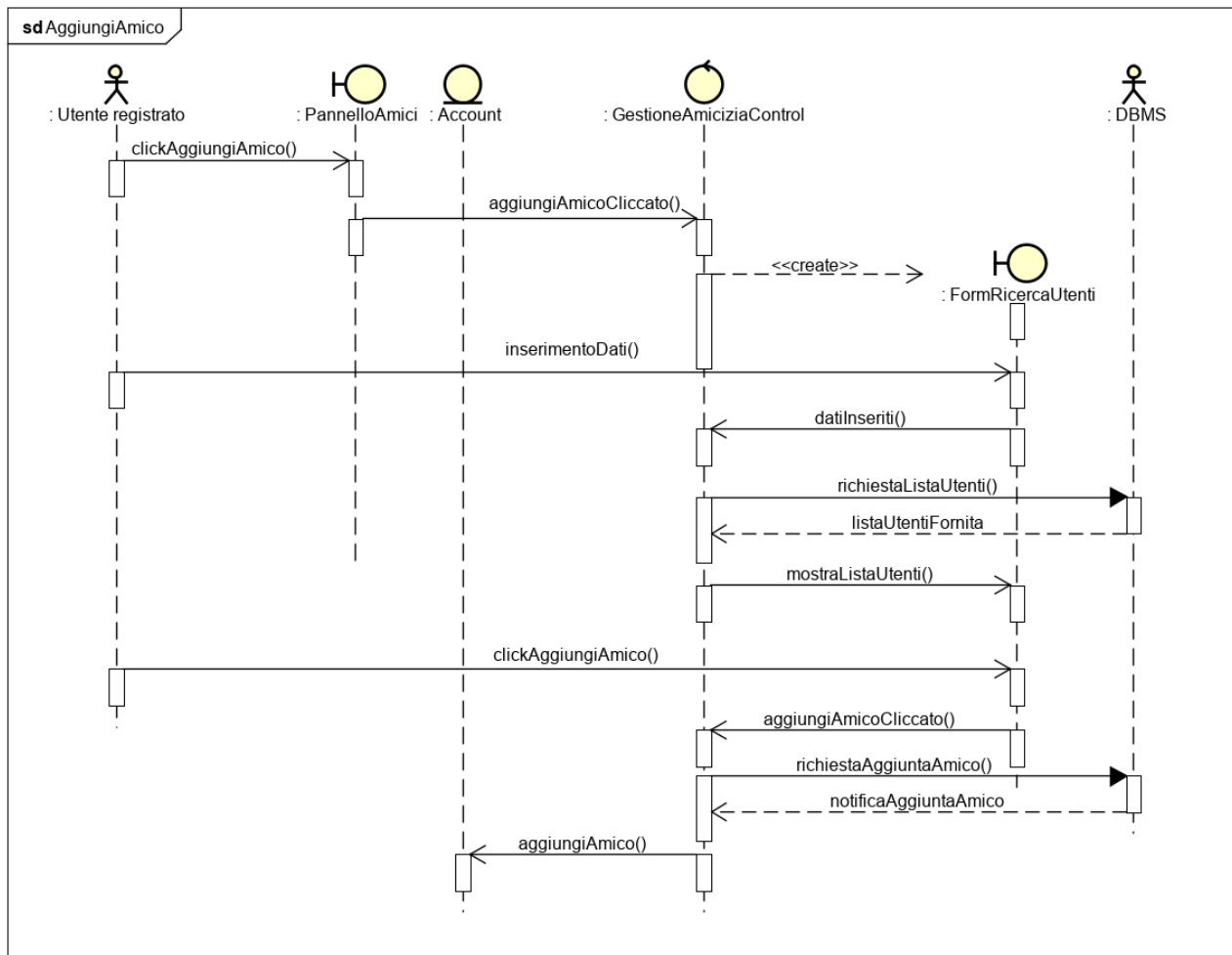
## Aggiungi brano



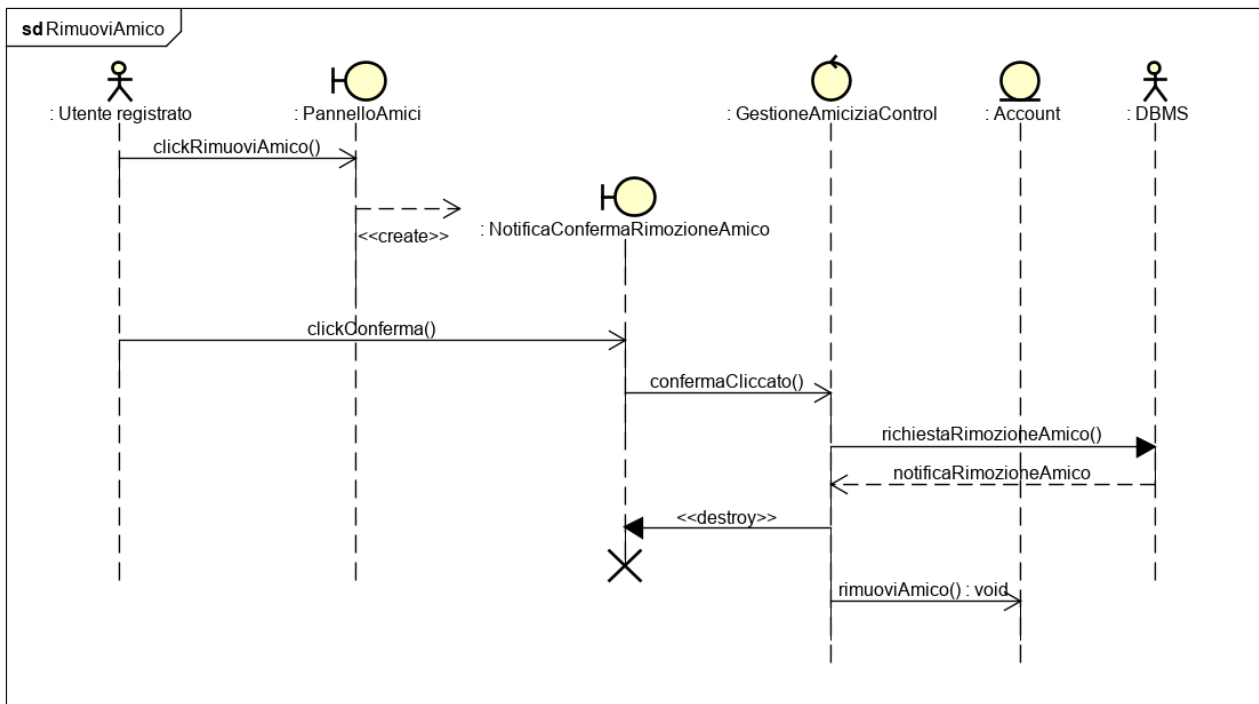
## Rimuovi brano



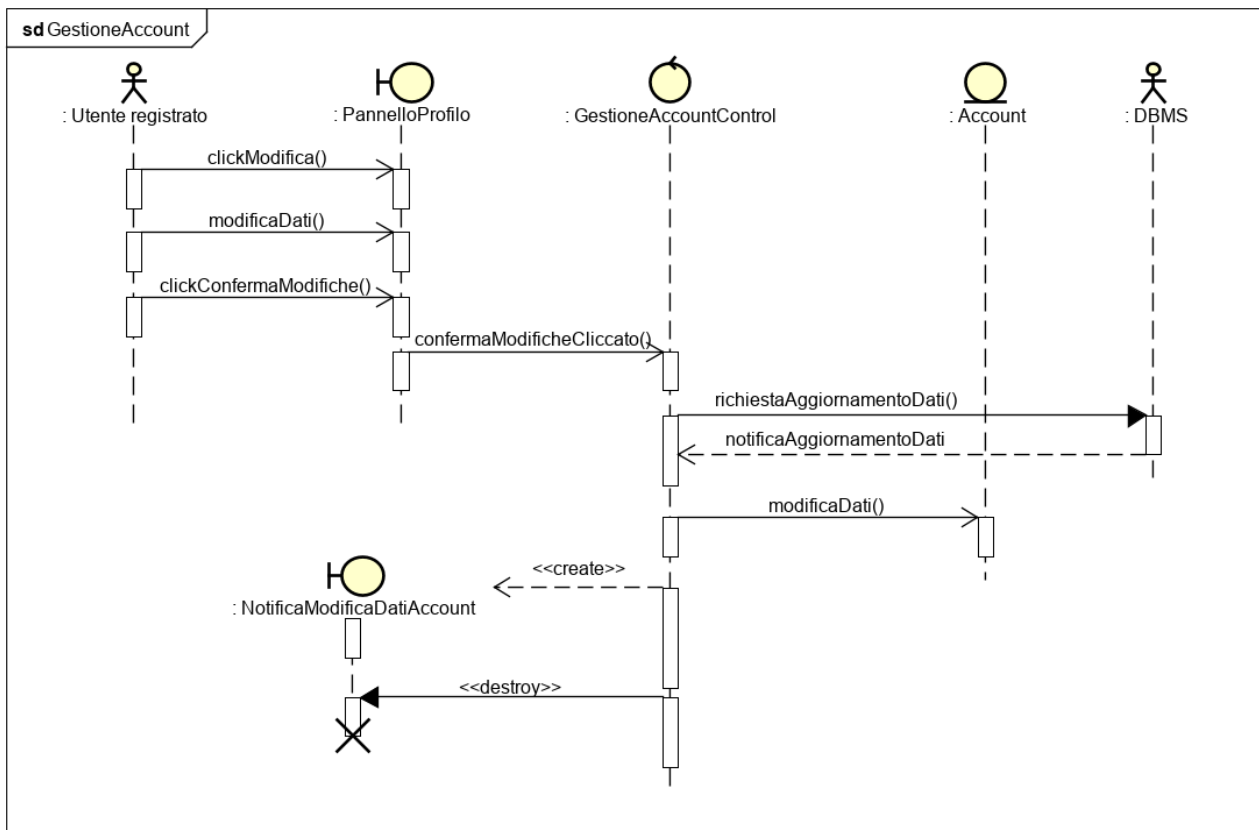
## Aggiungi amico



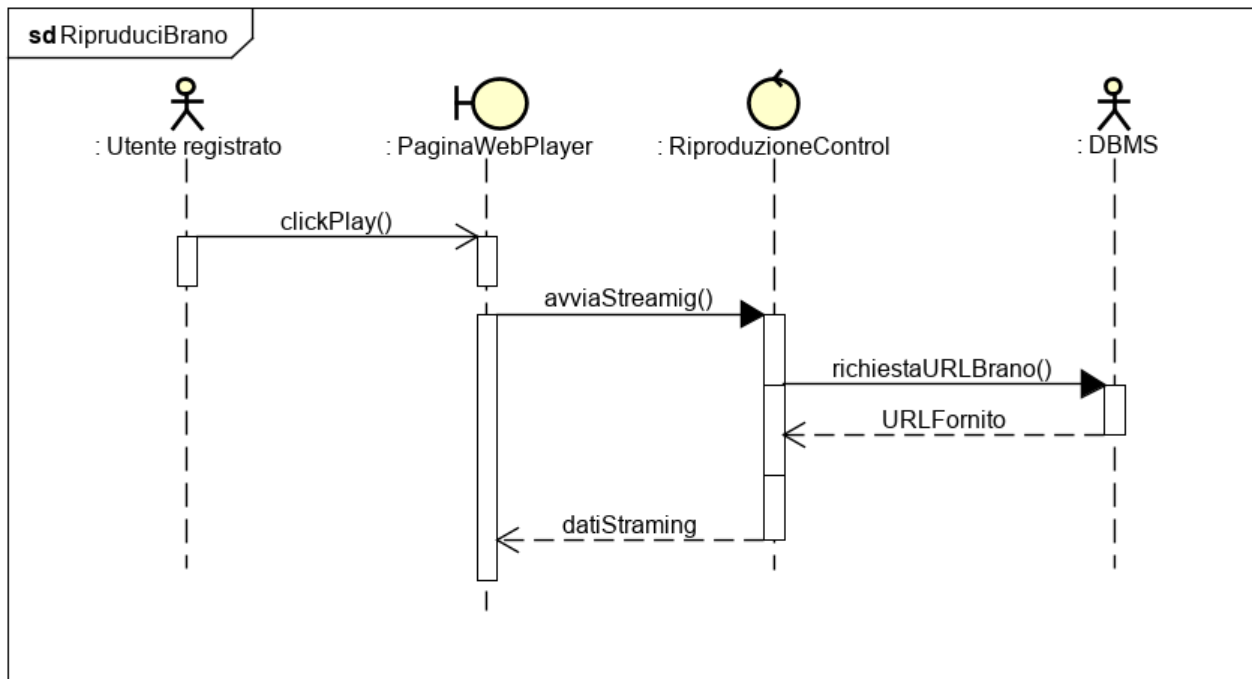
## Rimuovi amico



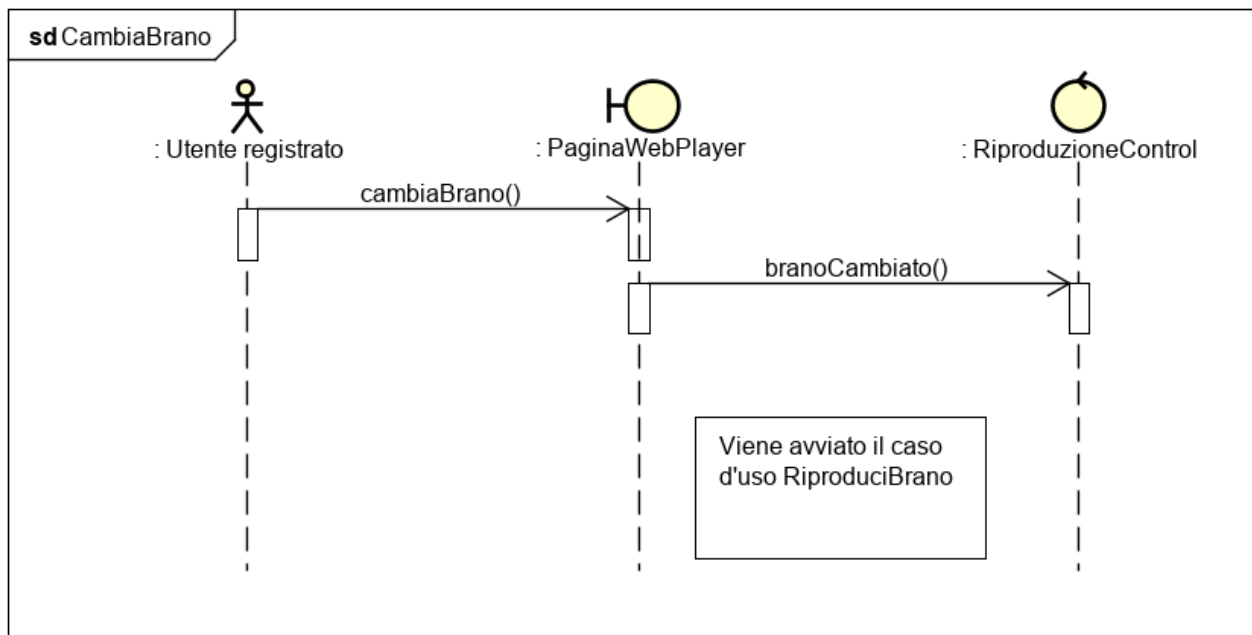
