

# PROGETTO OBJECT ORIENTATION

## Documentazione:

Traccia 1:

Sistema di gestione di una libreria musicale di un gruppo di utenti.

Dario Morace N86003778 Mattia Marucci N86003853

12 Gennaio



Pagina volutamente bianca.

# Indice

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Descrizione del progetto</b>                                 | <b>4</b>  |
| 1.1      | Descrizione della traccia e analisi del problema . . . . .      | 4         |
| <b>2</b> | <b>Diagramma delle classi del dominio del problema</b>          | <b>5</b>  |
| 2.1      | Diagramma del problema . . . . .                                | 5         |
| 2.2      | Classi contenute nel diagramma e spiegazione . . . . .          | 6         |
| <b>3</b> | <b>Diagramma delle classi del dominio della soluzione</b>       | <b>7</b>  |
| 3.1      | Diagrammi della soluzione . . . . .                             | 7         |
| <b>4</b> | <b>Sequence Diagram di funzionalità</b>                         | <b>11</b> |
| 4.1      | Sequence diagram riempimento libreria con spiegazione . . . . . | 11        |
| 4.2      | Sequence diagram ricerca tracce con spiegazione . . . . .       | 12        |
| <b>5</b> | <b>Esempi d'uso</b>   | <b>13</b> |
| 5.1      | Log-in . . . . .  | 13        |
| 5.2      | Registrazione . . . . .   | 14        |
| 5.3      | Pannello Home . . . . .   | 15        |
| 5.4      | Pannello Libreria . . . . .                                     | 16        |
| 5.5      | Pannello Ricerca . . . . .                                      | 17        |
| 5.6      | Pannello Info . . . . .   | 18        |
| <b>6</b> | <b>Conclusioni</b>  | <b>19</b> |
| 6.1      | Repository GitHub utilizzata . . . . .                          | 19        |

# 1 Descrizione del progetto

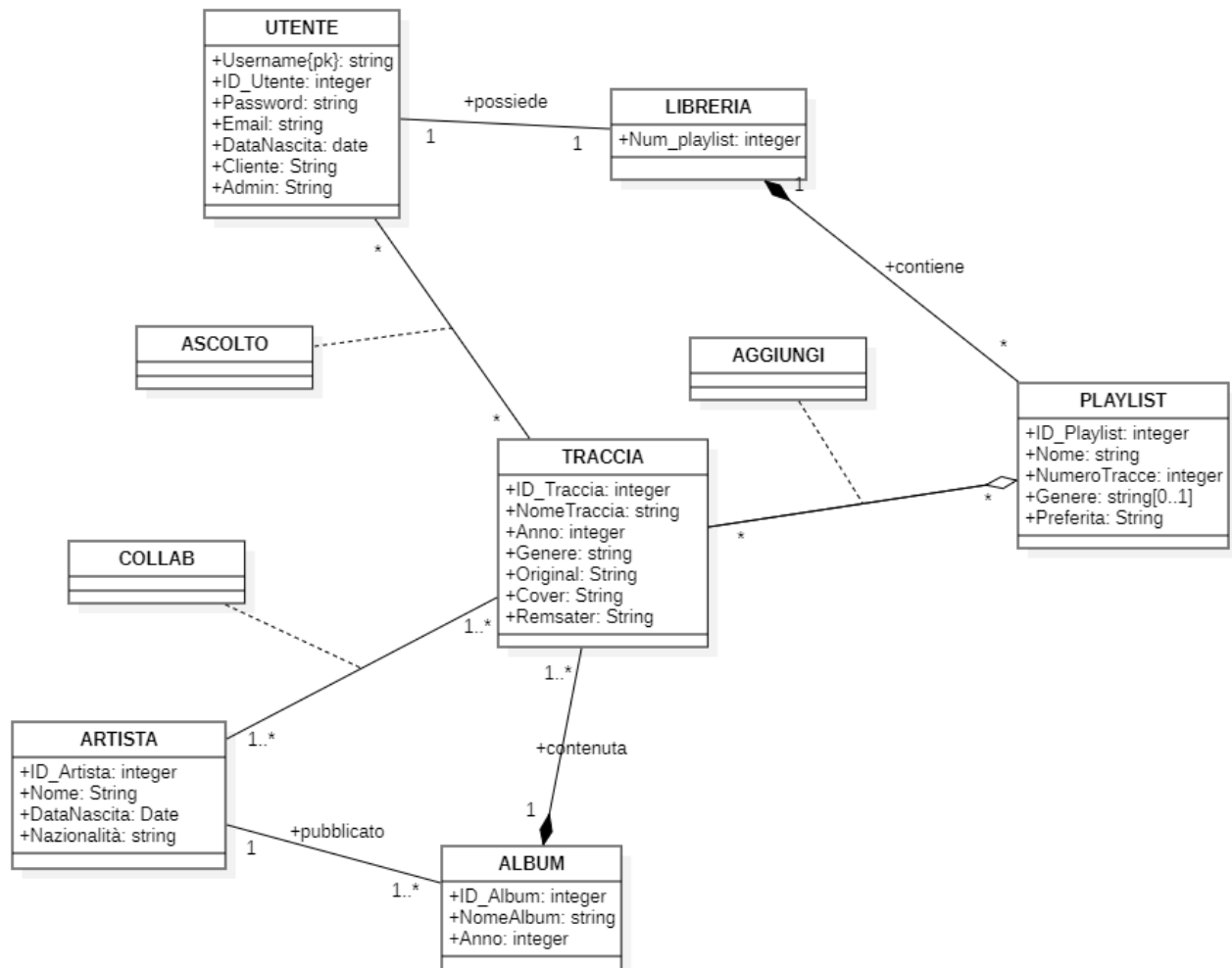
## 1.1 Descrizione della traccia e analisi del problema

