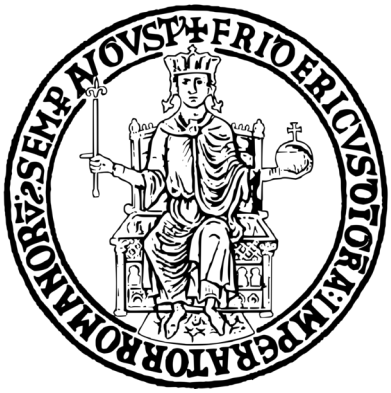
UNIVERSITA' DEGLI STUDI DI NAPOLI "FEDERICO II"
Scuola Politecnica e delle scienze di base
Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione



Progettazione e sviluppo di un applicativo Java per la gestione di una Galleria Fotografica Geolocalizzata

Michela Pollio
N86003697

Indice

Capitolo 1

Descrizione del progetto

1.1 Traccia

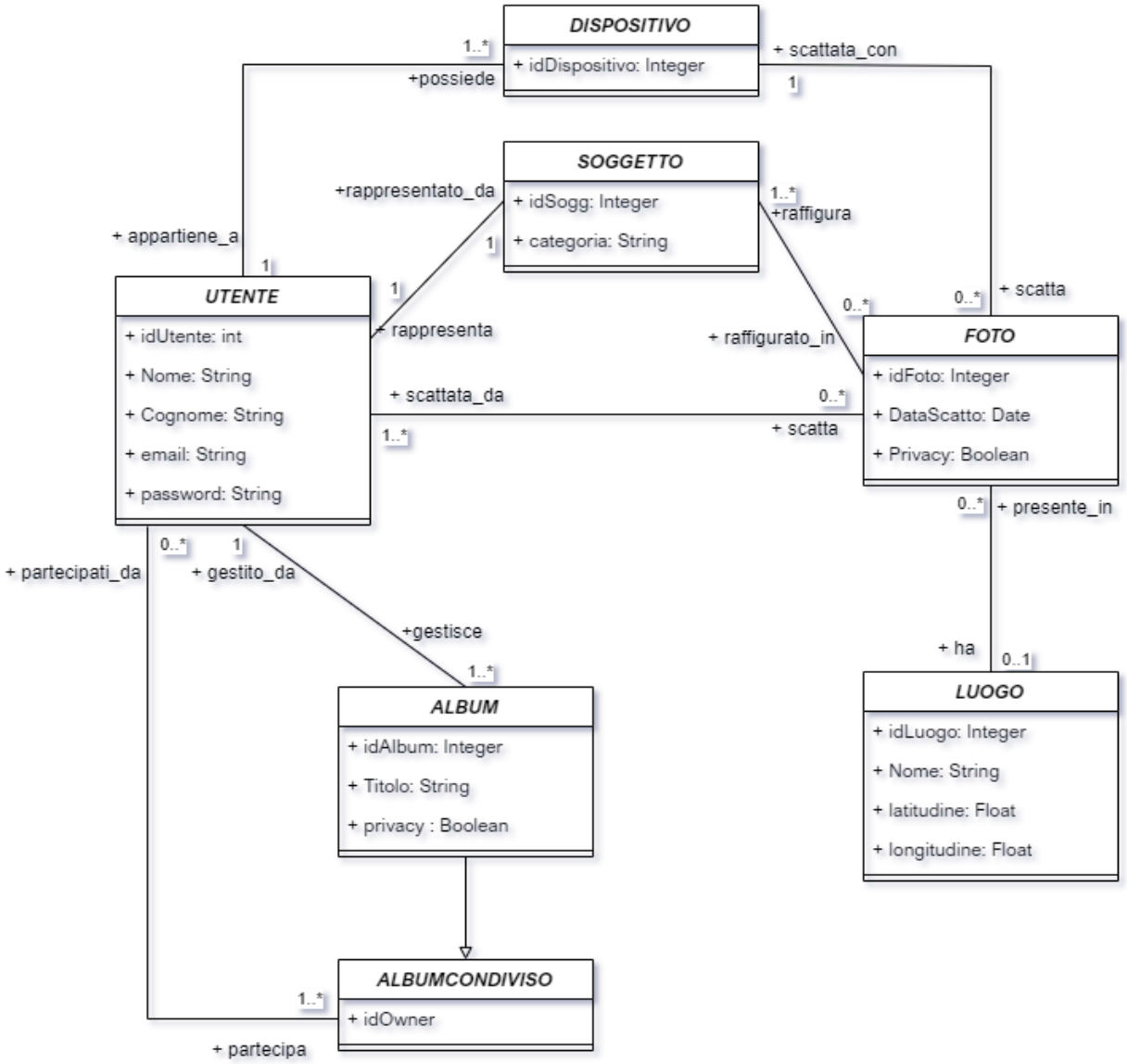
Questo progetto consiste nello sviluppo di un applicativo in linguaggio Java dotati di GUI Swing che consenta di gestire una Galleria Forografica Geolocalizzata. Ogni **utente** che accede al sistema ha la possibilità di poter accedere alla propria **galleria personale**, composta esclusivamente dalle foto scattate da lui. L'utente potrà inoltre accedere a **gallerie condivise** con altri utenti del sistema. Le **foto** presenti nella base di dati sono identificate univocamente e descritte tenendo conto della **data** dello scatto, del **dispositivo** con cui è stata scattata, dei **soggetti** o utenti presenti all'interno della foto e, se presente, del **luogo** in cui è stata scattata. Inoltre sarà possibile effettuare le seguenti operazioni

