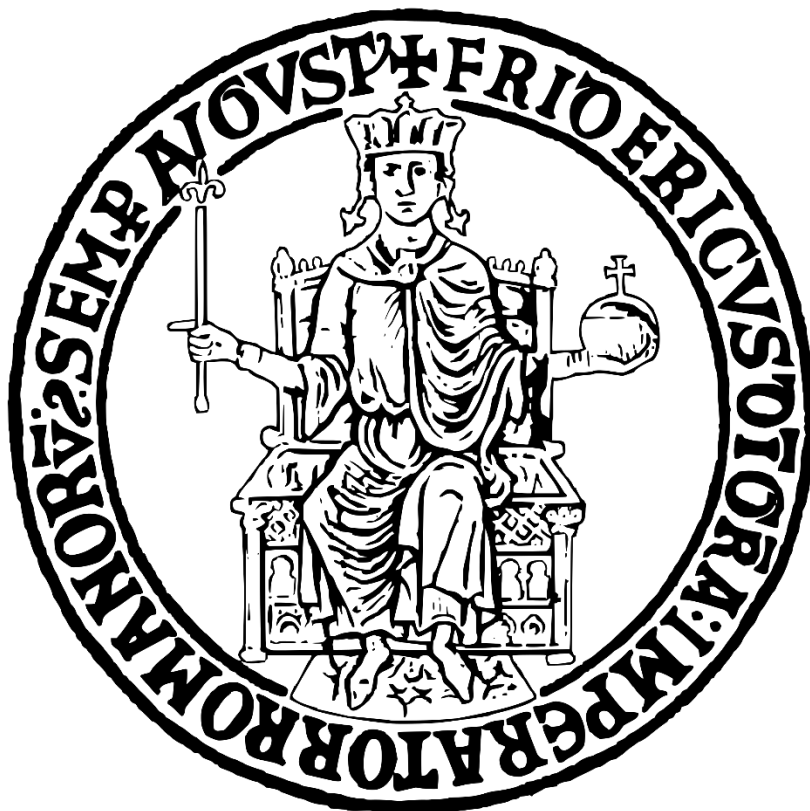


UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



BISCIOR SOUND

SISTEMA DI GESTIONE LIBRERIA MUSICALE DI UN GRUPPO DI UTENTI

TRACCIA 1

Vincenzo Meloni : N86003897

Loris Zannini : N86003744

Anno Accademico 2021/22

Docente:

Porfirio Tramontana

Capitolo 1

Introduzione

1.1 Descrizione del progetto

BisciorSound è un sistema informativo, composto da una base di dati relazionale e da un applicativo Java dotato di GUI , per la gestione delle librerie musicali di un gruppo di utenti. Il sistema permette di gestire le singole tracce degli utenti, gli album di appartenenza ,gli ascolti delle singole tracce, i remastering , le cover , una lista di preferiti delle tracce e delle cover che l'utente ha deciso di salvare. L'utente può accedere nella home, dove andrà ad interagire con la libreria musicale , effettuando il log-in o registrandosi (qualora non fosse dotato di account) inserendo il proprio username e la password. L'Admin, a differenza dell'utente, è già registrato nel sistema e vi accederà tramite un'interfaccia a parte che lo reindirizzerà nella home Admin , permettendogli di effettuare le medesime operazioni di un comune utente con l'aggiunta di una sezione riservata esclusivamente a chi vi è autorizzato, le Opzioni Admin , dentro il quale si potranno visionare tutti gli utenti registrati nel database , tra cui anche gli Admin. Avrà la possibilità di visionare , tramite una ricerca testuale , gli utenti che hanno effettuato più ascolti data una determinata traccia o cover. Inoltre è possibile andare ad individuare una fascia oraria dove un preciso utente ha effettuato più ascolti differenziando tracce e cover.