Si è progettato ed implementato un applicativo in Java con GUI in JavaFX/Swing per la gestione di una libreria musicale di vari utenti dovendo tenere conto di differenziare il loro ruolo all'interno di quest'ultimo. Si provvederà quindi in primis: alla progettazione di diagrammi delle classi che definiscano il dominio del problema e della soluzione per poi passare all'effettiva creazione dell'applicativo con spiegazione di due funzionalità tramite sequence diagram. Ad affiancare questa applicazione vi sarà un database relazionale il quale permetterà effettivamente all'applicativo di funzionare correttamente restituendoci in fatti tramite query le tuple di nostro interesse. Il database è stato realizzato tramite l'applicativo PostgreSQL e interfaccia grafica pgAdmin4 e risponde a quelle che sono le necessità richieste dalla traccia. Nell'applicativo sono quindi stati implementati vari pannelli che permettono il display di: un pannello Home dove è possibile "esplorare" tutto quello che è presente nel DB tramite varie richieste, un pannello Libreria per utente che contiene varie playlist che possono essere gestite(create, eliminate, rese preferite) e a loro volta conterranno varie tracce anche loro possono essere aggiunte, rimosse o ascoltate, un pannello Search che permette la ricerca di tracce con vari parametri(da qui possiamo poi aggiungerle ad una playlist o ascoltarle) ed un pannello Info che permette solo agli admin di recuperare informazioni riguardo gli utenti, le fasce orarie in cui usano l'applicativo e precisamente i loro ascolti.

## 2 Diagramma delle classi del dominio del problema

Di seguito troviamo il diagramma delle classi del dominio del problema dove sono illustrati i modelli di riferimento su cui sono stati fatti i ragionamenti per cominciare lo sviluppo delle richieste proposte.

### 2.1 Diagramma del problema



## 2.2 Classi contenute nel diagramma e spiegazione

All'interno del diagramma delle classi del dominio del problema troviamo quindi i modelli pensati ed utilizzati per descrivere e risolvere il problema.

Di seguito sono quindi descritti:

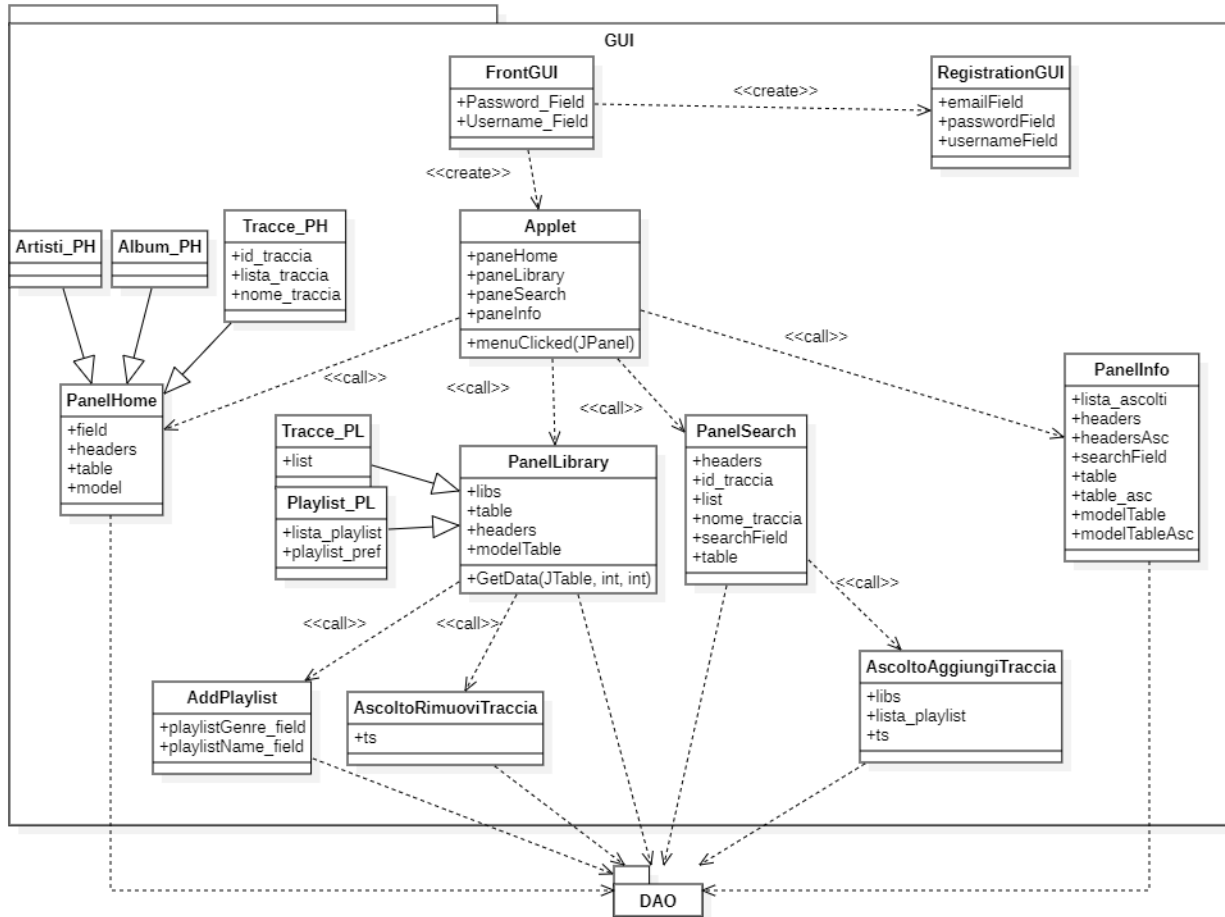
- **Utente:** Rappresenta l'utente che utilizza l'applicazione tramite parametri generici che ne descrivono la persona e una stringa speciale che invece ne descrive il tipo di utenza (Cliente o Admin).
- **Libreria:** Rappresenta la libreria che l'utente avrà a disposizione dove potrà creare, eliminare e gestire le proprie playlist con all'interno le tracce.
- **Playlist:** Rappresenta una playlist tramite vari parametri che ne permettono la corretta gestione come ad esempio il suo nome, il numero di tracce che contiene o il genere.
- **Aggiungi:** Rappresenta il metodo con cui andiamo a gestire le tracce aggiunte ad una determinata **Playlist**.
- **Traccia:** Rappresenta una traccia con tutti i suoi parametri che la descrivono come il suo nome, il genere o il suo id che ne permette poi l'**Ascolto**.
- **Ascolto:** Rappresenta il metodo con cui andiamo a tenere traccia degli ascolti di un determinato **Utente**.
- **Collab:** Rappresenta il metodo con cui andiamo a tenere traccia di un **Artista** quando collabora con uno o più artisti.
- **Artista:** Rappresenta un artista con tutti i suoi dati generici più un id univoco che ne permette di verificare l'unicità.
- **Album:** Rappresenta un album tramite i suoi attributi generici come il nome o l'anno ed in più possiede un id univoco per verificarne l'unicità.

