

# Documentazione Galleria Fotografica Geolocalizzata

Gennaro Ventrone

Luigi Solaro

Mariano Sommella

ANNO ACCADEMICO 2022-2023



# Indice

<b>1</b>	<b>Modello del Dominio</b>	<b>1</b>
1.1	Introduzione Documentazione . . . . .	1
1.2	Analisi dei requisiti . . . . .	1
1.3	Dominio del Problema Generale . . . . .	2
<b>2</b>	<b>Dizionari</b>	<b>3</b>
2.1	Dizionario delle Entità . . . . .	3
2.2	Dizionario delle Associazioni . . . . .	5
<b>3</b>	<b>Model</b>	<b>6</b>
3.1	Traduzione delle classi e delle relazioni . . . . .	6
3.2	Package Model Diagram . . . . .	8
<b>4</b>	<b>Connessione con il Database</b>	<b>9</b>
4.1	Connessione al Database . . . . .	9
4.2	Package Connection Diagram . . . . .	10
<b>5</b>	<b>Package DAO</b>	<b>11</b>
5.1	Interfaccia GalleriaDAO . . . . .	11
5.2	Interfaccia CollezioneDAO . . . . .	11
5.3	Interfaccia FotografiaDAO . . . . .	11
5.4	Interfaccia FrameDAO . . . . .	11
5.5	Interfaccia SoggettoDAO . . . . .	11
5.6	Interfaccia UtenteDAO . . . . .	12
5.7	Interfaccia VideoDAO . . . . .	12
5.8	Interfaccia LuogoDAO . . . . .	12
5.9	Package Diagram DAO . . . . .	12
<b>6</b>	<b>Implementazioni del DAO</b>	<b>13</b>
6.1	Exception handling . . . . .	13
6.2	Package Diagram Implementazione . . . . .	14
<b>7</b>	<b>Package Diagram DAO Completo</b>	<b>15</b>
<b>8</b>	<b>Graphical User Interface</b>	<b>16</b>
8.1	Package GUI diagram . . . . .	18
<b>9</b>	<b>Design pattern BCED</b>	<b>19</b>
9.1	Package Controller . . . . .	20

<b>10 Dominio della Soluzione</b>	<b>21</b>
10.1 Solution Diagram . . . . .	21
<b>11 Sequence Diagram</b>	<b>22</b>
11.1 Sequence Diagram Inserimento Fotografia . . . . .	22
11.2 Sequence Diagram Aggiungi Foto Collezione . . . . .	22

# Capitolo 1

## Modello del Dominio

### 1.1 Introduzione Documentazione

Nelle seguenti pagine si riporta la documentazione relativa alla progettazione e implementazione di un applicativo java e la sua connessione con la basi di dati utilizzata per la gestione di collezioni fotografiche geolocalizzate condivise.

*Per una visione migliore del documento si utilizzi l'indice e le varie sezioni menzionate.*

### 1.2 Analisi dei requisiti

In questa sezione analizzeremo la richiesta dell'utente individuando le entità e associazioni del nostro mini world. Di seguito saranno elencate tutte le caratteristiche che il nostro applicativo dovrà garantire:

1. **Informazioni sulla foto:** Possibilità di visualizzare per ogni foto l'utente che l'ha scattata, il dispositivo utilizzato e il luogo in cui è stata scattata. Il luogo può essere identificato da coordinate geografiche o da un nome mnemonico unico all'interno del sistema.
2. **Identificazione dei soggetti:** Ogni foto può raffigurare diversi soggetti o utenti, in entrambi i casi devono essere identificati univocamente e categorizzati.
3. **Accesso a collezioni fotografiche:** Ogni utente ha sempre la possibilità di vedere la propria galleria fotografica personale, che comprende esclusivamente le foto scattate da lui. Un utente può partecipare a collezioni condivise con altri utenti che possono contenere foto scattate da questi ultimi.
4. **Accesso alla foto:** Ogni utente deve avere la possibilità di accedere solo alle proprie foto o a quelle condivise con gli altri utenti. Le foto private non verranno condivise con gli altri utenti.
5. **Eliminazione delle foto:** Gli utenti devono avere la possibilità di eliminare le proprie foto, ma solo l'amministratore del sistema può eliminare un utente e tutte le sue foto (eccetto le fotografie che contengono come soggetto un altro utente).

