



# Progettazione e sviluppo di un software grafico per la gestione di una libreria musicale

Davide Maresca  
N86003819

Giuseppe Scotto Di Fasano  
N86003661

June 2022

# Indice

<b>1</b>	<b>DESCRIZIONE DEL PROGETTO</b>	<b>3</b>
1.1	Descrizione e analisi del progetto . . . . .	3
1.2	Requisiti identificati . . . . .	3
<b>2</b>	<b>PROGETTAZIONE</b>	<b>4</b>
2.1	Introduzione . . . . .	4
2.2	Dominio del problema . . . . .	4
2.3	Dominio della soluzione . . . . .	5
2.4	Sequence diagram . . . . .	6
2.4.1	Ricerca Traccia . . . . .	6
2.4.2	Modifica Credenziali . . . . .	7
<b>3</b>	<b>MANUALE D'USO</b>	<b>9</b>
3.1	Accedere ai file . . . . .	9

# DESCRIZIONE DEL PROGETTO

## 1.1 Descrizione e analisi del progetto

In questo progetto si sviluppa un software grafico per la gestione di una libreria musicale per un gruppo di utenti. Il software conterrà i dati di: utenti, artisti, album e tracce. Ogni utente avrà accesso ad una propria lista di brani preferiti (tipo playlist) che può gestire aggiungendo o rimuovendo delle tracce. Inoltre sarà presente un admin che potrà, nella sua sezione dedicata, visualizzare un insieme di utenti che hanno effettuato il maggior numero di ascolti divisi per fasce orarie (da 1 a 6, 4 ore per ciascuna fascia), l'admin dovrà selezionare semplicemente la traccia e cliccare un bottone per avere il risultato. Verrà indicata anche la versione per ogni traccia che si seleziona: originale, remastering (creata dall'artista della traccia originale) o cover (creata da un artista diverso dalla traccia originale). Per ogni artista è possibile visualizzare le tracce prodotte, gli album incisi e i suoi dati. Per ogni album è possibile visualizzare tutte le tracce contenute e i dati, come tempo totale e anno produzione. Ogni utente ha la possibilità di visualizzare gli altri utenti registrati e i loro brani preferiti. Sempre l'utente può modificare le sue credenziali: nickname, password, nome e cognome, nella sezione di impostazioni.

## 1.2 Requisiti identificati

All'interno della traccia proposta dal professore abbiamo individuato le seguenti classi con relative associazioni:

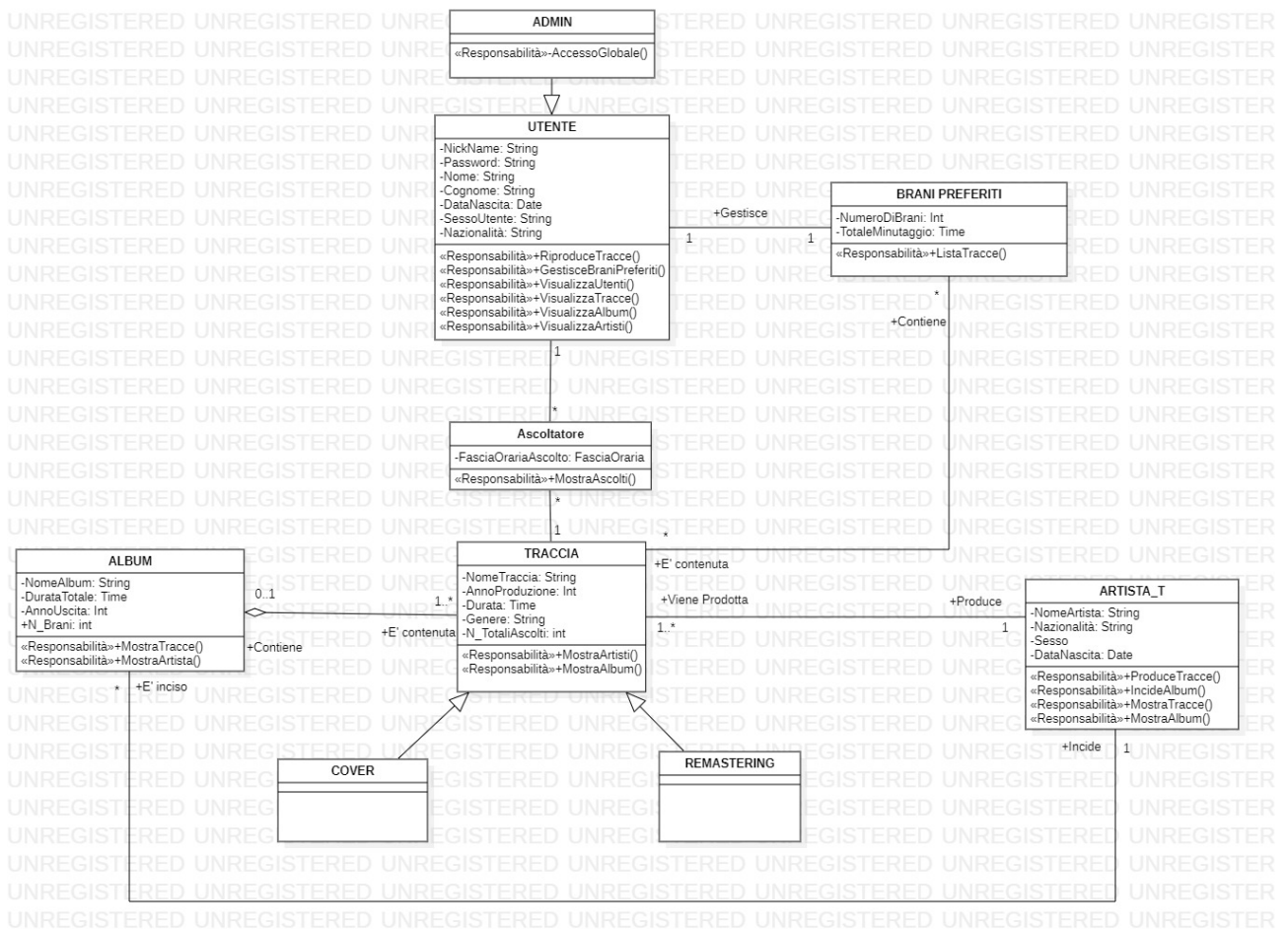
- **Utente**: indica una classe, la classe admin è ereditata dall'utente. Accede alla libreria musicale per ascoltare canzoni. L'utente è in relazione con Brani Preferiti (playlist), ha accesso alla propria playlist dove potrà salvare delle nuove tracce oppure eliminarle. Inoltre è in relazione con ascoltatore, dove viene registrato ogni singolo ascolto, includendo anche la fascia oraria di ascolto.
- **Tracce**: indica una classe. La traccia può essere di tre tipi: originale, remastering e cover. La traccia, oltre che con la classe ascoltatore, ha anche diverse associazioni con altre classi: 1) Con album, la traccia può far parte di un album dedicato (la traccia può essere pubblicata senza album). 2) Con artista, le tracce vengono prodotte da un unico artista senza featuring (due o più artisti per singola traccia). 3) Con Brani Preferiti, le tracce possono essere salvate all'interno di una playlist personale.
- **Artista**: indica una classe, produce le tracce. L'artista oltre ad essere associato con la classe Traccia, è associato con la classe Album per conoscere chi ha inciso quel determinato album, l'album può essere inciso da un unico artista.
- **Album**: indica una classe, contiene una lista di tracce, l'album può contenere una lista di tracce prodotte dallo stesso artista che ha inciso l'album.
- **Ascoltatore**: indica una classe. Ascoltatore tiene traccia di tutti gli ascolti effettuati dall'utente con relativa fascia oraria.

# PROGETTAZIONE

## 2.1 Introduzione

In questo capitolo introcuciamo la progettazione del software grafico, nel quale verranno identificati i concetti e i requisiti più importanti. Per farlo usiamo la rappresentazione in UML, che ci permette di dare una vista generale del problema con classi, attributi, metodi, associazioni e molteplicità delle associazioni.