### 3 Diagramma delle classi del dominio della soluzione

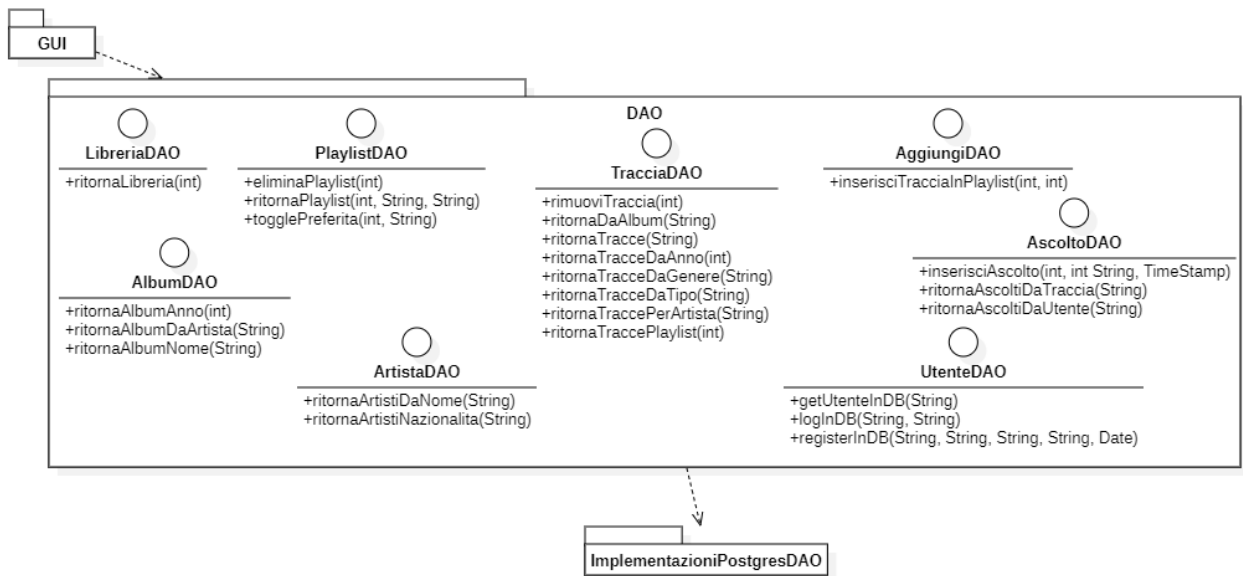
Di seguito troviamo i vari grafici sviluppati in base alla soluzione proposta e consegnata.

#### 3.1 Diagrammi della soluzione

1. **Modello di dettaglio package GUI:** All'interno di questo package troviamo tutte le classi di implementazione della GUI Swing di Java che ci permettono di avere su video quello che ci interessa.