6. **Operazione di ricerca:** Il sistema deve permettere la ricerca di foto in base al luogo in cui sono state scattate o al soggetto rappresentato. Inoltre, deve essere possibile visualizzare una classifica dei 3 luoghi più immortalati.
7. **Gestione dei video:** Ogni fotografia può essere visualizzata in sequenza, andando a formare quindi un video.
8. **Sicurezza:** Il sistema deve garantire la sicurezza delle informazioni, in particolare la gestione degli account degli utenti e la protezione dei dati personali. Inoltre, il sistema deve essere in grado di gestire la concorrenza tra utenti per evitare problemi di accesso simultaneo alle stesse informazioni.

Il sistema prevede che le categorie di utenti siano così rappresentate:

- **Utente:** Può effettuare i punti dal 1 al 7 (In particolare esiste un servizio riservato al ruolo di *amministratore* nel punto 5).
- **Amministratore:** Può effettuare il punto 5 e avere accesso agli stessi servizi di *utente*.

### 1.3 Dominio del Problema Generale

Dopo aver preso in considerazione le idee discusse durante l'Analisi dei requisiti, è stato creato lo schema concettuale seguente, rappresentato attraverso un diagramma UML:

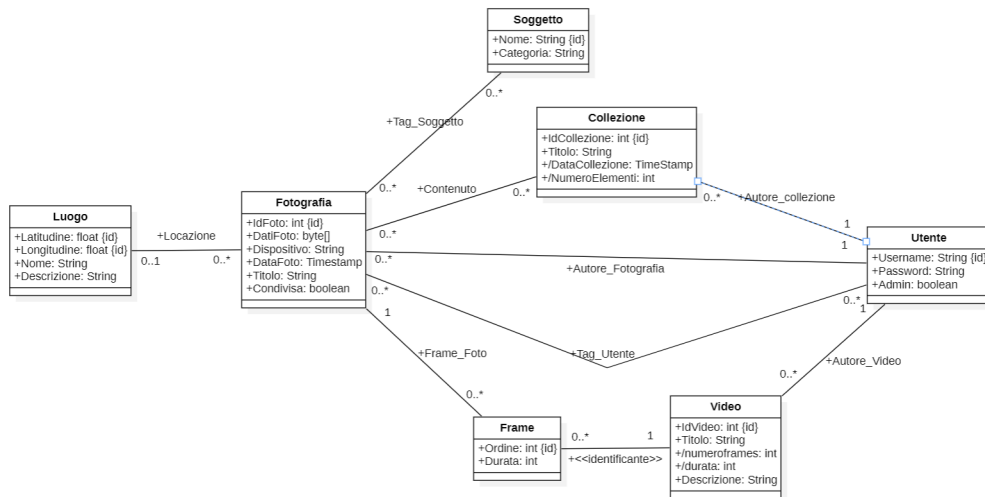


Figura 1.1: Diagramma Dominio Generale

## Capitolo 2

# Dizionari

### 2.1 Dizionario delle Entità

<b>Fotografia</b>	<b>Entità principale del nostro database, specifica tutte le caratteristiche legate alle fotografie</b>
<b>ID_Foto:</b>	L'identificativo primario di ogni foto
<b>Titolo:</b>	Titolo con cui è stata salvata la foto, per riconoscere il contenuto in modo immediato.
<b>Dati_foto:</b>	Dati di codifica di ogni fotografia
<b>Dispositivo:</b>	Specifica il dispositivo utilizzato per scattare la fotografia.
<b>Condivisa:</b>	Flag per gestire la privacy delle fotografie.
<b>Username_Autore:</b>	Autore della fotografia.
<b>Latitudine e Longitudine:</b>	Coordinate geografiche del luogo di scatto.
<b>Luogo</b>	<b>Entità che specifica la locazione dello scatto delle fotografia</b>
<b>Latitudine:</b>	Coordinata di Latitudine, usata come identificativo.
<b>Longitudine:</b>	Coordinata di Longitudine, usata come identificativo.
<b>Nome:</b>	Nome mnemonico, serve per facilitare all'utente il riconoscimento immediato del luogo dello scatto.
<b>Descrizione:</b>	Descrizione dettagliata del luogo, poiché lo stesso luogo può essere molto ampio, c'è bisogno di una descrizione.
<b>Collezione</b>	<b>Entità dove andranno inserite tutte le fotografie condivise</b>
<b>ID_Collezione:</b>	Identificativo univoco della Collezione
<b>Username:</b>	Autore della Collezione.
<b>Titolo:</b>	Titolo che sintetizza brevemente il contenuto della Collezione
<b>Data_Collezione:</b>	Data di creazione della Collezione
<b>Numero_Elementi:</b>	Numero di fotografie che sono all'interno della Collezione
<b>Soggetto</b>	<b>Entità che specifica il soggetto di una fotografia</b>
<b>Nome:</b>	Nome generico associato ad un soggetto, utilizzato come identificativo.
<b>Categoria:</b>	Categoria d'appartenenza del soggetto.