1.2 Requisiti identificati

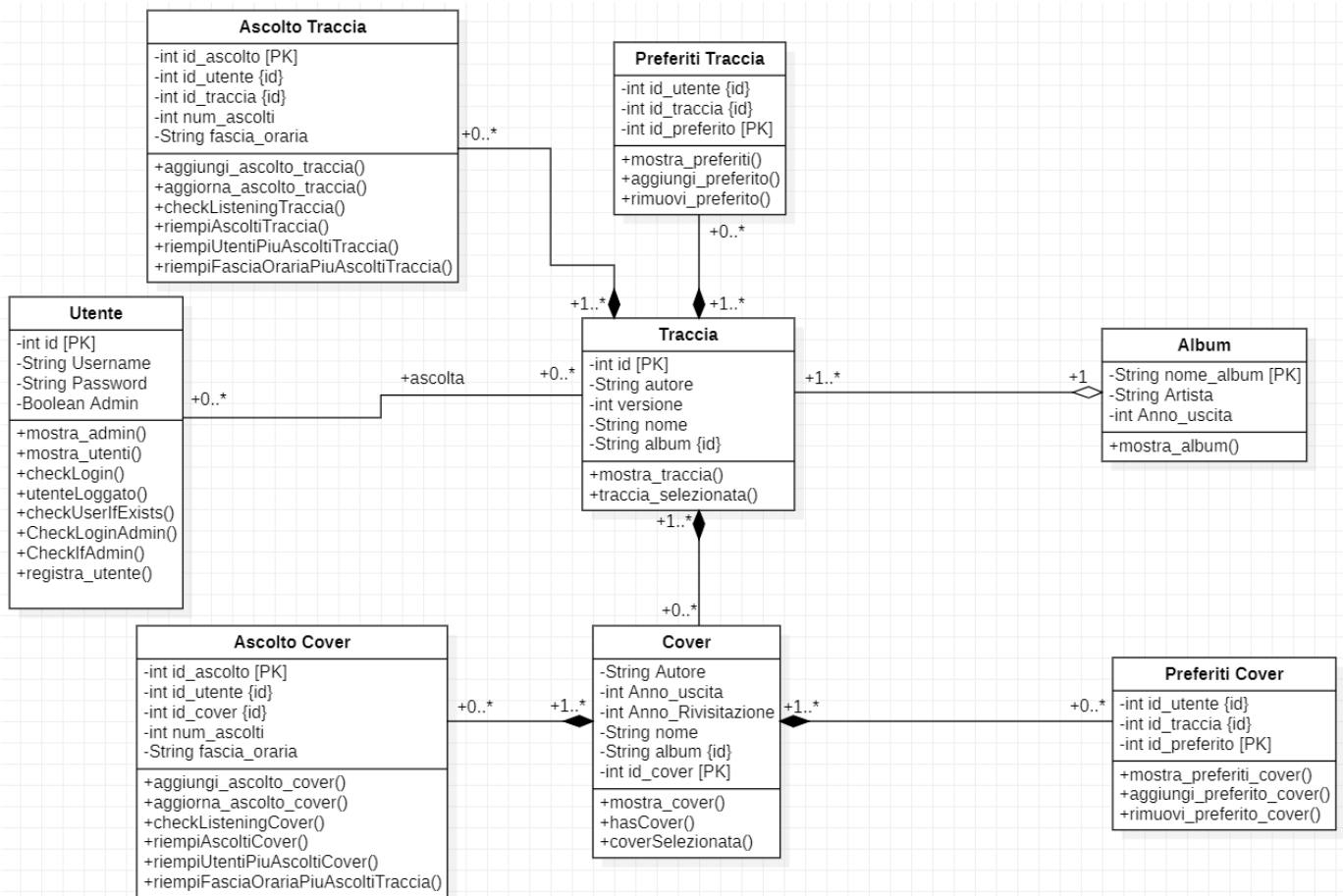
In primo luogo abbiamo suddiviso il progetto in pacchetti , ovvero: "ConfigurazioneDB, Controller, DAO , GUI, ImplementazioniPG_DAO, Model". Successivamente nella configurazione abbiamo creato una classe "ConnessioneDB". All'interno abbiamo scritto il codice per effettuare la connessione con il database. Il primo requisito individuato è stato creare una classe nel package "Model" con il nome di "Utente" , all'interno di esso abbiamo dichiarato le variabili in maniera coerente con la tabella "Utente" situata all'interno del database. Una volta creata la classe Utente , ne abbiamo creata una medesima interfaccia nel package "DAO" con il nome di "UtenteDAO". Inserendo i prototipi delle funzioni implementate nel package "ImplementazioniPG_DAO" con il nome di "UtenteImplementazionePG_DAO()" . Creando nei model tutte le classi in maniera coerente con le tabelle create nel database , inserendole nei "DAO" come interfacce e implementandole una per una. Tutte le implementazioni e i DAO vengono invocate nel Controller in modo tale da avere un unico package che gestisce l'intero funzionamento dell'applicativo. L'Utente una volta effettuato il log-in (o la registrazione) ,mediante un username univoco , ha la possibilità di visionare la lista delle tracce presenti , gli album , le cover e gli ascolti . Per ogni traccia l'utente può scegliere se riprodurla o aggiungerla ai preferiti. Le tracce hanno un proprio album di appartenenza visionabile nella sezione "Album". Nella sezione Preferiti , l'utente può osservare le tracce salvate e in caso rimuoverle. Per la sezione "Tracce" e "Preferiti" vi è