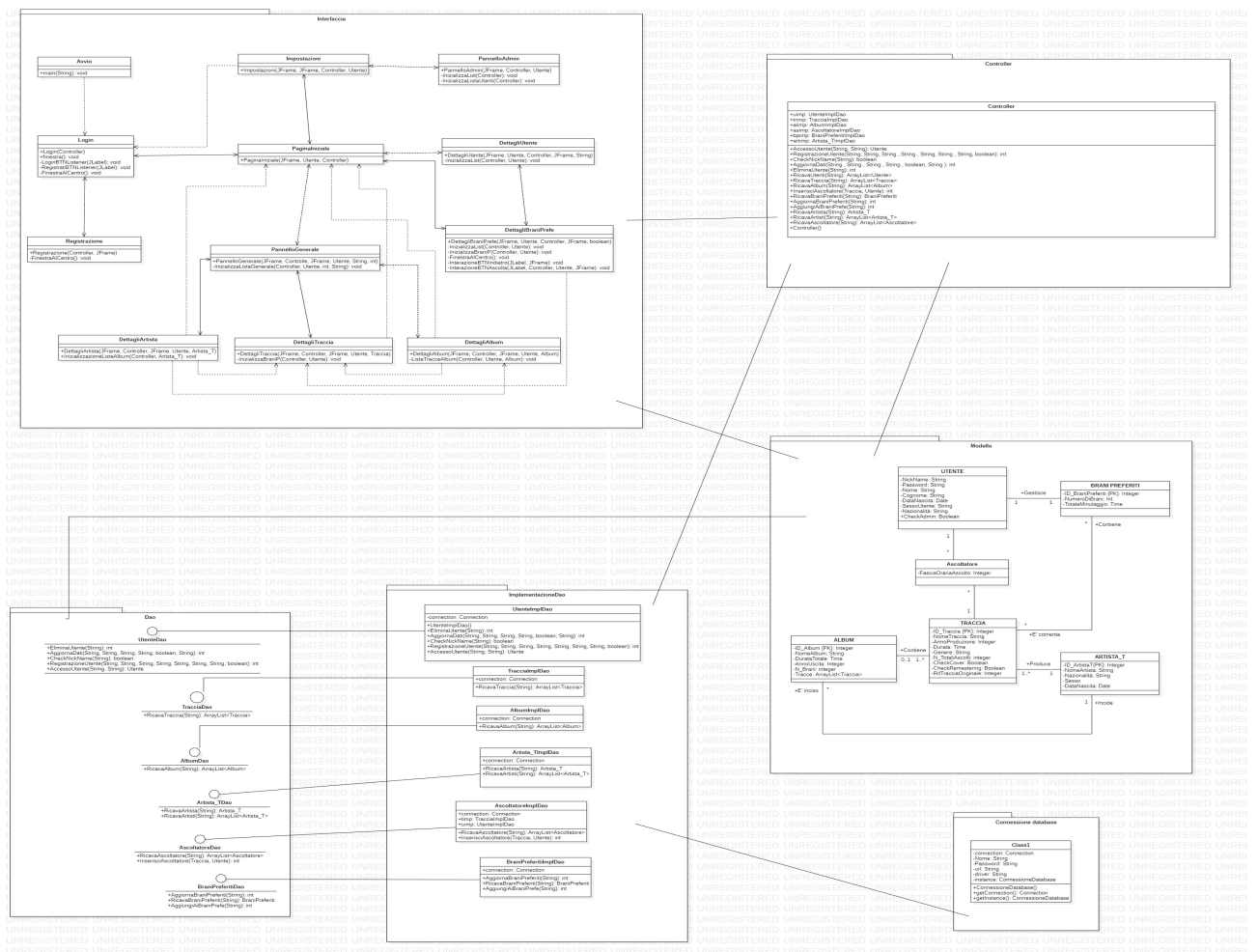
## 2.2 Dominio del problema



## 2.3 Dominio della soluzione

Come da immagine identifichiamo 6 package: interfaccia, controller, modello, implementazione, dao, dao e connessioneDatabase