Tabella 2.1: Dizionario delle entità (Parte 1)

<b>Utente</b>	<b>Entità che specifica gli utenti dell'applicativo</b>
<b>Username:</b>	Username unico per ogni utente, così viene identificato all'interno del sistema.
<b>Password:</b>	Password dell'utente, indispensabile per poter accedere al sistema.
<b>Admin:</b>	Attributo booleano che conferisce il ruolo di amministratore di sistema ad un utente.
<b>Video</b>	<b>Entità che identifica i video creati dagli utenti</b>
<b>ID_Video:</b>	Identificativo del Video
<b>Autore:</b>	Autore del Video.
<b>Titolo:</b>	Titolo del video che ne sintetizza brevemente il contenuto.
<b>Numero_Frames:</b>	Il numero di frame che compongono il video.
<b>Durata:</b>	Tempo totale del video.
<b>Descrizione:</b>	Descrizione dettagliata del contenuto del video.
<b>Frame</b>	<b>Entità debole di Video, serve ad associare ogni frame singolo, composto da una fotografia, ad un Video</b>
<b>ID_Video:</b>	Identificativo del Video di appartenenza.
<b>Ordine:</b>	Indice in cui è localizzato il frame all'interno del Video
<b>ID_Foto:</b>	Identificativo della fotografica che rappresenta.
<b>Durata:</b>	Specifica il tempo che occupa un determinato frame all'interno di un video.

Tabella 2.2: Dizionario delle entità (Parte 2)

## 2.2 Dizionario delle Associazioni

<b>Autore_Collezione</b>
<b>Tipologia:</b> Uno a Molti (1:N) Associa un'Utente con le Collezione da esso create.
<b>Autore_Fotografia</b>
<b>Tipologia:</b> Uno a Molti (1:N) Associa un'utente alle fotografie che ha scattato.
<b>Autore_Video</b>
<b>Tipologia:</b> Uno a Molti (1:N) Associa un'utente ai video da esso creati.
<b>Frame_Foto</b>
<b>Tipologia:</b> Uno a Molti (1:N) Associa un Frame di un Video alla fotografia che rappresenta.
<b>Frame_Video</b>
<b>Tipologia:</b> Uno a Molti (1:N) Associa un singolo Frame al video di appartenenza.
<b>Locazione</b>
<b>Tipologia:</b> Uno a Molti (1:N) Associa una fotografia al Luogo in cui è stata scattata.
<b>Contenuto</b>
<b>Tipologia:</b> Molti a Molti (N:M) Associa Le Collezioni con le Fotografie che esse contengono.
<b>Tag_Soggetto</b>
<b>Tipologia:</b> Molti a Molti (N:M) Associa i Soggetti con le fotografie in cui sono taggati.
<b>Tag_Utente</b>
<b>Tipologia:</b> Molti a Molti (N:M) Associa Un'Utente alle fotografie in cui è taggato.

Tabella 2.3: Dizionario delle Associazioni



## Capitolo 3

# Model

### 3.1 Traduzione delle classi e delle relazioni

In questa sezione analizzeremo le classi e le associazioni del nostro applicativo. Possiamo trovare due tipi principali di relazioni: Uno a Molti (1:N) e Molti a Molti (N:M). Le relazioni Uno a Molti implicano che un record in una tabella può essere associato a molti record in un'altra tabella, mentre le relazioni Molti a Molti indicano che molti record in una tabella possono essere associati a molti record in un'altra tabella.