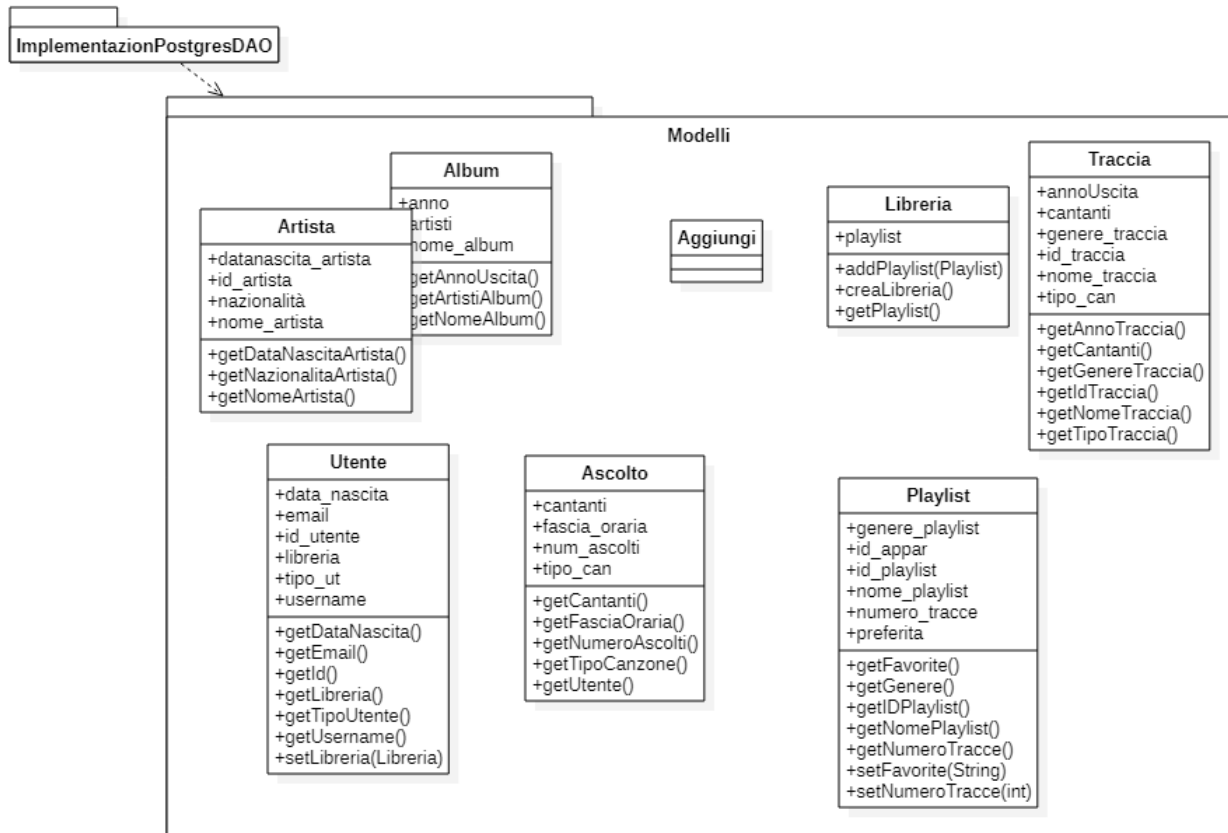


2. **Modello di dettaglio package DAO:** All'interno di questo package troviamo tutte le classi di tipo DAO ovvero Data-Access-Object di tipo interface che si occupano di isolare l'application layer dal persistence layer ovvero ad esempio un DataBase relazionale come quello usato in questa soluzione.

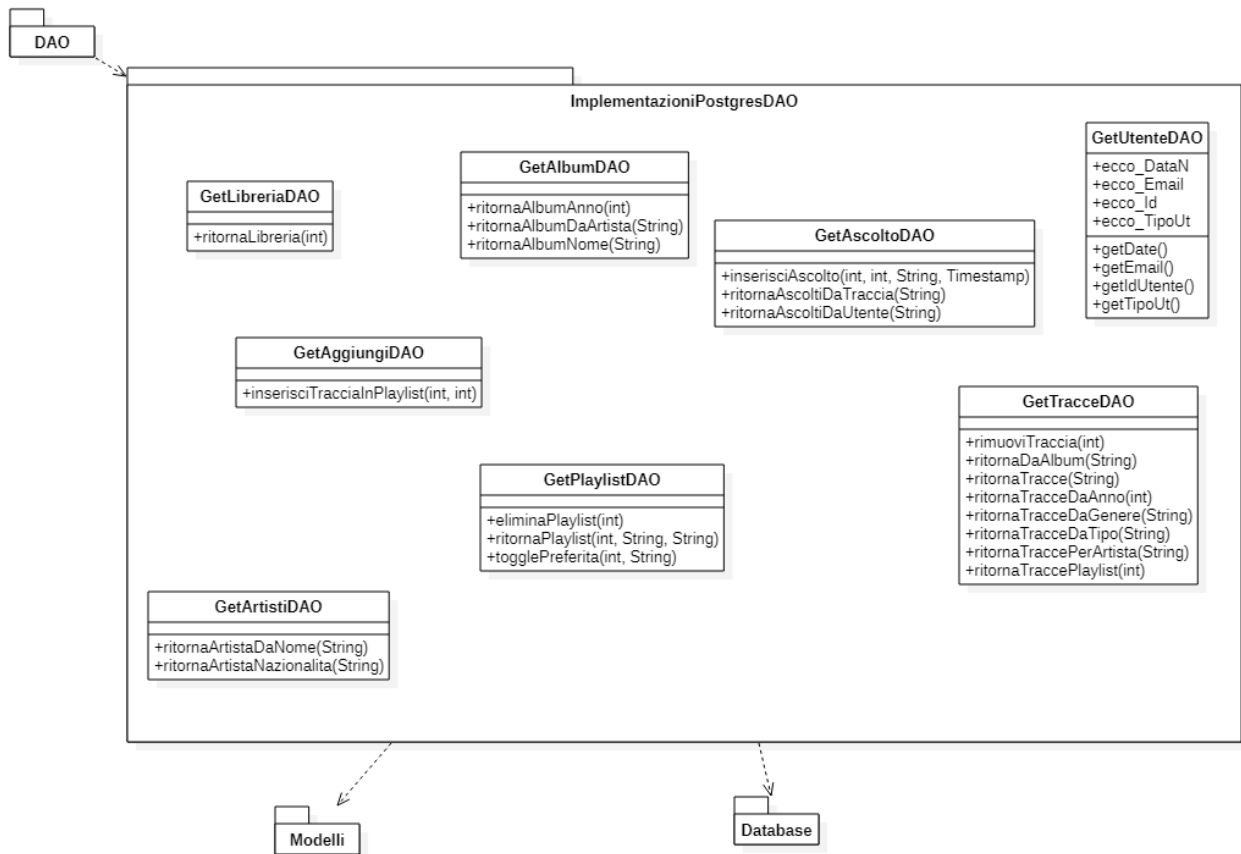




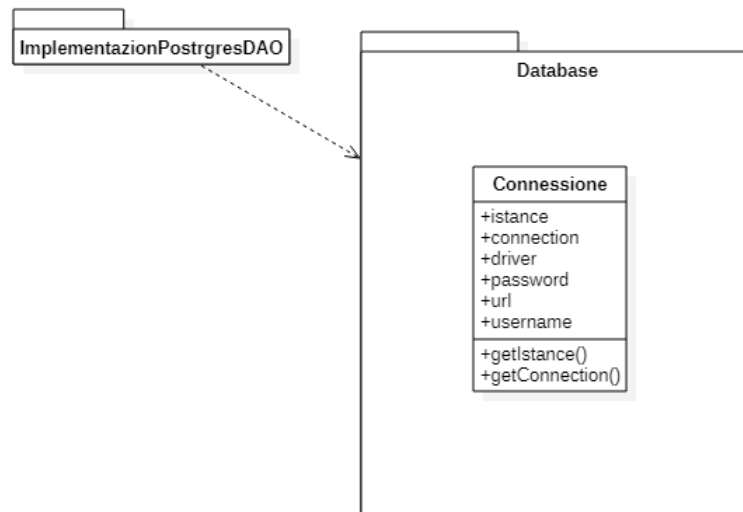
3. **Modello di dettaglio package Modelli:** All'interno di questo package troviamo i modelli prima citati e descritti con cui possiamo descrivere i componenti del nostro mini-world.



