

Relazione del Progetto di Programmazione Web e Mobile (PWM)

Dipartimento di Informatica Giovanni Degli Antoni

Samuele Manclossi
09882A

30 giugno 2023

*“Non quia difficilia sunt non audemos, sed quia non
audemos difficilia sunt”*

— Seneca

Abstract

Questo PDF è la relazione del progetto conclusivo del corso di Programmazione Web e Mobile, tenuto dal prof. Valerio Bellandi alla Università Statale di Milano nell’A.A. 2022-2023.

Esso consiste nella realizzazione di una applicazione web dal nome SOCIAL NETWORK FOR MUSIC, che si deve prefiggere lo scopo principale di implementare un sistema di gestione delle playlist musicali create da utenti ed (eventualmente) condivise.



Contents

1	Analisi delle specifiche	1
1.1	Funzionalità da implementare	1
1.1.1	Gestione degli utenti	1
1.1.2	Ricerca e visualizzazione dei dati	1
1.1.3	Preferiti e Playlist	1
1.1.4	Opzionale: creazione di gruppi	2
2	Tecnologie utilizzate e interazioni tra esse	3
3	Interfaccia utente e front-end	4
3.1	Pagine e relative funzioni e visibilità	4
3.1.1	Welcome page - <i>index.html</i>	4
3.1.2	Registrazione - <i>register.html</i>	4
3.1.3	Login - <i>login.html</i>	4
3.1.4	Ricerca - <i>search.html</i>	4
3.1.5	Playlist - <i>playlists.html</i>	4
3.1.6	MyPlaylists - <i>myplaylists.html</i>	4
3.2	Navbar condivisa tra tutte le pagine	4
3.2.1	Pulsante <i>SNM</i>	4
3.2.2	Pulsante <i>Search</i>	4
3.2.3	Pulsante <i>Groups</i>	4
3.2.4	Dropdown <i>User</i>	4
4	Appendice A: brani di codice	5
5	Appendice B: response status	5

1 Analisi delle specifiche

1.1 Funzionalità da implementare

SOCIAL NETWORK FOR MUSIC (di seguito **SNM**) deve essere in grado di svolgere diverse funzioni, tra cui le seguenti.

1.1.1 Gestione degli utenti

La gestione degli utenti è necessaria, e deve essere affrontata attraverso diverse fasi:

Registrazione La registrazione degli utenti avviene attraverso una pagina apposita, a cui può accedere qualsiasi utente. Questa pagina consente (dopo appositi controlli) di effettuare una richiesta POST al backend, che si occuperà di registrare i dati dopo averli nuovamente verificati.

I campi che saranno richiesti in fase di registrazione dovranno essere:

- **Email**, campo unico che verrà usato per autenticarsi.
- **Password**
- **Conferma password**
- **Nome utente**, che sarà il nome mostrato agli altri quando una playlist viene condivisa
- **Preferenze musicali** (genere preferito, da una lista restituita dal backend)
- **Gruppi musicali preferiti**

Login Mediante il login un utente entra nel proprio profilo, diventando in grado di vedere le proprie playlist e il proprio profilo.

Questo avverrà attraverso la richiesta di due campi:

- Email
- Password

Nonostante sia stato suggerito di salvarsi le informazioni in locale sarebbe più opportuno usare i token JWT. A questo sarà dedicata una apposita sezione.

Logout Questa funzione si spiega da sola, senza bisogno di tanti commenti.

Cambiare i propri dati Ogni dato deve essere modificabile tramite apposite richieste al backend.

Eliminare l'account Deve essere possibile eliminare l'account, cancellando tutte le informazioni che lo riguardano. Se l'account viene eliminato, vengono rimosse tutte le playlist create da quell'utente. Sarebbe pertanto consigliabile la realizzazione di un sistema che consenta, all'eliminazione dell'account, di stabilire a chi passa la proprietà di quelle playlist, oppure scegliere di eliminarle.

1.1.2 Ricerca e visualizzazione dei dati

Questo deve essere fatto attraverso due pagine apposite, una che si occupi della ricerca e ne mostri i risultati e una che mostri le informazioni sul singolo brano, permettendo l'inserimento di questo nelle playlist (eventualmente la creazione di una nuova playlist in caso non dovesse già esistere).

Vista l'ampia varietà di campi per cui è possibile cercare, una soluzione sarebbe prima mostrare alcuni risultati per ogni campo e poi restringere la ricerca su un campo specifico.