- **Relazioni Uno a Molti (1:N):** In questo tipo di relazione, un'istanza di una classe può essere associata a molte istanze di un'altra classe. Queste relazioni vengono implementate con riferimenti a oggetti della classe "molti" all'interno della classe "uno".
  - **Autore\_Collezione:** La relazione tra la classe "*Utente*" e la classe "*Collezione*" è una relazione Uno-a-Molti (1:N) **parziale**, in cui un'istanza di *Utente* può avere un riferimento a molte istanze di *Collezione*. Non tutti gli *Utenti*, tuttavia, devono necessariamente avere una *Collezione* correlata.
  - **Autore\_Fotografia:** Analogamente, la relazione tra la classe "*Utente*" e la classe "*Fotografia*" è una relazione Uno-a-Molti (1:N) **parziale**, dove un *Utente* può avere un riferimento a molte *Fotografie*. Anche in questo caso, non tutti gli *Utenti* devono avere *Fotografie* correlate.
  - **Autore\_Video:** La relazione tra la classe "*Utente*" e la classe "*Video*" segue lo stesso schema. Un *Utente* può avere un riferimento a molti *Video*, ma non tutti gli *Utenti* devono necessariamente avere un *Video* correlato.
  - 
  - **Frame\_Foto:** Nella relazione tra la classe "*Frame*" e la classe "*Fotografia*", un *Frame* può fare riferimento a una sola *Fotografia*, ma una *Fotografia* può essere utilizzata in molti *Frame* di diversi *Video*. Anche questa relazione è **parziale**, dal momento che non tutte le *Fotografie* devono necessariamente essere utilizzate in un *Frame*.

- **Frame\_Video:** La relazione tra la classe "*Video*" e la classe "*Frame*" è una relazione Uno-a-Molti (1:N) **totale**, dove un Video può avere un riferimento a molti Frame. Ogni Video deve avere almeno un Frame, quindi ogni istanza nel lato "uno" (Video) ha almeno una istanza correlata nel lato "molti" (Frame).
- **Locazione:** La relazione tra la classe "*Luogo*" e la classe "*Fotografia*" è una relazione Uno-a-Molti (1:N) **parziale**, in cui un Luogo può essere associato a molte Fotografie, ma ogni Fotografia è associata a un solo Luogo. Anche questa relazione è parziale, poiché non tutte le Fotografie devono necessariamente essere correlate a un Luogo.
- **Relazioni Molti a Molti (N:M):** In questo tipo di relazione, molte istanze di una classe possono essere associate a molte istanze di un'altra classe. Queste relazioni vengono implementate con liste di riferimenti alle classi coinvolte.
  - **Contenuto:** La relazione tra la classe "*Collezione*" e la classe "*Fotografia*" è una relazione Molti-a-Molti (N:M), in cui una Collezione può avere un riferimento a molte Fotografie e una Fotografia può essere contenuta in molte Collezioni.
  - **Tag\_Soggetto:** Analogamente, la relazione tra la classe "*Soggetto*" e la classe "*Fotografia*" è una relazione Molti-a-Molti (N:M), in cui un Soggetto può avere un riferimento a molti Tag e un Tag può essere associato a molti Soggetti.
  - **Tag\_Utente:** La relazione tra la classe "*Fotografia*" e la classe "*Utente*" segue lo stesso schema. Una Fotografia può avere un riferimento a molti Tag degli Utenti e un Utente può essere associato come Tag a molte Fotografie.