## Gestione account



## Riproduci brano

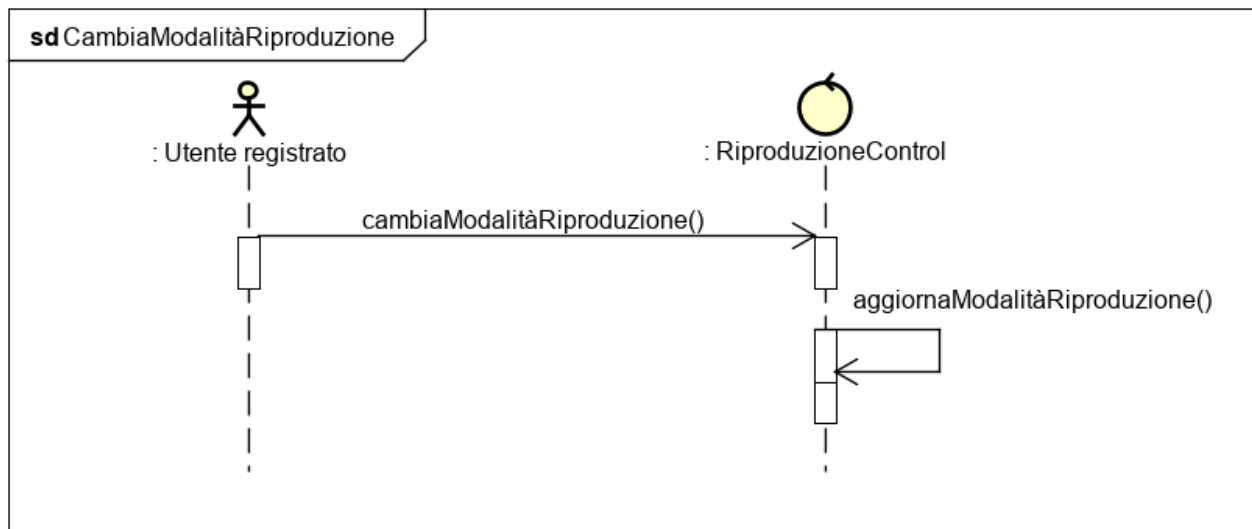


## Cambia brano



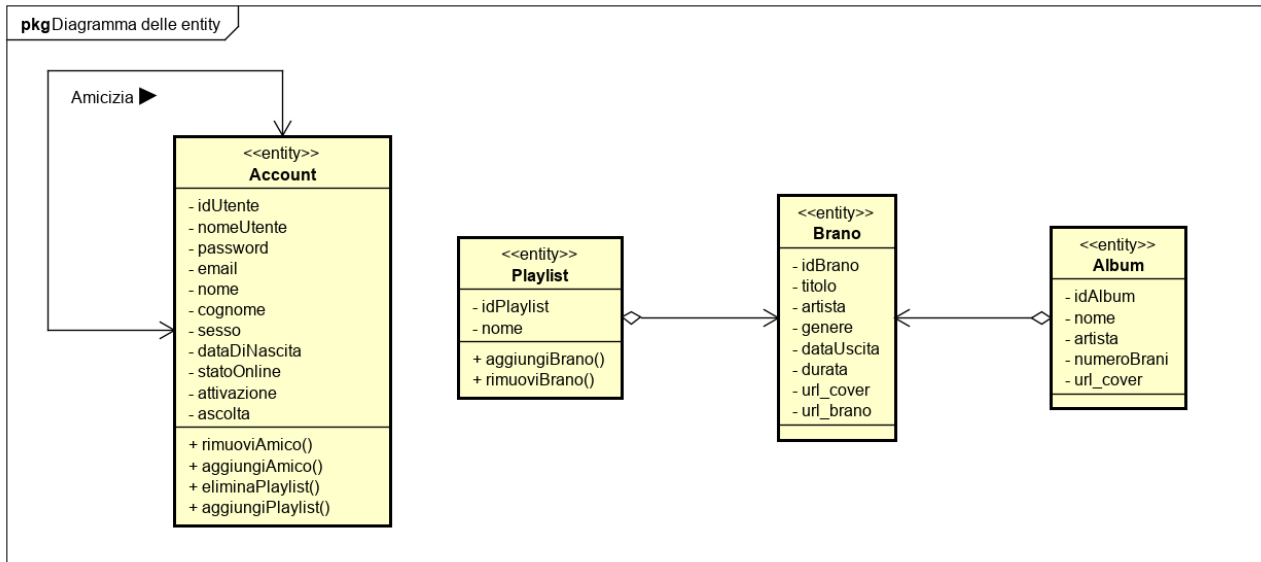


## Cambia modalità riproduzione

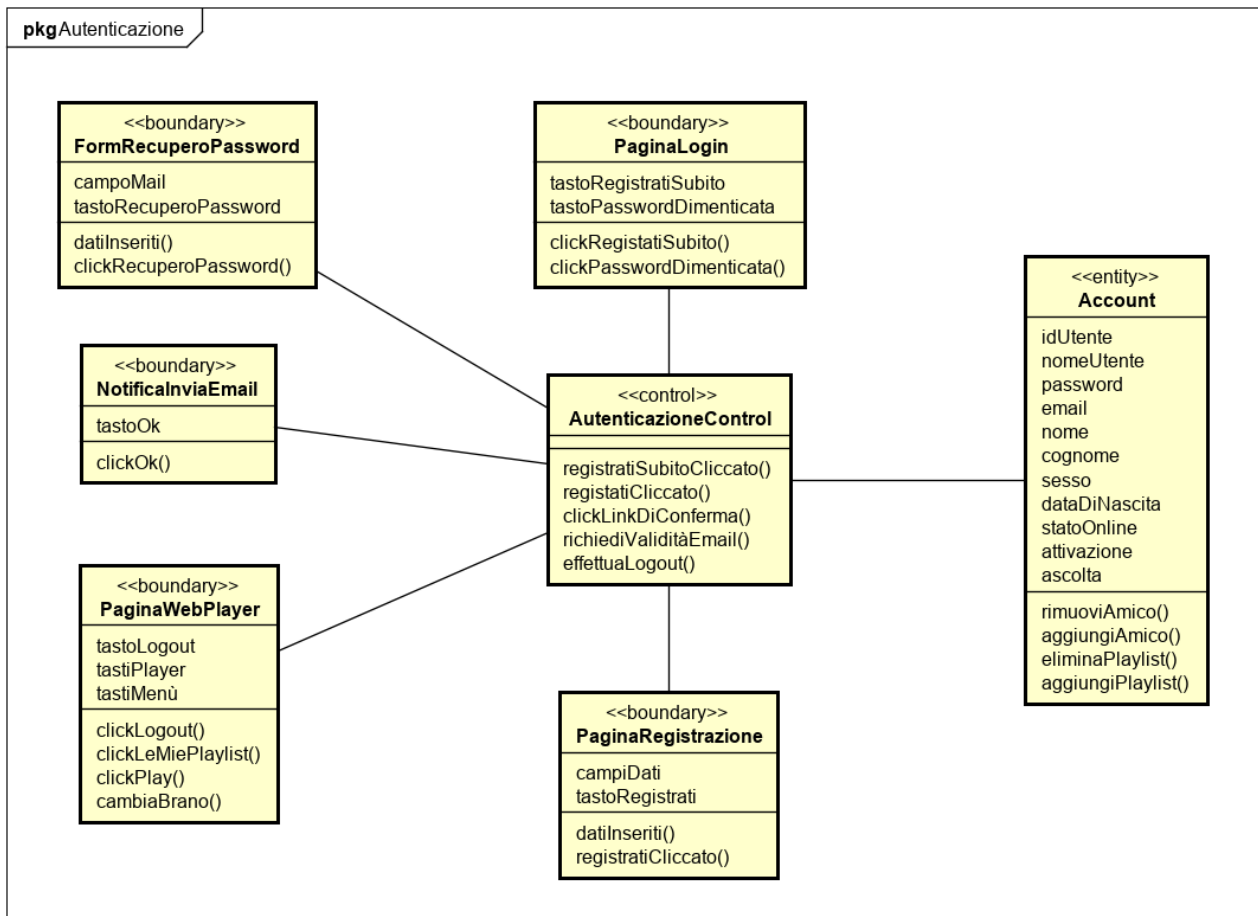