- Il package interfaccia raggruppa tutte le interfacce create, i metodi che le compongono e le interazioni tra di esse.
- Il package controller contiene un'unica classe controller nella quale vengono richiamati i metodi contenuti nelle implementazioni di ogni classe, implementazioneDao.
- Il package ImplementazioneDao contiene tutte le classi che corrispondono al package Dao, in ogni classe viene definito il funzionamento di ogni metodo, questi metodi avranno il compito di inviare le istruzioni al database.
- Il package dao contiene una serie di prototipi per ogni classe, come scritto in precedenza questi prototipi poi verranno definiti nel package ImplementazioneDao.
- Il package connessioneDatabase definisce la connessione con il database.
- Il package modello rappresenta il modello UML delle classi individuate.



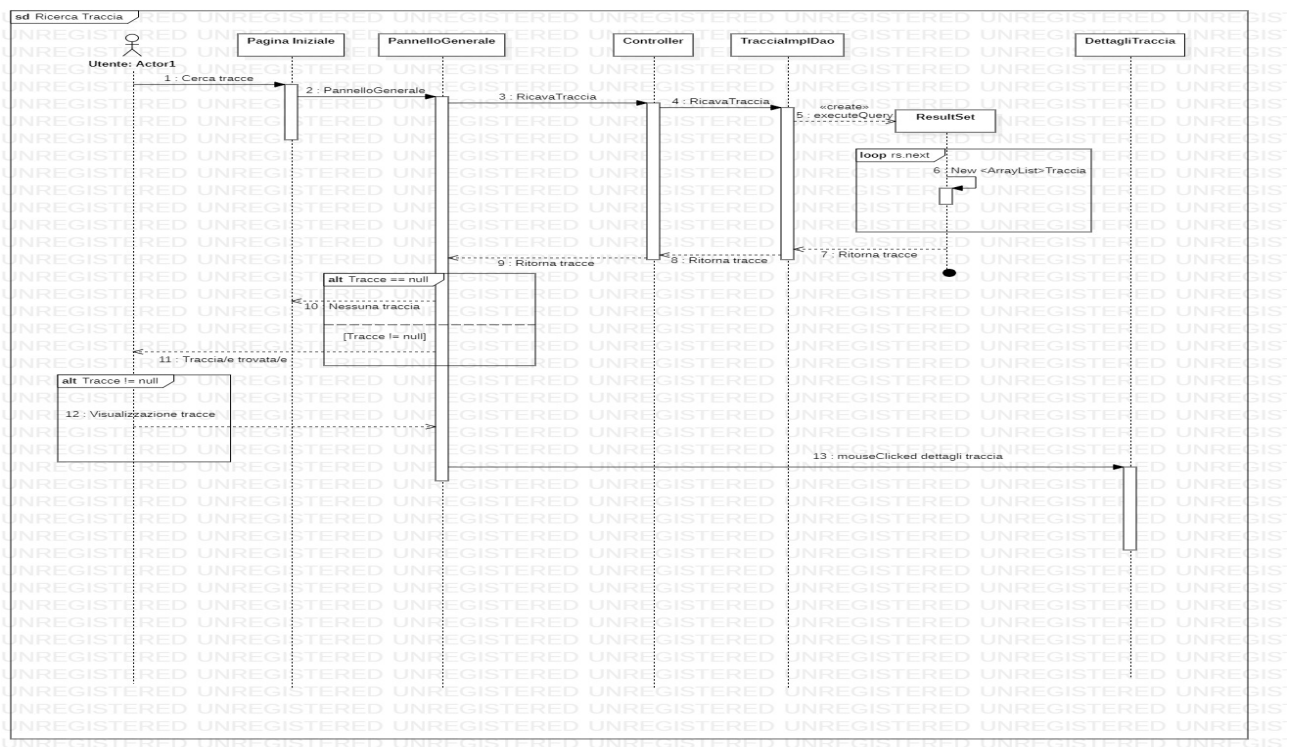
## 2.4 Sequence diagram

Un Sequence Diagram è un diagramma previsto dall'UML utilizzato per descrivere uno scenario. Uno scenario è una determinata sequenza di azioni in cui tutte le scelte sono state già effettuate.