- Recupero di tutte le foto scattate nello stesso luogo
- Recupero delle foto che condividono lo stesso soggetto
- Classifica dei top 3 luoghi immortalati

Capitolo 2

Progettazione Concettuale

2.1 Class Diagram UML



2.2 Dizionario delle classi

Classe	Attributi
Utente	idUtente (int): identificativo per l'utente Nome(String): nome dell'utente Cognome(String): cognome dell'utente Email(String): email per l'accesso Password(String): password per l'accesso
Foto	idFoto(int): identificativo univoco della foto datascatto(data): data del giorno in cui la foto è stata scattata privacy(boolean): descrizione della visibilità della foto, cioè"pubblica" o"privata
Luogo	idLuogo(int): identificativo univoco di un luogo nome(String): nome identificativo del luogo latitudine(float): latitudine delle coordinate del luogo longitudine(float): longitudine delle coordinate del luogo
Album	idAlbum(int): identificativo univoco di un album nome(String): titolo dell'album provacy(boolean): descrizione della visibilità dell'album, cioè "pubblico" o "privato"
Album Condiviso	specializzazione di Album
Soggetto	idSoggetto(int): identificativo univoco del soggetto Categoria(String): descrizione del soggetto
Dispositivo	idDispositivo(int): identificativo univoco del dispositivo tipo(String): descrizione della tipologia di dispositivo