4. **Modello di dettaglio package ImplementazioniPostgresDAO:** All'interno di questo package troviamo i metodi utilizzati per effettuare le query con il DB ognuna implementa una specifica classe DAO.



5. **Modello di dettaglio package Connessione DB:** All'interno di questo package troviamo la classe che si occupa di aprire(o ri-aprire) la connessione col nostro DataBase.

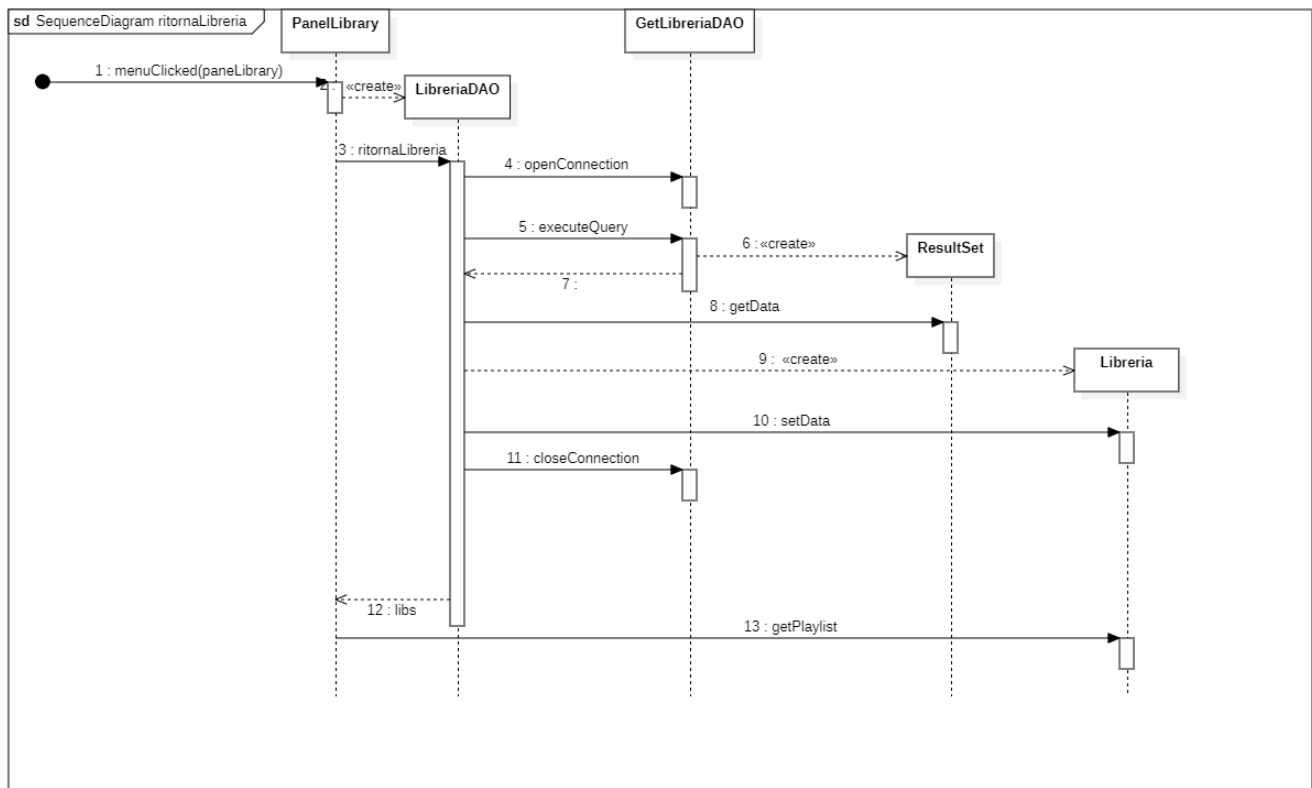


## 4 Sequence Diagram di funzionalità

Di seguito troviamo due Sequence Diagram di due funzionalità della soluzione proposta. I Sequence Diagram sono diagrammi UML che descrivono uno scenario in cui avere chiaro la sequenzialità di determinate azioni è importante. Le funzionalità scelte e rappresentate nei seguenti Sequence Diagram sono il riempimento della **Libreira** di un utente e la ricerca di una **Traccia** da parte di un utente.