## 4.3. Diagrammi delle classi

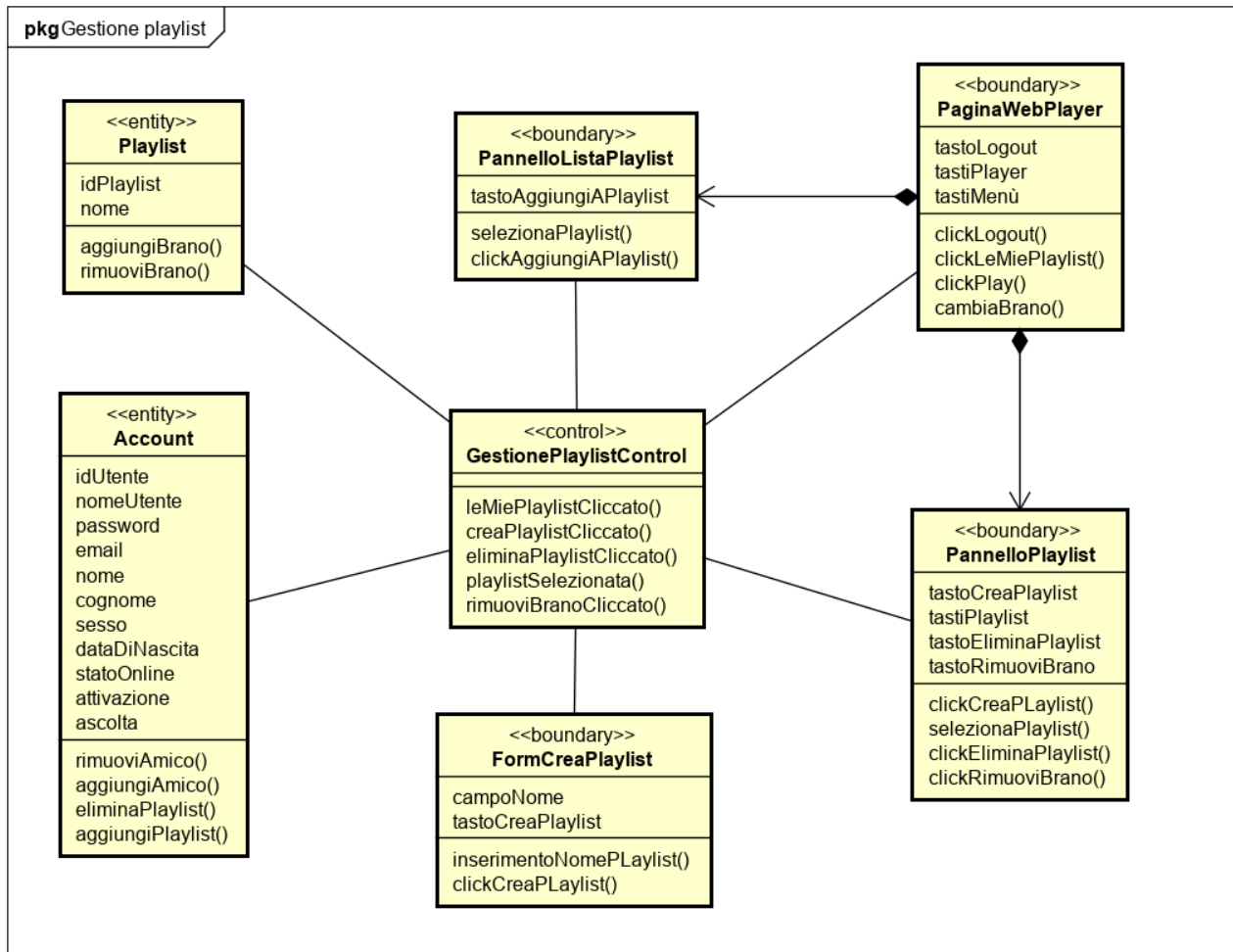
### Diagramma delle entity



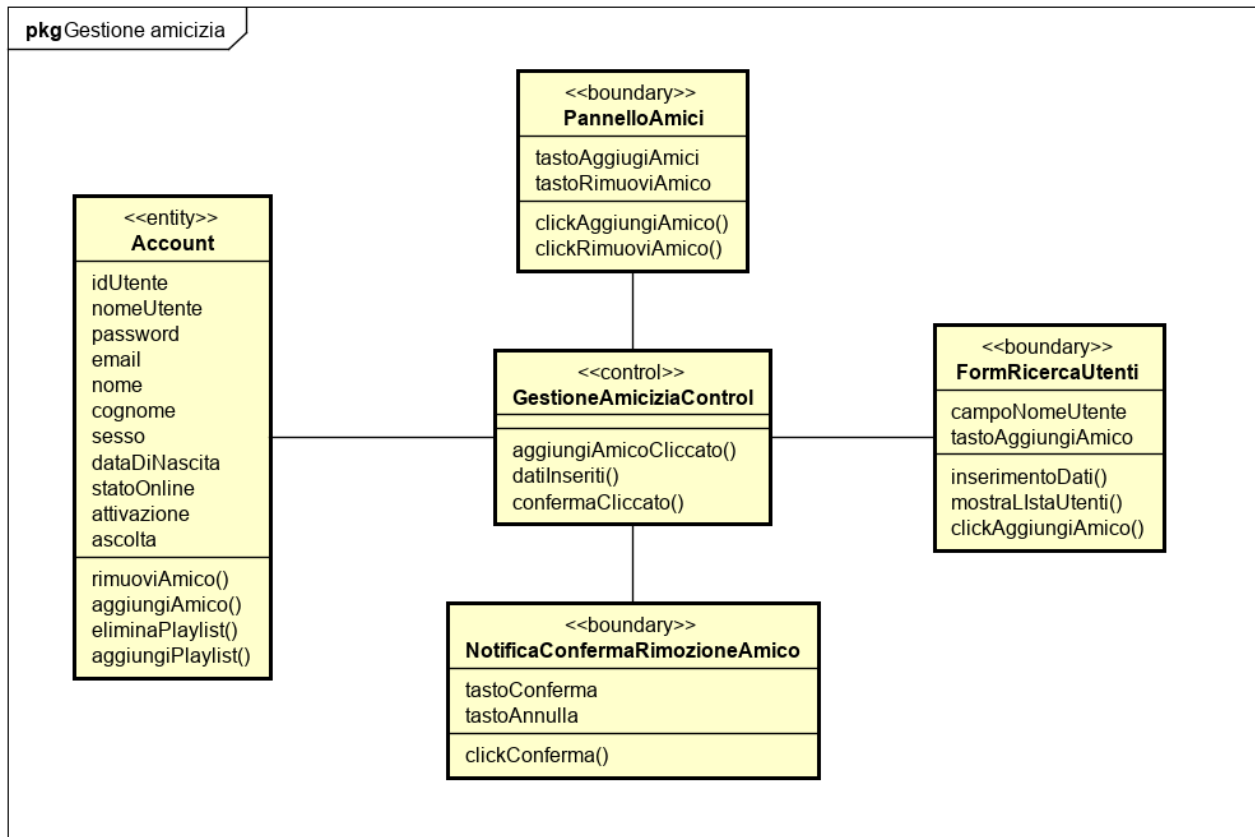
# Autenticazione



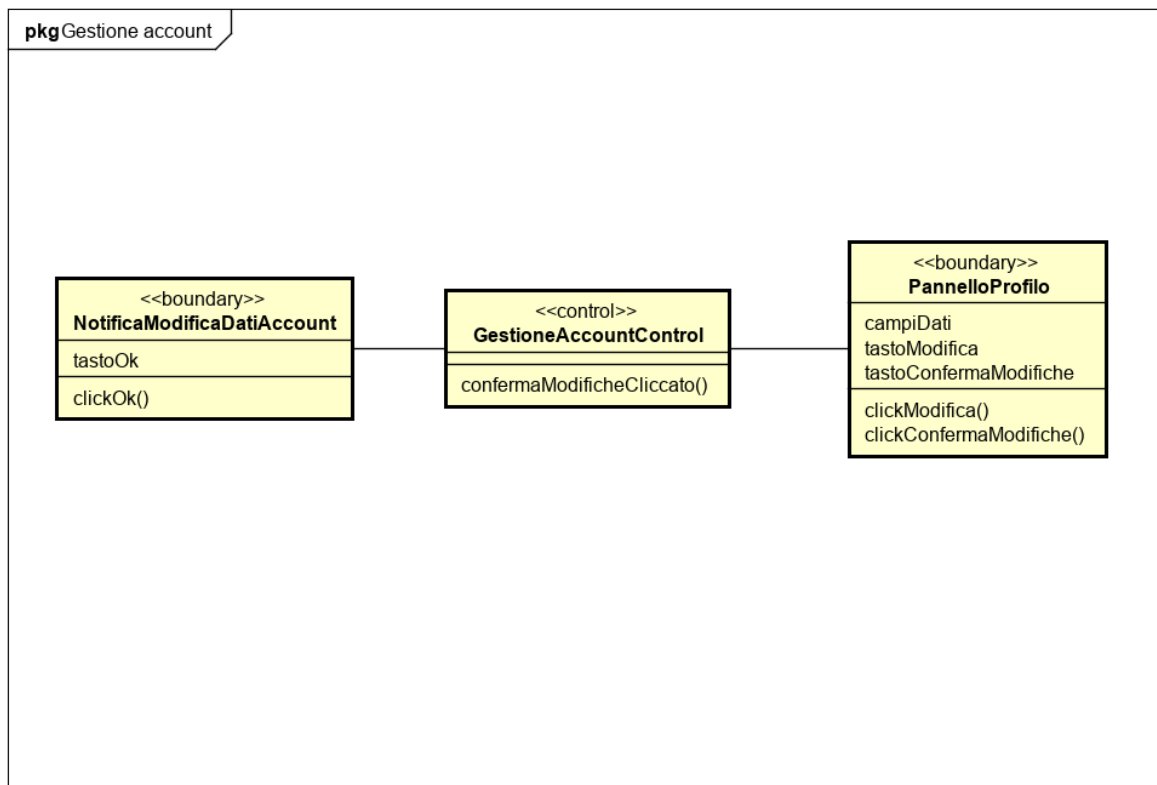
## Gestione playlist



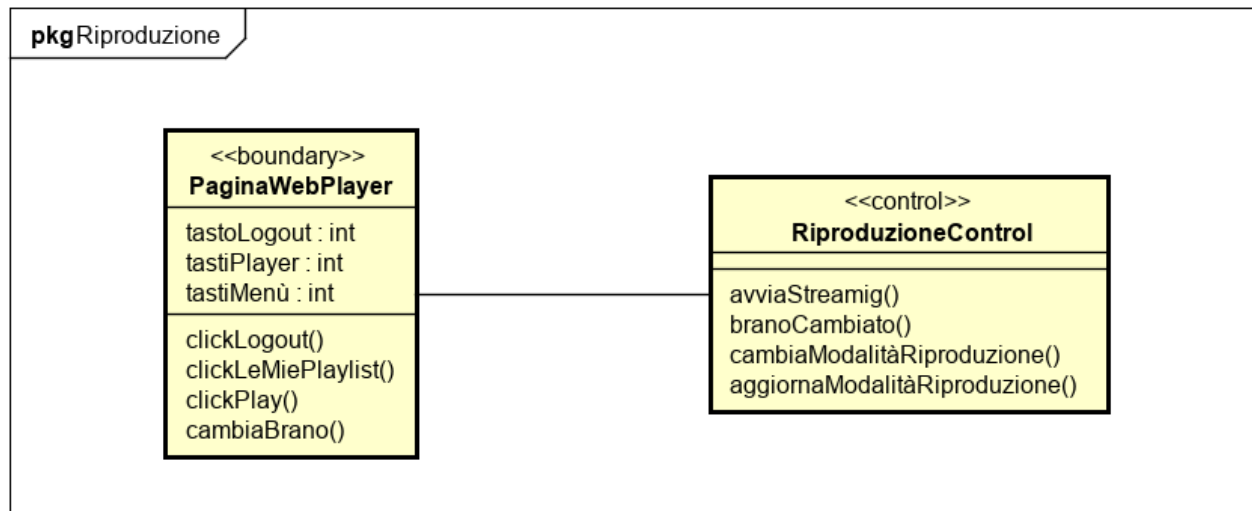
## Gestione amicizia



## Gestione Account



## Riproduzione



## **5. Architettura software proposta**

### **5.1. Panoramica generale**

Il sistema proposto è strutturato in un'architettura di tipo client-server. I nodi client sono tutti i dispositivi degli utenti che accedono al sito web dell'applicazione SoundWave o gli smartphone su cui è installata l'app mobile. Il nodo server è unico e consiste nel vero e si occupa di fornire tutte le informazioni e i servizi richiesti dai client che usufruiscono della piattaforma.