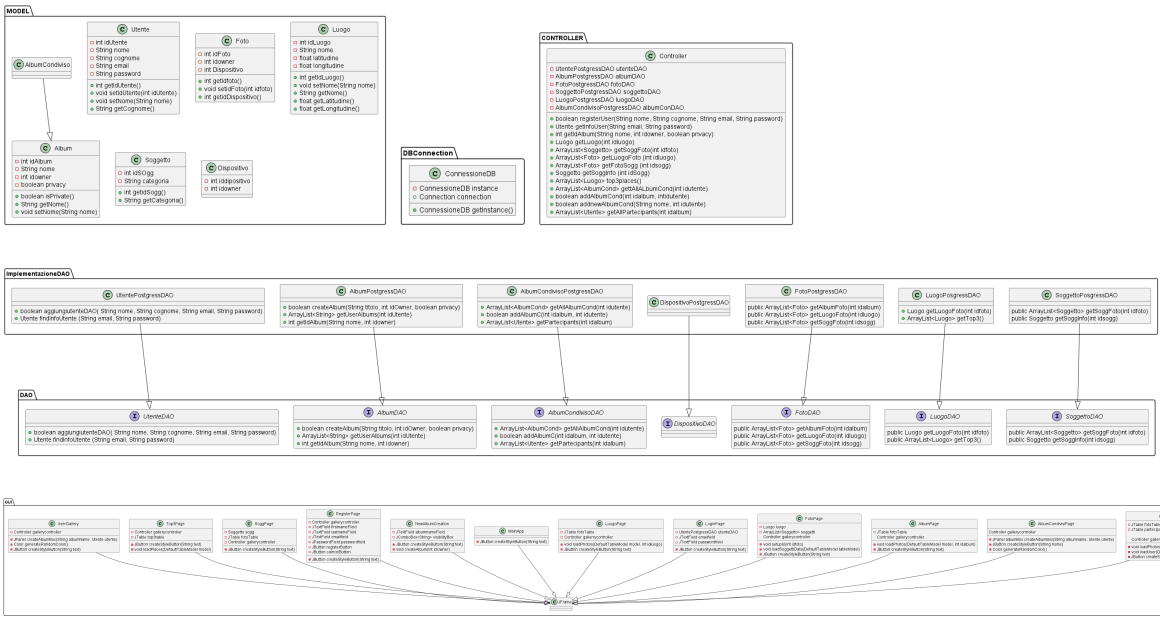
2.3 Dizionario delle Associazioni

Associazione	Classi Coinvolte	Descrizione
scattata_da...scatta	Utente...Foto	un utente può scattare una o più foto, la foto è scattata da un utente
gestito_da...gestisce	Utente...Album	un utente ha sempre un proprio album personale ogni album è creato da un utente
partecipa..partecipati_da	Utente...Album Condiviso	ogni utente può accedere ad uno o più album condivisi ogni album condiviso ha almeno un partecipante
rappresentato_da...rappresenta	Utente..Soggetto	ogni utente può essere o meno presente in foto e quindi essere un soggetto all'interno della foto
possiede...appartiene_a	Utente...Dispositivo	ogni utente può associare zero o più dispositivi ogni dispositivo è associato ad un solo utente
scattata_con...scatta	Foto...Dispositivo	una foto è scattata da un solo dispositivo un dispositivo può scattare da zero a molte foto
raffigura...raffigurato_in	Foto...Soggetto	un soggetto può essere presente in zero o più foto una foto può contenere zero o molti soggetti
presente_in...ha	Foto...Luogo	una foto può essere scattata in un solo luogo. un luogo può essere sede di scatti di numerose foto

Capitolo 3

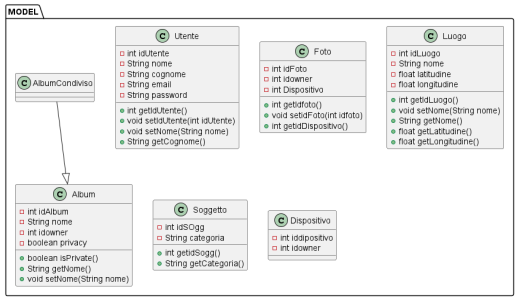
Progettazione della Soluzione

3.1 class diagram del dominio della soluzione



3.2 Package Model Diagram

Il seguente diagramma raffigura il package model implementato nel codice sorgente dell'applicativo

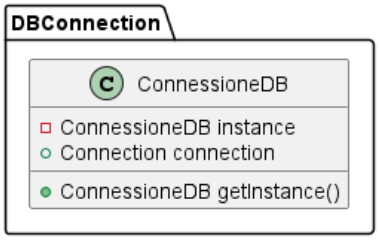


3.3 Comunicazione con il Database

Per la comunicazione con il Database, si è scelto di utilizzare il pattern architetturale DAO(Data Access Object). Tale pattern è utilizzato per fornire un'interfaccia unificata per l'accesso ai dati, in modo da separare l'applicazione dall'attività di archiviazione e recupero dei dati.