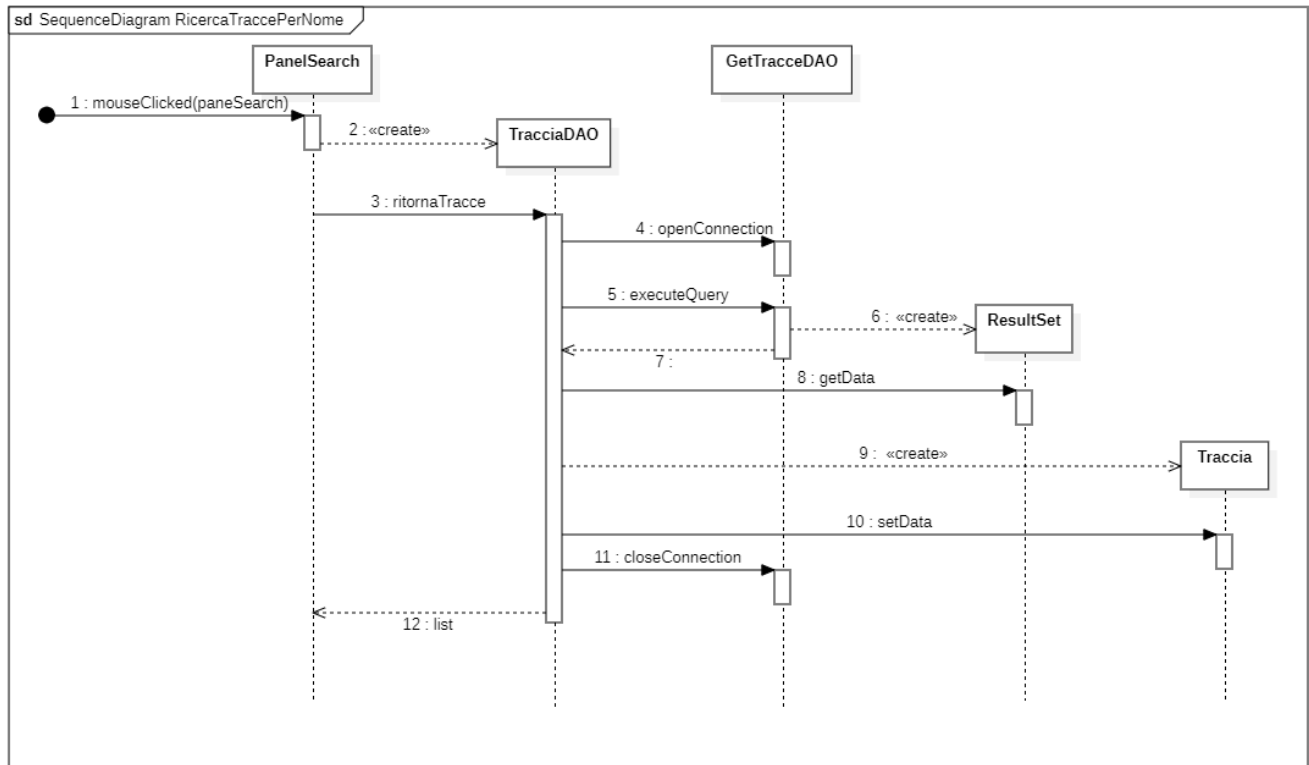
### 4.1 Sequence diagram riempimento libreria con spiegazione

Qui di seguito il Sequence Diagram che rappresenta il riempimento della libreria di un **Utente**.



## 4.2 Sequence diagram ricerca tracce con spiegazione

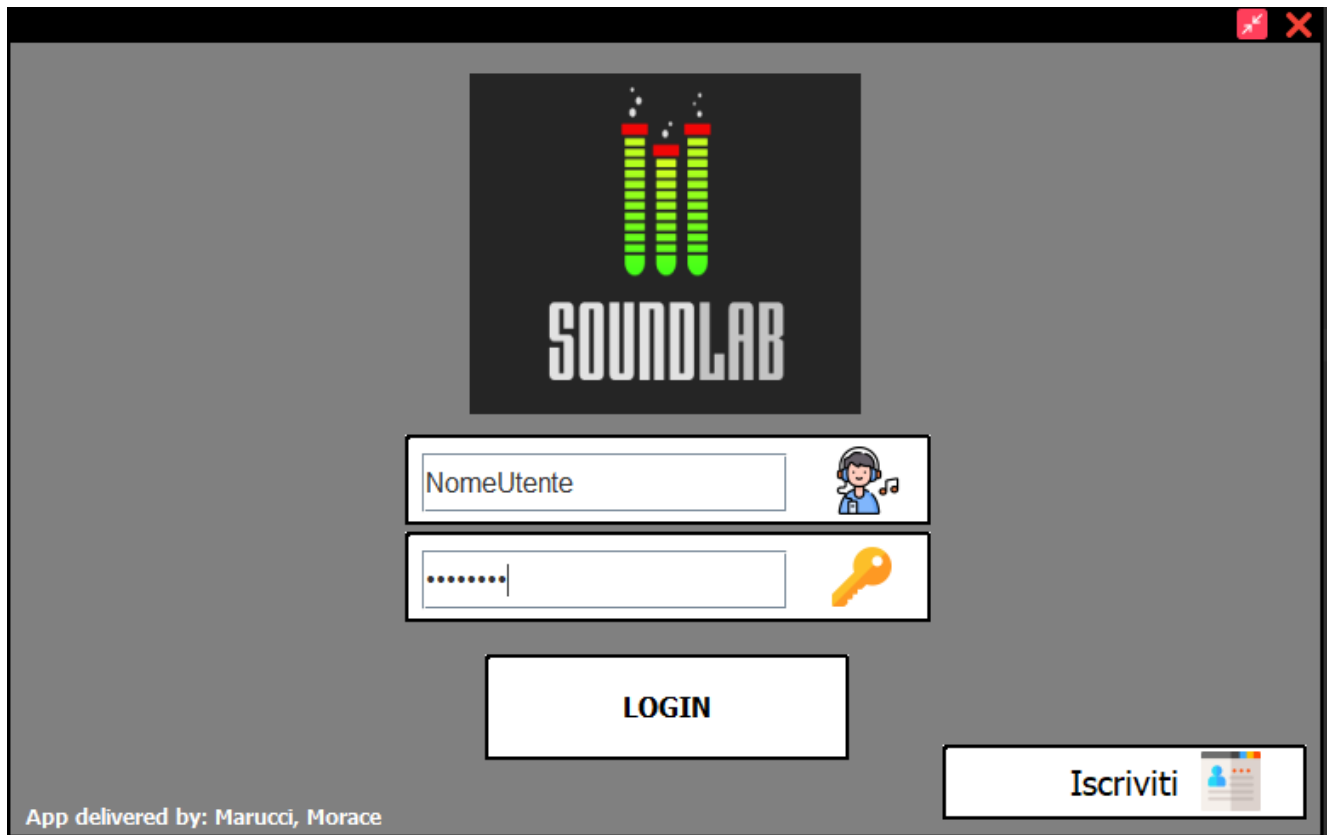
Qui di seguito il Sequence Diagram che rappresenta la ricerca di una traccia da parte di un **Utente**.