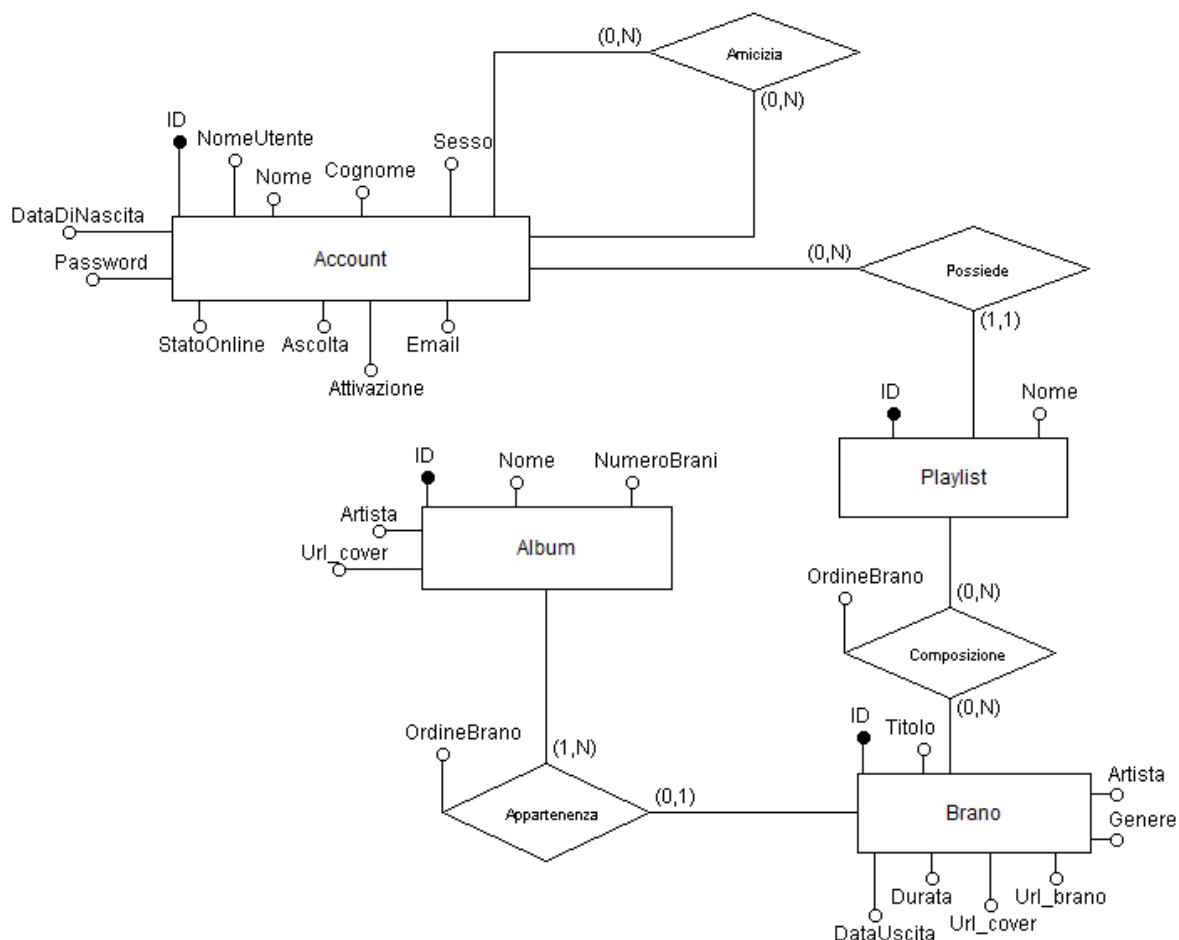
### **5.2. Database**

#### **5.2.1. Descrizione verbale del database**

Un utente registrato al sistema ha un account contenente le proprie informazioni personali: nome utente, indirizzo e-mail (diversi da quelli degli altri utenti), nome, cognome, data di nascita e sesso. Per quanto riguarda le funzionalità social offerte dal sistema, gli utenti possono aggiungere altri utenti agli amici, visualizzare se essi sono online e il loro brano attualmente in riproduzione. Inoltre un utente può accedere al sistema esclusivamente se ha attivato l'account tramite il link di attivazione ricevuto per e-mail.

Un utente può creare le sue playlist personali (caratterizzate da un numero identificativo e da un nome) ed inserirvi tutti i brani che desidera. Questi ultimi specificano titolo, artista, genere di appartenenza, data di uscita, durata in secondi e una cover. Inoltre un brano può appartenere o meno ad un album, il quale ha titolo, artista, numero di brani contenuti e una cover.

### 5.2.2. Diagramma entità-relazione



### 5.2.3. Descrizione delle tabelle

Qui di seguito sono riportate tutte le tabelle del database derivanti da una traduzione del modello entità-relazione.

#### Account

Nome colonna	Tipo	Vincoli	Descrizione
IDUtente	INT	PK, NOT NULL, AUTO_INCREMENT	Identificativo dell'account utente
NomeUtente	VARCHAR(45)	NOT NULL, UNIQUE	Nickname dell'utente
Password	VARCHAR(45)	NOT NULL	Password dell'utente tramite la quale può accedere

Email	VARCHAR(45)	NOT NULL, UNIQUE	Indirizzo e-mail dell'utente
Nome	VARCHAR(45)	NOT NULL	Nome dell'utente
Cognome	VARCHAR(45)	NOT NULL	Cognome dell'utente
Sesso	CHAR(1)	NOT NULL	Genere dell'utente
DataDiNascita	DATE	NOT NULL	Data di nascita dell'utente
StatoOnline	BIT(1)	NOT NULL, DEFAULT 0	Booleano che indica se l'utente è attualmente online o meno
Attivazione	BIT(1)	NOT NULL, DEFAULT 0	Booleano che indica se l'utente ha attivato il suo account
Ascolta	VARCHAR(45)	NOT NULL, DEFAULT “~”	Brano attualmente in riproduzione dall'utente

## Brano

Nome colonna	Tipo	Vincoli	Descrizione
IDBrano	INT	PK, NOT NULL, AUTO_INCREMENT	Numero identificativo del brano
Titolo	VARCHAR(45)	NOT NULL	Titolo del brano
Artista	VARCHAR(45)	NOT NULL	Artista che ha composto il brano
Genere	VARCHAR(45)	NOT NULL	Genere di appartenenza del brano
DataUscita	DATE	NOT NULL	Data di pubblicazione del brano
Durata	INT UNSIGNED	NOT NULL	Durata in secondi del brano
Url_cover	VARCHAR(150)	NOT NULL, DEFAULT URL	Indirizzo utile a trovare la cover



		COVER DI DEFAULT DEL BRANO	del brano nel server.
Url_brano	VARCHAR(45)	NOT NULL	Indirizzo utile a trovare il file audio nel server.

## Album

Nome colonna	Tipo	Vincoli	Descrizione
IDAlbum	INT	PK, NOT NULL, AUTO_INCREMENT	Numero identificativo dell'album
Nome	VARCHAR(45)	NOT NULL	Nome dell'album
Artista	VARCHAR(45)	NOT NULL	Artista che ha composto l'album
NumeroBrani	INT	NOT NULL	Numero di brani contenuti all'interno dell'album
Url_cover	VARCHAR(150)	NOT NULL, DEFAULT URL COVER DI DEFAULT DELL'ALBUM	Indirizzo utile a trovare la cover dell'album nel server.

## Playlist

Nome colonna	Tipo	Vincoli	Descrizione
IDPlaylist	INT	PK, NOT NULL, AUTO_INCREMENT	Numero identificativo della playlist
Nome	VARCHAR(45)	NOT NULL	Nome della playlist

## Amicizia

Nome colonna	Tipo	Vincoli	Descrizione
Ref1_IDUtente	INT	NOT NULL	Identificativo dell'account utente che ha aggiunto l'amico

Ref2_IDUtente	INT	NOT NULL	Identificativo dell'account utente che è stato aggiunto agli amici
---------------	-----	----------	--

Altri vincoli:

PK(Ref1\_IDUtente, Ref2\_IDUtente)

FOREIGN KEY(Ref1\_IDUtente) REFERENCES Account(IDUtente)

FOREIGN KEY(Ref2\_IDUtente) REFERENCES Account(IDUtente)

### Possiede

Nome colonna	Tipo	Vincoli	Descrizione
Ref_IDUtente	INT	NOT NULL	Identificativo dell'account utente possiede la playlist
Ref_IDPlaylist	INT	NOT NULL	Identificativo della playlist associata all'utente

Altri vincoli:

PK(Ref\_IDPlaylist, Ref\_IDUtente)

FOREIGN KEY(Ref\_IDPlaylist) REFERENCES Playlist(IDPlaylist)

FOREIGN KEY(Ref\_IDUtente) REFERENCES Account(IDUtente)

### Composizione

Nome colonna	Tipo	Vincoli	Descrizione
OrdineBranco	INT	NOT NULL	Numero che indica l'ordine del brano all'interno della playlist
Ref_IDPlaylist	INT	NOT NULL	Identificativo della playlist che contiene il brano

Ref_IDBranco	INT	NOT NULL	Identificativo del brano che è contenuto all'interno della playlist
--------------	-----	----------	---

Altri vincoli:

PK(OrdineBrano, Ref\_IDPlaylist)

FOREIGN KEY(Ref\_IDPlaylist) REFERENCES Playlist(IDPlaylist)

FOREIGN KEY(Ref\_IDBranco) REFERENCES Brano(IDBranco)

### Appartenenza

Nome colonna	Tipo	Vincoli	Descrizione
OrdineBrano	INT	NOT NULL	Numero che indica l'ordine del brano all'interno dell'album
Ref_IDAlbum	INT	NOT NULL	Identificativo dell'album che contiene il brano
Ref_IDBranco	INT	NOT NULL	Identificativo del brano che è contenuto all'interno dell'album

Altri vincoli:

PK(OrdineBrano, Ref\_IDAlbum)

FOREIGN KEY(Ref\_IDAlbum) REFERENCES Album(IDAlbum)

FOREIGN KEY(Ref\_IDBranco) REFERENCES Brano(IDBranco)

All'interno del database si fa inoltre uso di una vista "Brani\_Playlist" la quale contiene i dati dell'utente, della playlist e dei brani contenuti in quella playlist, per agevolare le richieste ed ottenere i dati in maniera più efficiente.

## 5.2.4. Vincoli di tupla

### Account

1. “NomeUtente” deve iniziare con una lettera, contenere minimo due caratteri e può contenere anche numeri.
2. “Password” deve contenere minimo sei caratteri, di cui almeno una lettera maiuscola e un numero.
3. “Email” deve essere una stringa che rispetti il formato [“nome@servizio.dominio”](#).
4. “Nome” e “Cognome” possono contenere solo lettere e devono iniziare con una lettera maiuscola.
5. “DataDiNascita” deve essere antecedente alla data attuale e con almeno “18 anni” di distanza.