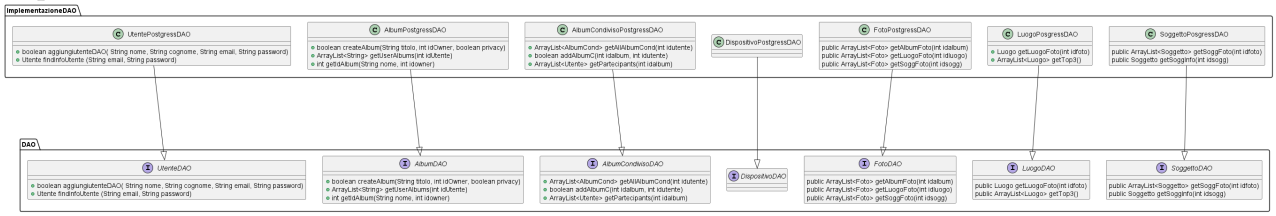
3.3.1 Connessione al Database

La connessione al database avviene all'intermo del package **DBConnection** il quale contiene la classe Connessione che si occupa di creare un'istanza con il database.



3.4 Package DAO e Package ImplPostgressDAO

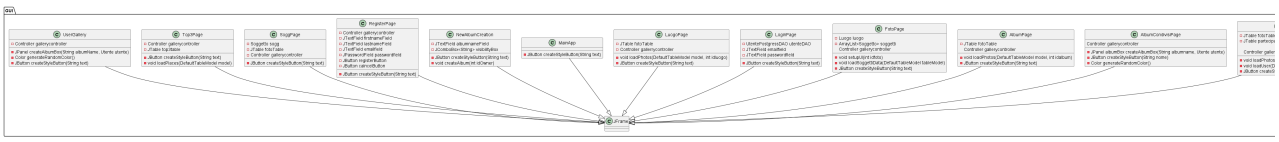
il package **DAO** è formato dalle interfacce DAO mentre il **ImplPostgressDAO** dalle relative implementazioni



3.5 Graphical User Interface

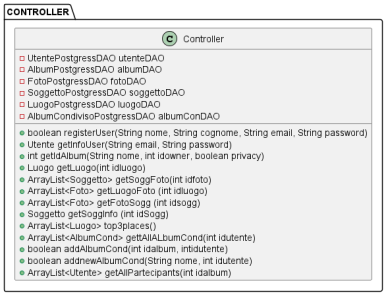
Questa sezione riguarda l'implementazione dell'interfaccia grafica. Come framework è stato scelto Swing. L'intrefaccia utente è implementata appunto nel package GUI e i file vengono gestiti attraverso questo pattern:

- **Main App**: è la prima schermata che appare all'avvio della applicazione. L'utente che accede può scegliere se registrarsi (in caso di nuvoo utente) oppure se effettuare il Login
- **RegisterPaga**: l'utente non ancora registrato, potrà farlo tramite questa schermata. Verrà pertanto aggiunto un nuovo utente al DB e da li verrà indirizzato alla sua pagina personale.
- **LoginPage**: pagina di log-in per l'utente che è già registrato
- **UserGallery**: pagina personale dell'utente il quale avrà la possibilità di vedere tutti gli album di cui fa parte (sia privati che condivisi con altri utenti) o di crearne altri, oltre alla possibilità di vedere quali sono i tre top luoghi immortalati.



3.6 Package Contorller

Il seguente diagramma illustra il package Controller con la sua relativa classe



3.7 Design pattern BCED

Il pattern utilizzato per creare l'applicativo Java e il design *Entity-controller-boundary-database*