## 5 Esempi d'uso

### 5.1 Log-in

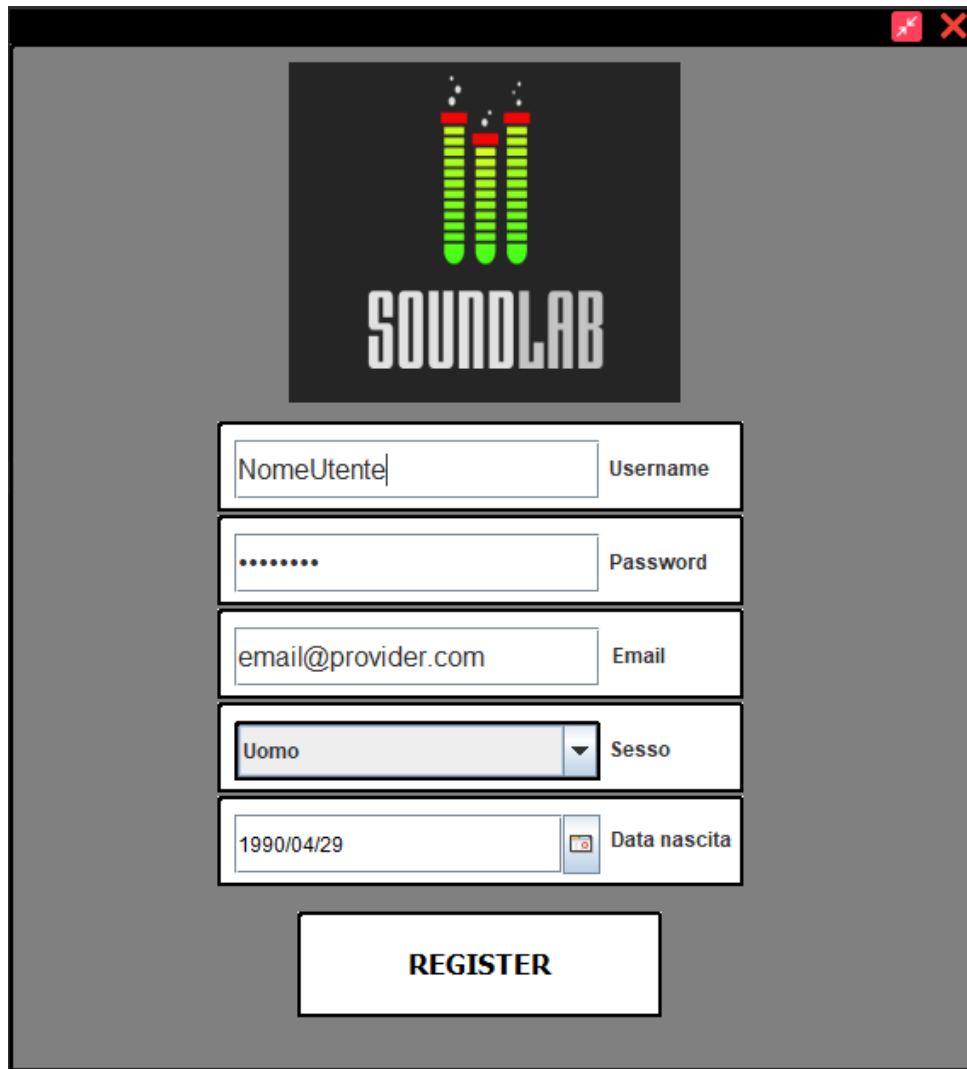
Qui di seguito la pagina che permette il Log-In.



The image shows a login interface for 'SOUNDLAB'. At the top center is a logo featuring three green test tubes with red caps and the word 'SOUNDLAB' in a stylized, blocky font. Below the logo are two input fields: the first is labeled 'NomeUtente' and is accompanied by a small icon of a person wearing headphones; the second field contains a series of dots, indicating a password, and is accompanied by a yellow key icon. Below these fields is a large white button with the text 'LOGIN'. In the bottom right corner, there is a button labeled 'Iscriviti' next to a small icon of a person with a list. At the bottom left, a small text label reads 'App delivered by: Marucci, Morace'.

## 5.2 Registrazione

Qui di seguito la pagina che permette la Registrazione.



The image shows a registration form for 'SOUNDLAB' within a window frame. At the top center is the 'SOUNDLAB' logo, which consists of three stylized test tubes with green liquid and red caps, emitting small white smoke-like particles. Below the logo, the form contains five input fields, each with a label to its right: 'Username' (containing 'NomeUtente'), 'Password' (containing seven dots), 'Email' (containing 'email@provider.com'), 'Sesso' (a dropdown menu with 'Uomo' selected), and 'Data nascita' (containing '1990/04/29' and a calendar icon). At the bottom center of the form is a large white button with the text 'REGISTER' in bold black capital letters. The window has a standard title bar with a red close button in the top right corner.

**SOUNDLAB**

NomeUtente Username

..... Password

email@provider.com Email

Uomo Sesso

1990/04/29 Data nascita

**REGISTER**

## 5.3 Pannello Home

Qui di seguito la pagina Home che permette di esplorare tutto il contenuto del nostro DB.