### Brano

1. “DataUscita” deve essere antecedente alla data attuale.
2. “Durata” deve essere un intero maggiore di zero.
3. “Url\_cover” e “Url\_branco” devono essere delle stringhe di indirizzi valide.

### Album

1. “NumeroBrani” deve essere un intero maggiore di zero.
2. “Url\_cover” deve essere una stringa di indirizzo valida.

### Playlist

1. “Nome” non può essere duplicato per uno stesso utente (un utente non può possedere due playlist aventi lo stesso nome).

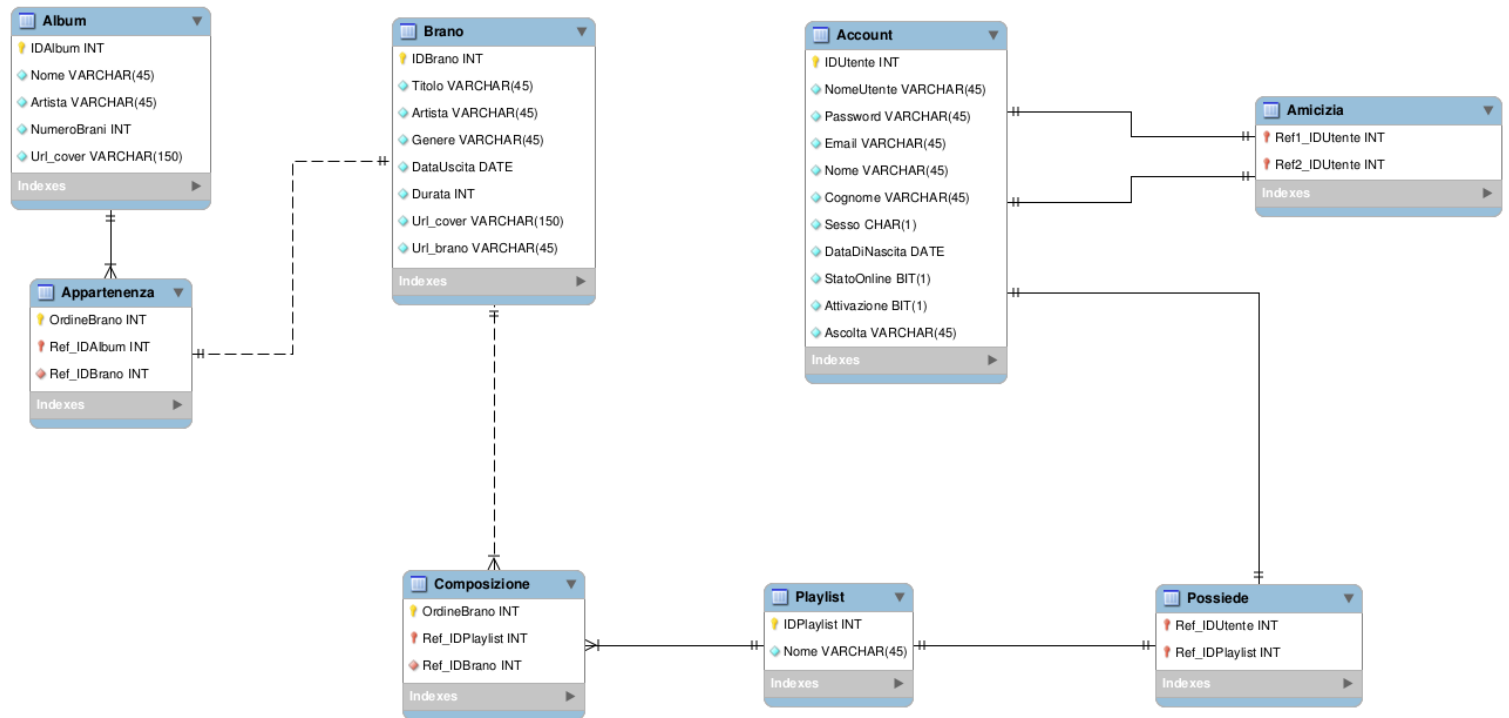
### Composizione

1. “OrdineBrano” deve essere un intero maggiore di zero.
2. “Ref\_IDBrano” deve essere diverso per ogni coppia OrdineBrano, Ref\_IDPlaylist, in quanto una playlist non può contenere più volte lo stesso brano.

### Appartenenza

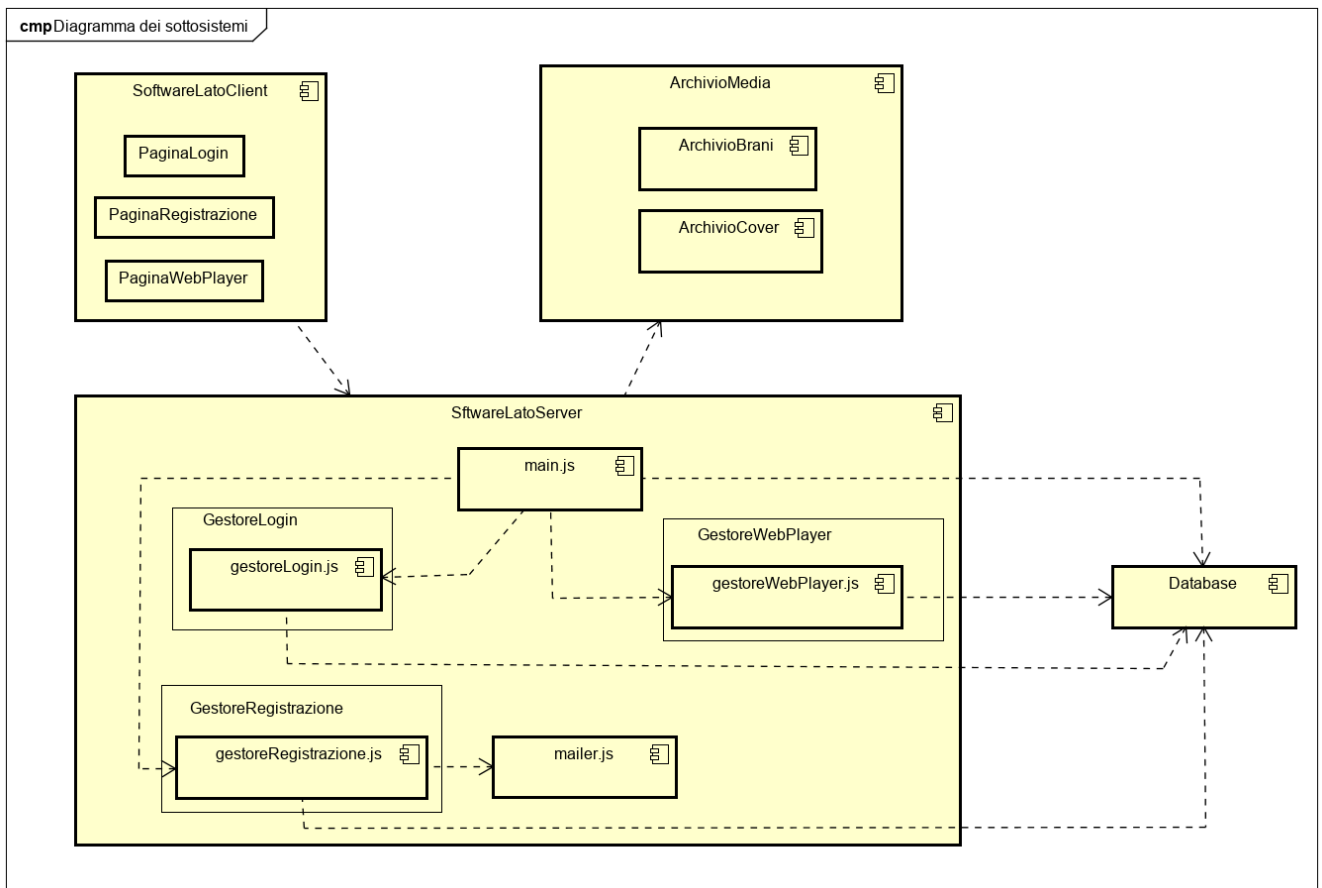
1. “OrdineBrano” deve essere un intero maggiore di zero.
2. “Ref\_IDBrano” deve essere diverso per ogni coppia OrdineBrano, Ref\_IDAlbum, in quanto un album non può contenere più volte lo stesso brano.