una sottosezione “cover” dove sono elencate tutte le cover. Nella sezione “Ascolti” l’utente può osservare le tracce che ha riprodotto con il numero di ascolti affiancato per ogni traccia e/o cover ascoltata. L’admin è in grado di visionare tutto ciò che vede un comune utente , con la particolarità che, data una certa traccia o cover, può andare ad osservare gli utenti che hanno effettuato più ascolti. Inoltre potrà effettuare una ricerca immettendo un nome utente presente nel sistema , andando a differenziare per tracce e cover la fascia oraria in cui il medesimo ha effettuato più ascolti. L’admin ha a portata di mano una lista completa di Utenti e di Admin registrati nel sistema.

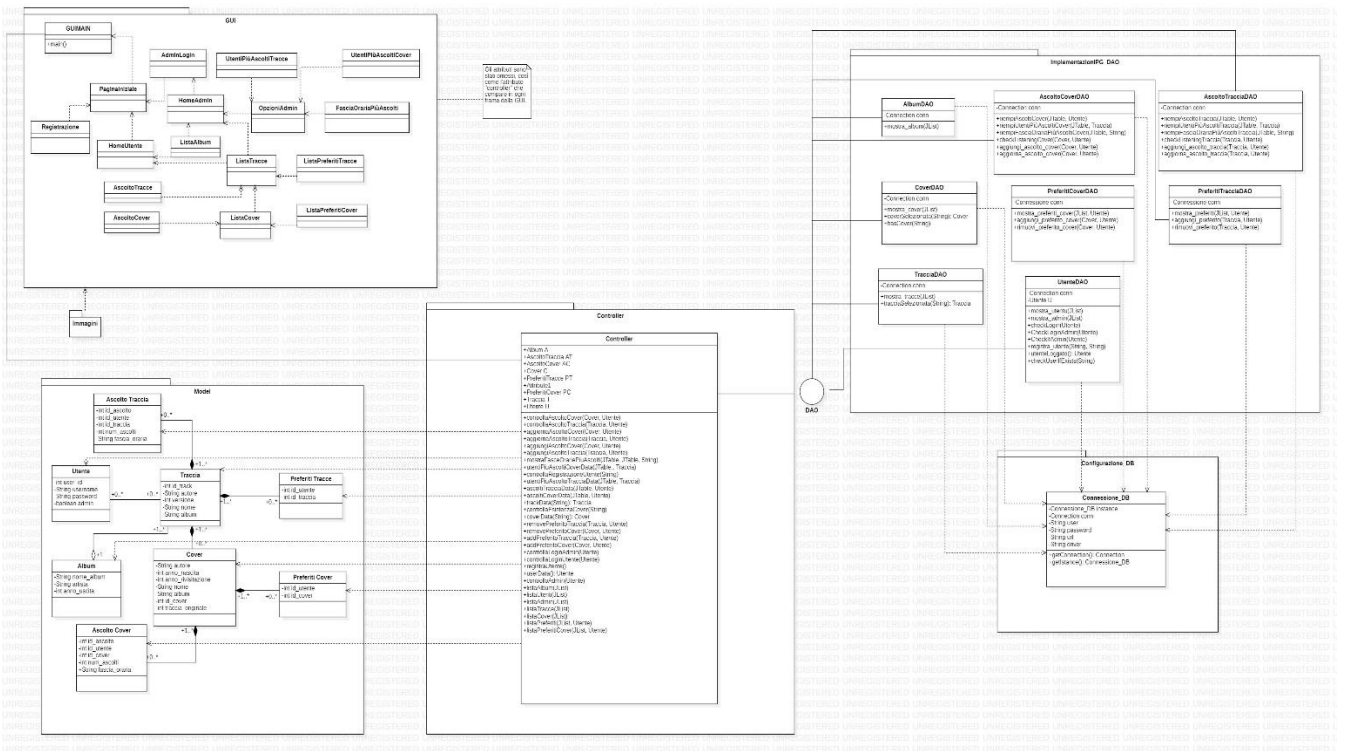
Capitolo 2

Progettazione Concettuale

2.1 Class Diagram

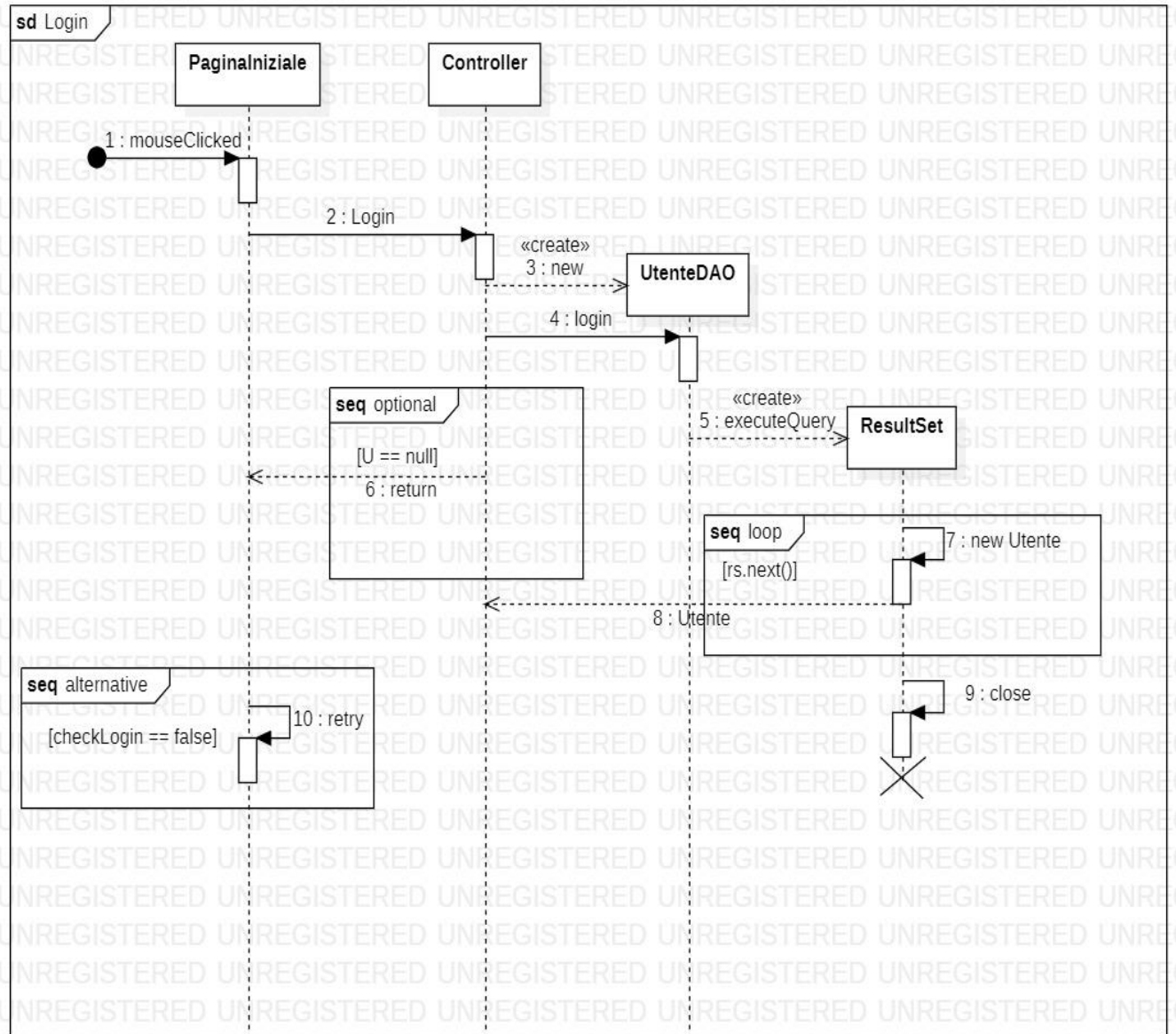


2.2 Diagramma Di Dettaglio



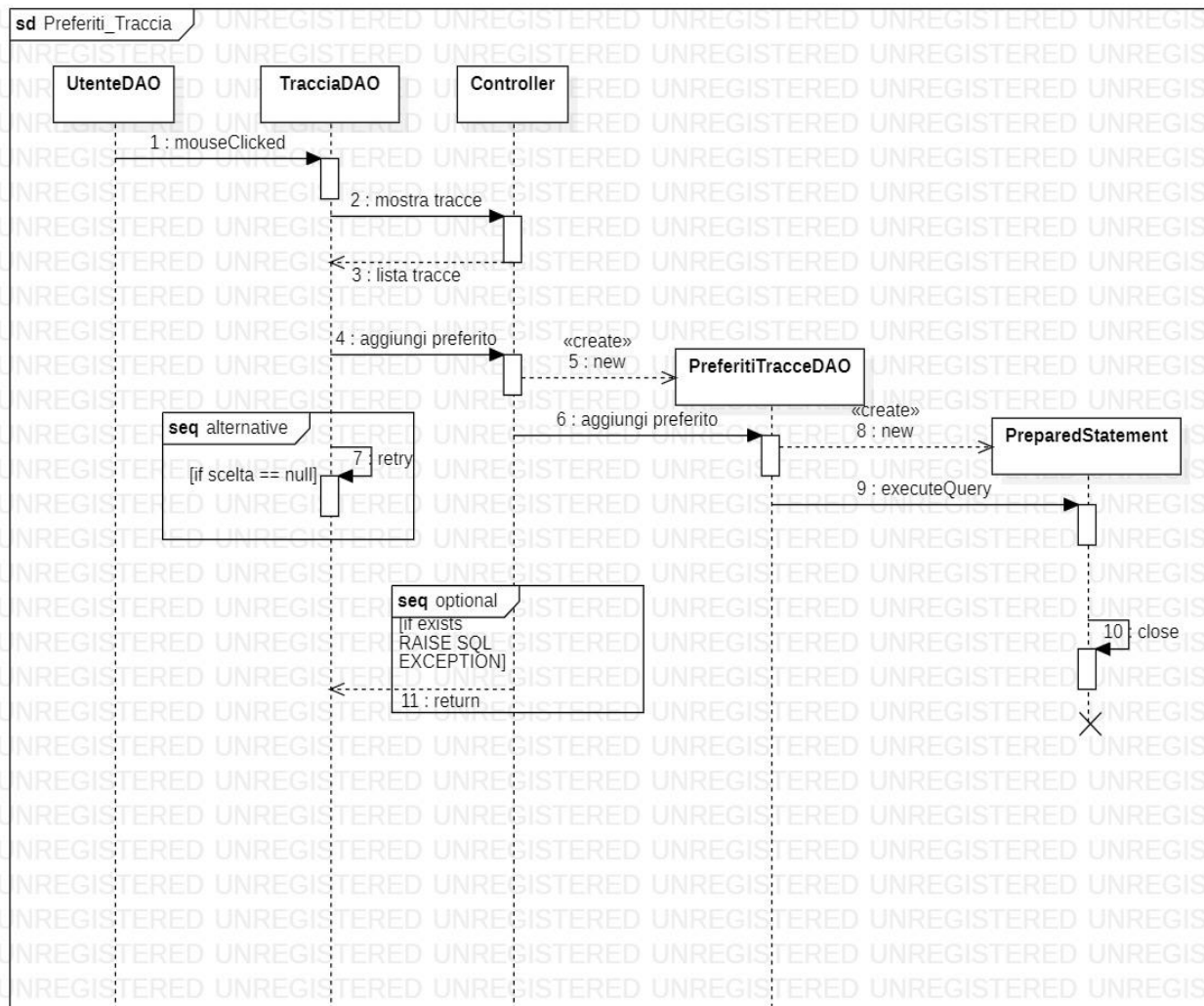
2.3 Sequence Diagram

2.3.1 Log-In



Quando viene cliccato il bottone di log-in, viene creato un Utente tramite la funzione `userData()` gestita dal controller, che con l'ausilio di una funzione booleana verifica se le credenziali sono corrette andando ad eseguire una query finché il `resultSet` risulta non nullo, altrimenti si rimane sulla schermata iniziale avvisando l'utente che le credenziali sono errate tramite uno `JOptionPane`.

2.3.2 Preferiti Traccia



Nel momento in cui un utente , scegliendo una traccia, vuole aggiungerla ai preferiti, viene eseguita la funzione `aggiungi_preferito()`, gestita dal controller, che permette di inserire la traccia selezionata nei Preferiti tramite il PreparedStatement , che esegue la query di INSERT ed effettuando una verifica tramite un trigger , accennato nella sequenza opzionale , inserisce la traccia nei preferiti se quest'ultima non è già presente, altrimenti viene visualizzata un'eccezione.