**SOUNDLAB**

Benvenuto: mattia

**HOME**

**LIBRARY**

**SEARCH**

**SIGN OUT**

**INFO**

App delivered by: Marucci, Morace

**Home**

Esplora categorie:

**Artisti**

Select  **VAI**

**Album**

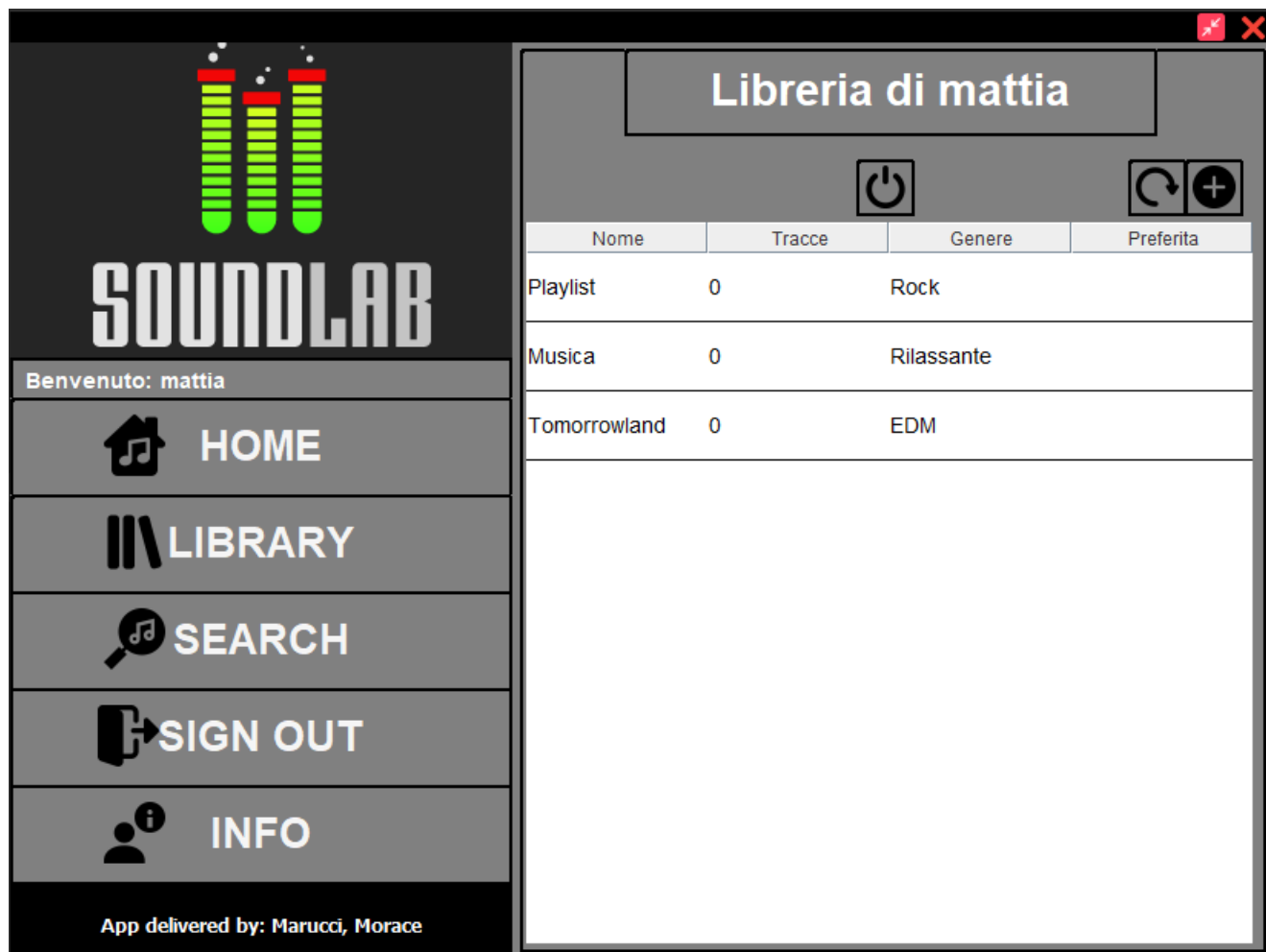
Select  **VAI**

**Tracce**

Select  **VAI**

## 5.4 Pannello Libreria

Qui di seguito la pagina Libreria che permette di esplorare, ascoltare e gestire le proprie playlist e canzoni.





## 5.5 Pannello Ricerca

Qui di seguito la pagina di Ricerca che permette di esplorare canzoni in base a diversi tipi di ricerca.

**SOUNDLAB**

Benvenuto: mattia

**HOME**

**LIBRARY**

**SEARCH**

**SIGN OUT**

**INFO**

App delivered by: Marucci, Morace

### Ricerca tracce

Traccia

Poker Face

Ricerca

| Nome       | Genere | Tipo Canzone | Anno | Artista   |
|------------|--------|--------------|------|-----------|
| Poker Face | Pop    | Original     | 2008 | Lady Gaga |

## 5.6 Pannello Info

Qui di seguito la pagina di Info che permette di esplorare ad un Admin gli ascolti di determinati utenti o tracce specifiche.

The screenshot displays the SoundLab app interface. On the left is a dark sidebar with the SoundLab logo (three green test tubes) and the text 'SOUNDLAB'. Below the logo, a grey bar says 'Benvenuto: Cringio'. The sidebar contains five menu items: 'HOME' with a house icon, 'LIBRARY' with a bookshelf icon, 'SEARCH' with a magnifying glass icon, 'SIGN OUT' with a door icon, and 'INFO' with a person and information icon. At the bottom of the sidebar, it says 'App delivered by: Marucci, Morace'.

The main content area is titled 'Info' and features a search form. The form has a dropdown menu labeled 'Utente' with a downward arrow, a text input field containing 'mattia', and a 'Ricerca' button. Below the form is a table with three columns: 'Nome', 'Fascia Oraria', and 'Numero Ascolti'.

| Nome   | Fascia Oraria | Numero Ascolti |
|--------|---------------|----------------|
| mattia | Pomeriggio    | 14             |
| mattia | Mattina       | 9              |

## 6 Conclusioni

L'applicazione sviluppata ha bisogno del suo DB per funzionare, quest'ultimo può essere trovato insieme al codice sorgente all'interno della repository GitHub in formato .sql o restore di Postgres. Inoltre all'interno della repository è possibile visionare tutti gli schemi UML presenti in questa documentazione e tutta la linea temporale dei commit effettuati.

### 6.1 Repository GitHub utilizzata

La repository utilizzata è stata la seguente: [LINK](#)