1.1.3 Preferiti e Playlist

Preferiti Un utente può decidere di selezionare un numero imprecisato di brani come suoi brani preferiti. Questo li aggiunge alle informazioni (private) del suo account.

Playlist Una playlist è una collezione di brani denotata da alcune informazioni:

- **Proprietario**
- **Nome**
- **Descrizione**
- **Canzoni contenute** (si ipotizza di porre gli oggetti che descrivono la canzone in un array)
- **Stato di condivisione** (public/private)
- **Gruppi consentiti** (Array di nomi (che godono della proprietà di essere **unique**), dei gruppi)

Una playlist privata può essere vista solamente dal creatore, una playlist pubblica può essere vista liberamente da chiunque mediante una apposita pagina.

Si noti che le playlist devono implementare le seguenti azioni:

- **Cancellazione**
- **Rendere privata**
- **Rendere pubblica**
- **Condividere con un gruppo**
- **Rimuovere condivisione con un gruppo**
- **Follow/Unfollow**
- **Trasferimento della proprietà ad un nuovo owner**
- **Aggiunta o Rimozione di canzoni**
- **Recuperare le informazioni**

1.1.4 Opzionale: creazione di gruppi

Si potranno creare delle comunità di utenti, di qui in avanti **gruppi**, a cui gli utenti potranno iscriversi / disiscriversi. Quando un utente è iscritto, esso è in grado di vedere tutte le playlist condivise con quella comunità specifica, e risulta anche in grado di parteciparvi condividendo altre playlist.

Il creatore del gruppo può essere in grado di escludere qualcuno dal gruppo (forse).

2 Tecnologie utilizzate e interazioni tra esse

3 Interfaccia utente e front-end

3.1 Pagine e relative funzioni e visibilità

3.1.1 Welcome page - *index.html*

La pagina di accoglienza sarà una pagina con pochissime funzionalità, destinata prevalentemente a fare da “vetrina” del servizio offerto.

Vi saranno due pulsanti: **login** e **register**, e la navbar per navigare invece la parte pubblica. Se si è già loggati, nel load si verrà reindirizzati alla pagina dove sono mostrate le playlist pubbliche o a quella dove è possibile effettuare ricerche.

3.1.2 Registrazione - *register.html*

La pagina di registrazione sarà costituita da un form. Se si è loggati, si verrà reindirizzati automaticamente alla pagina di login. Altrimenti, ci si potrà registrare. Se la registrazione va a buon fine si viene reindirizzati alla pagina di login, altrimenti viene mostrato il messaggio d'errore restituito dal backend.

3.1.3 Login - *login.html*

La pagina di login sarà costituita da un form. Se si è già loggati si viene reindirizzati alla pagina di visualizzazione dati pubblici, altrimenti ci si può loggare. In ogni caso saranno emessi appositi messaggi per segnalare lo stato.

3.1.4 Ricerca - *search.html*

La pagina di ricerca consentirà di effettuare una ricerca per vari campi. Il meccanismo è ancora da studiare nei dettagli.

3.1.5 Playlist - *playlists.html*

In questa pagina si potranno vedere le playlist pubbliche.

3.1.6 MyPlaylists - *myplaylists.html*

In questa pagina si potranno vedere le proprie playlist, divise tra private, pubbliche e quelle altrui che vengono seguite.

3.2 Navbar condivisa tra tutte le pagine

Tutte le pagine avranno accesso a una navbar condivisa, costituita dai seguenti elementi.

3.2.1 Pulsante *SNM*

Consente di tornare alla pagina delle playlist pubbliche.

3.2.2 Pulsante *Search*

Consente di tornare alla pagina di ricerca.

3.2.3 Pulsante *Groups*

Consente di tornare alla pagina dove sono mostrati i gruppi, ed eventualmente visualizzare le informazioni su di essi.

3.2.4 Dropdown *User*

Esso conterrà diverse operazioni sul profilo, tra cui le seguenti:

Nome	Cosa fa	Condizioni di visualizzazione
<i>login</i>	Redirige alla pagina di login	Non essere loggati
<i>register</i>	Redirige alla pagina di registrazione	Sempre visibile
<i>profile</i>	Redirige alla pagina del profilo	Essere loggati
<i>logout</i>	Effettua il logout e redirige alla “vetrina”	Essere loggati
<i>my playlists</i>	Redirige alla pagina privata delle playlist personali	Essere loggati

4 Appendice A: brani di codice

5 Appendice B: response status