### 2.4.1 Ricerca Traccia

L'utente nella fase iniziale si trova nella pagina iniziale, in questa può digitare nella barra di ricerca e cercare una traccia oppure una lista di tracce, quando l'utente preme il pulsante invio si apre un nuovo frame chiamato "PannelloGenerale", la GUI chiamerà il metodo "ricavaTraccia" dal controller che a sua volta chiamerà RicavaTraccia in TracciaImplDao. In questa fase verrà eseguita la query dove instanzierà un ArrayList<Traccia> e restituirà alla GUI l'ArrayList<Traccia> (in base alla ricerca dell'utente l'ArrayList potrà contenere 0 istanze, 1 istanza o più istanze).

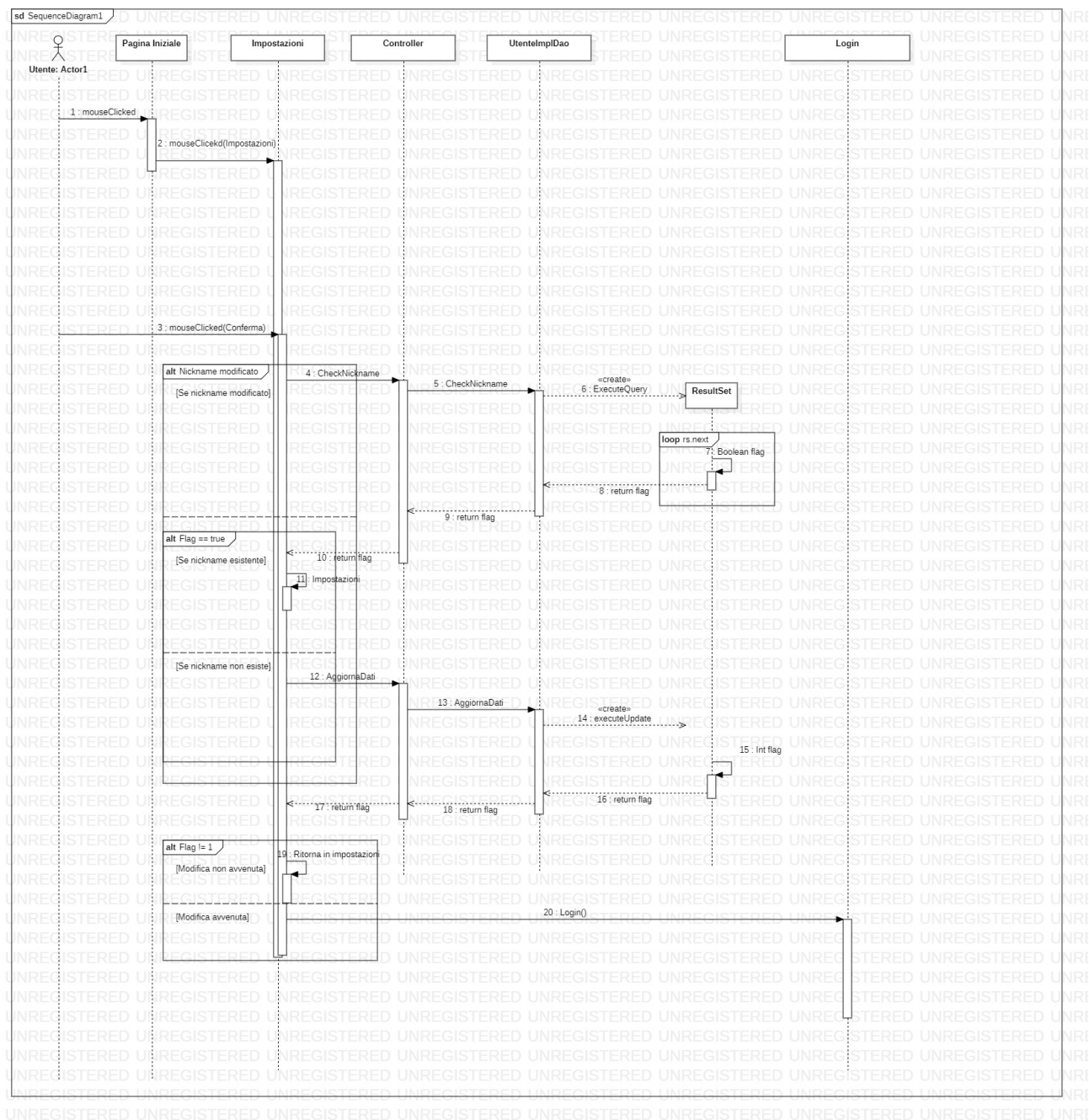
- Se l'ArrayList<Traccia> è null significa che ci sono 0 istanze, quindi non avendo risultati l'utente è invitato a tornare nella pagina iniziale.
- Se l'ArrayList<Tracce> contiene almeno un'istanza l'utente dovrà selezionare (nella lista che uscirà nel pannello generale) una traccia e cliccando un bottone "Dettagli-Traccia" potrà visionare i dettagli di quella traccia.