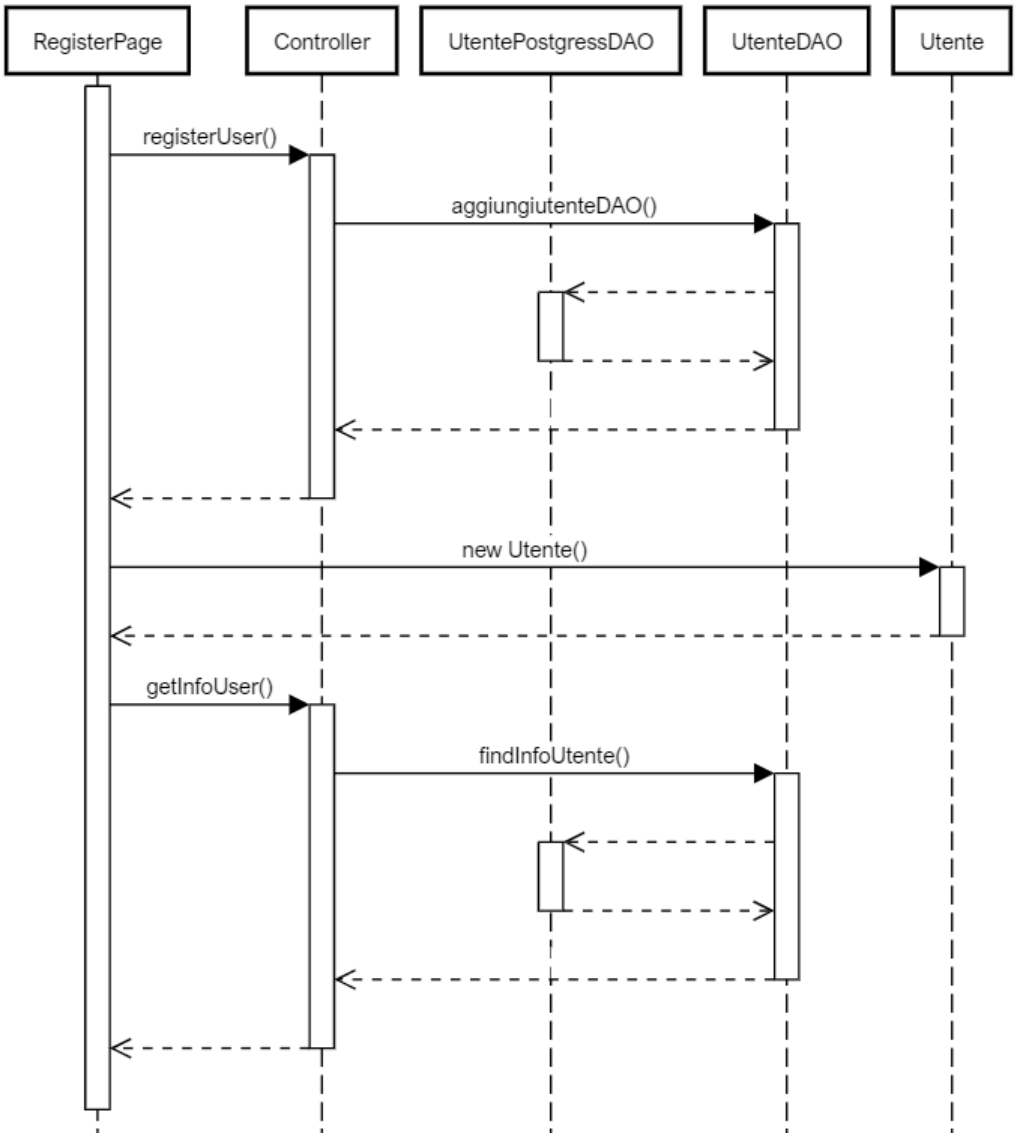
- **Entity**: l'insieme di classi che riproduce il diagramma del dominio del problema da rappresentare, in questo caso il package MODEL
- **Controller**: la classe controller ha appunto il ruolo di gestire la logica di controllo dell'applicazione, fungendo da intermediario tra il MODEL e l'interfaccia grafica.
- **Boundary**: ovvero il package GUI, l'interazione con l'utente
- **Database**: Package DAO, che gestisce appunto i dati del Database.

Capitolo 4

Sequence Diagram

In questa sezione verranno analizzati due sequence diagram per descrivere il flusso di computazione di due metodi.

4.1 Sequence Diagram Registrazione Utente



4.2 Sequence Diagram Creazione Album