### 5.2.5. Diagramma entità-relazione MYSQL Workbench



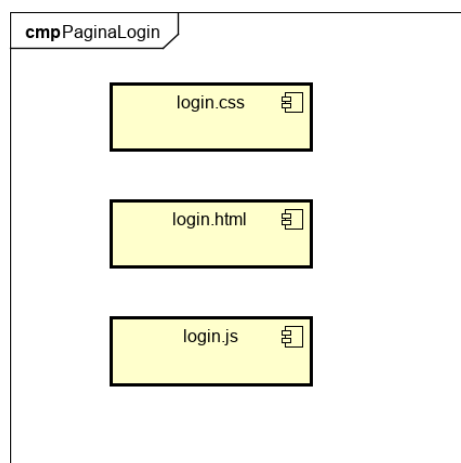
### 5.3. Decomposizione del sistema

Il sistema è stato decomposto del seguente modo:

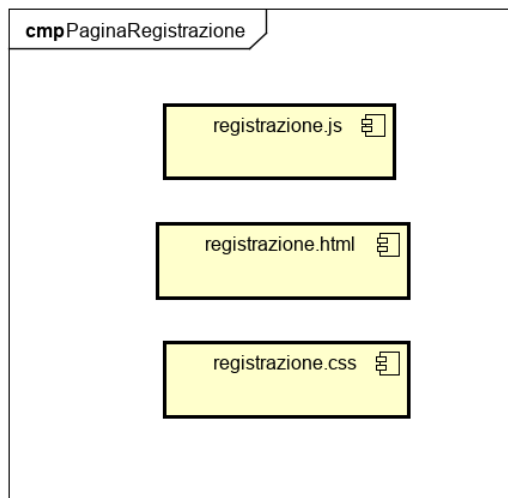


Il sottosistema “SoftwareLatoClient” comprende tutti i file statici del client (html, css e javascript) organizzati secondo il modello di Model-View-Controller. Nella reale organizzazione dei file e delle cartelle, tutti i file facenti parte di questo sottosistema si trovano all’interno della cartella “public”. Di seguito vi sono i diagrammi che esplorano il sottosistema “SoftwareLatoClient” nel dettaglio.

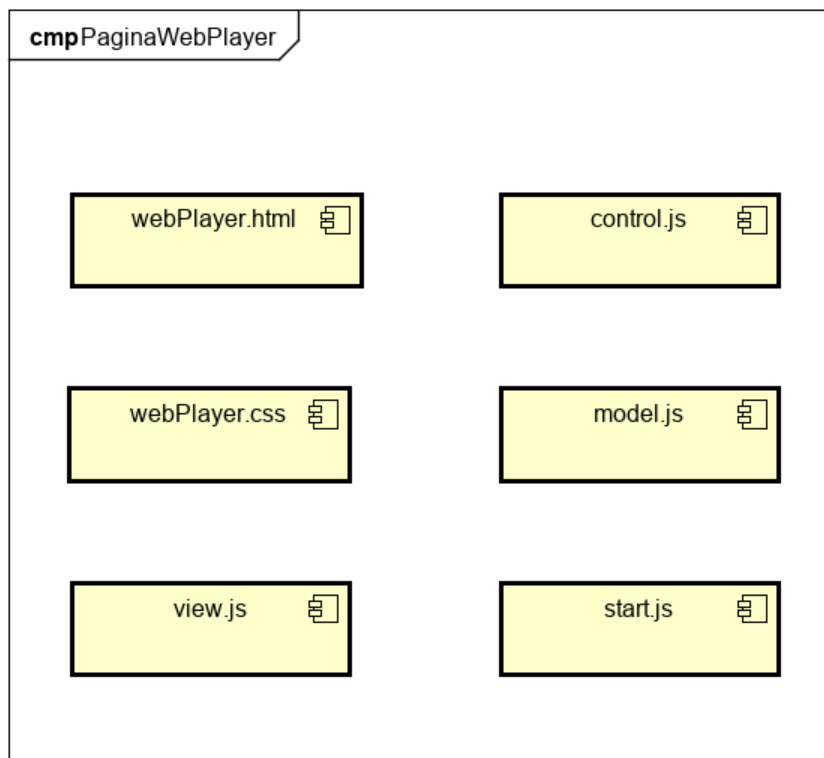
#### PaginaLogin



## PaginaRegistrazione



## PaginaWebPlayer



### 5.3.1. Suddivisione dei ruoli all'interno del team di sviluppo

Qui di seguito vengono riportati i lavori di gruppo e nello specifico i lavori di ogni singolo componente del gruppo durante lo sviluppo dell'applicazione web.

Lavori di gruppo:

- Analisi dei requisiti funzionali e non funzionali in base all'obiettivo del sistema.
- Realizzazione dei modelli del sistema: stesura dei casi d'uso e realizzazione dei diagrammi.
- Progettazione del software.
- Progettazione del database.
- Stesura e realizzazione della documentazione prodotta.

Coppola Gianmarco:

- Implementazione del lato server dell'applicazione e di tutti i suoi file.
- Implementazione e realizzazione del database.
- Implementazioni delle funzioni lato client che comunicano con il server.
- Implementazione della componente javascript della pagina di registrazione.

Moscarelli Giuseppe:

- Implementazione della pagina di login in tutte le sue componenti (html, css e javascript).
- Implementazione della pagina del web player (html e css) e delle componenti javascript model.js e view.js.
- Gestione della visualizzazione responsive e adattamento delle pagine in tutti i dispositivi.

Vara Andrea:

- Implementazione della pagina di registrazione nelle sue componenti html e css.
- Strutturazione della pagina del web player e realizzazione di tutti i suoi pannelli.
- Implementazione di tutte le funzionalità che concernono il player all'interno della pagina del web player e del file javascript control.js.



## 5.4. Diagramma di deployment

Il diagramma di deployment consiste in una mappatura hardware/software del sistema sulla base dell'architettura client-server scelta (come illustrato sopra).

Il nodo server contiene:

- un'istanza Node.JS Server che si occupa di fornire i contenuti in base alle richieste da parte dei nodi client e della gestione interna del server.
- un'istanza di MySQL Server che si occupa di interrogare il database richiedendo e modificando tutti i dati in base alle richieste ricevute da parte del web server di Node.JS. Tali richieste sono gestite dal protocollo TCP/IP.
- un archivio contenente i file audio che vengono inviati al client (i brani per i quali viene richiesto lo streaming).
- i file html, css e javascript lato client che vengono forniti a tutti i client che si collegano al sito web dell'applicazione.

I moduli utilizzati da Node.js lato server sono i seguenti:

- ~ express;
- ~ express-session, modulo che si occupa della gestione delle sessioni degli utenti che accedono al sistema;
- ~ body-parser, modulo che permette di interpretare il corpo di una risposta http;
- ~ debug, modulo che permette l'inizializzazione della porta di ascolto;
- ~ http, modulo necessario alla creazione del server http;
- ~ mysql, modulo che gestisce l'interazione con il database MySQL;
- ~ nodemail, modulo per l'invio delle mail;
- ~ crypto, modulo che permette la crittografia delle password;
- ~ generate-password, modulo che permette di generare una password casuale;
- ~ mediaserver, modulo che gestisce lo streaming della musica.

I nodi client consistono in tutti i dispositivi che si collegano tramite web all'applicazione e contengono una copia cache delle librerie utilizzate (come Bootstrap, JQuery, ecc). I nodi client possono accedere alle funzionalità anche tramite l'applicazione Android SoundWave, in tal caso contengono una copia locale statica del software lato client opportunamente modificato.