## 2.4.2 Modifica Credenziali

L'utente nella fase iniziale si trova nella pagina iniziale, in questa può cliccare il pulsante impostazioni, qui la GUI aprirà il frame "Impostazioni" dove sono presenti tutti i dati dell'utente. Cliccando il pulsante "Modifica Credenziali" i campi contenenti: nickname, nome, cognome e password, diventano modificabili. Si può modificare quel che si vuole e cliccare il pulsante conferma. Qui l'interfaccia eseguirà un primo controllo vedendo se il campo nickname è stato modificato. Nel caso in cui venisse modificato l'interfaccia chiamerà il metodo `CheckNickName` dal controller che a sua volta chiamerà il metodo `CheckNickName` in `UtenteImplDao`. In questa fase verrà eseguita una query che verificherà se il nickname scritto è già presente o meno nel database (come valore di ritorno abbiamo un booleano).

- Se il valore di ritorno dovesse essere vero vuol dire che il NickName scritto è già presente nel database e quindi viene stampato un messaggio di errore che avverte l'utente e l'interfaccia richiama nuovamente il frame Impostazioni, mostrando così all'utente lo stato di partenza.
- Se il valore di ritorno dovesse essere falso vuol dire che il nickname può essere cambiato e quindi l'interfaccia è pronta a chiamare il metodo `AggiornaDati` in controller che chiamerà a sua volta il metodo `AggiornaDati` in `UtenteImplDao`, da qui verrà eseguita una query `UpdateQuery` che aggiorna i dati restituendo come valore un intero.
  - Nel caso in cui il valore di ritorno dovesse essere diverso da 1 allora significa che la modifica non è avvenuta e l'interfaccia richiama nuovamente il frame Impostazioni, mostrando così all'utente lo stato di partenza.
  - Nel caso in cui il valore di ritorno dovesse essere uguale a 1 l'utente viene inviato al frame login e il frame impostazioni viene chiuso





# MANUALE D'USO

## 3.1 Accedere ai file

Col seguente link è possibile accedere a tutti i file relativi al progetto di Object Orientation: <https://github.com/Vainest/Progetto00.git>. Ovviamente il programma è appoggiato ad una base di dati, il codice della base di dati è già nel link precedente in README.md. Si consiglia di estrarlo dalla cartella compressa e usare il programma PostgreSQL per avviarlo, per l'accesso al Database usare le seguenti credenziali:

- Nome/LocalHost = localhost
- Nome utente = UtenteAdmin
- Password = admin

Nel login del programma per accedere come admin si consiglia l'uso della seguente combinazione di nickname e password:

- Nickname: amministratore2
- Password: admin

Nel login del programma per accedere come utente si consiglia l'uso della seguente combinazione di nickname e password:

- Nickname: Lordong
- Password: Odino

E' comunque possibile creare nuovi utenti nella pagina di registrazione.