## 3.2 Package Model Diagram

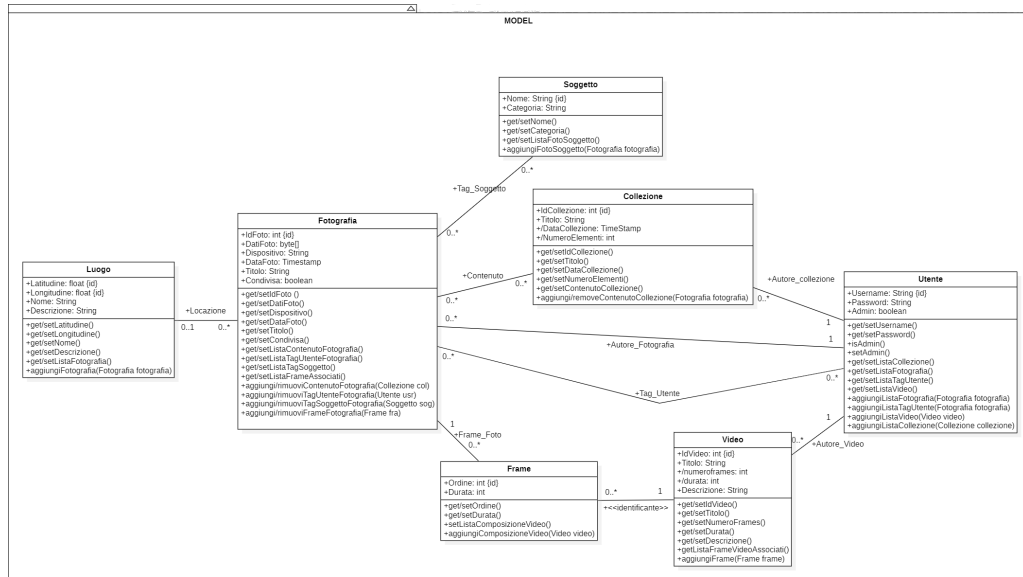


Figura 3.1: Diagramma UML Model

## Capitolo 4

# Connessione con il Database

L'interazione con il Database della Galleria avviene attraverso l'impiego del modello architetturale DAO (Data Access Object). Questo modello facilita la persistenza dei dati e la regolazione dei loro accessi, offrendo un'interfaccia unificata per il recupero e l'archiviazione dei dati, isolando l'applicazione dai dettagli di come i dati vengono gestiti.

### 4.1 Connessione al Database

La classe `Connessione`, contenuta nel pacchetto `Connessione DBconnection`, si incarica di stabilire una connessione con il database Postgres. Abbiamo optato per mantenere il database online, consentendo così all'applicativo di funzionare su diversi dispositivi, senza la necessità di creare un database locale. Per ulteriori dettagli sul database, si consiglia di consultare la sottocartella del progetto che contiene l'implementazione del database utilizzato.

## 4.2 Package Connection Diagram

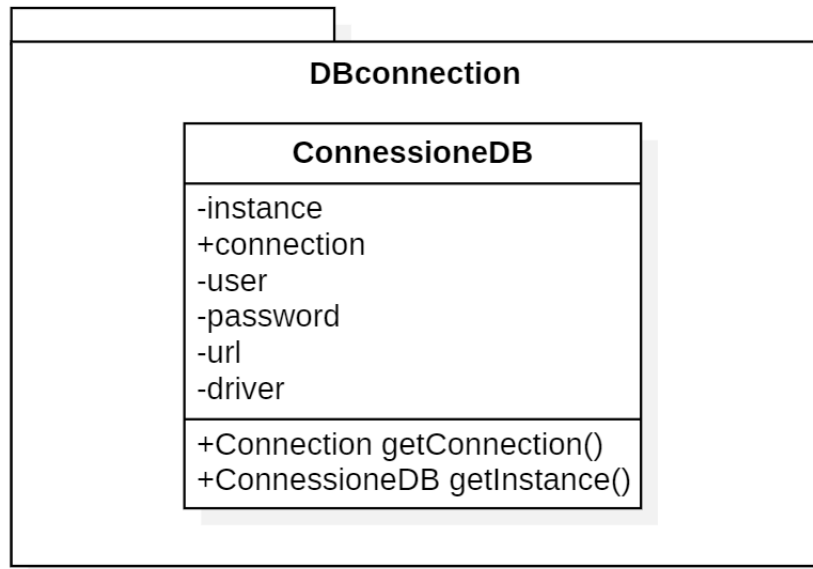


Figura 4.1: Diagramma Connection

## Capitolo 5

# Package DAO

### 5.1 Interfaccia GalleriaDAO

Abbiamo separato le funzioni di caricamento dei dati iniziali dalle subroutine utilizzate durante l'esecuzione del programma. L'interfaccia GalleriaDAO si occupa di recuperare dal database i dati iniziali relativi a fotografie, video, collezioni, frame, utenti, soggetti e anche luoghi degli scatti.

### 5.2 Interfaccia CollezioneDAO

L'interfaccia CollezioneDAO si occupa della creazione, modifica o eliminazione di una collezione condivisa. Essa implementa anche dei metodi utili per l'aggiunta di un nuovo elemento alla collezione o per la visualizzazione del contenuto della collezione stessa.

### 5.3 Interfaccia FotografiaDAO

L'interfaccia FotografiaDAO garantisce la creazione, modifica o eliminazione di una fotografia. Inoltre, questa interfaccia si occupa anche di gestire l'aggiunta, l'eliminazione e la visualizzazione dei tag\_utente, dei tag\_soggetto e delle collezioni condivise in cui la fotografia è presente.

### 5.4 Interfaccia FrameDAO

L'interfaccia FrameDAO, oltre a gestire la creazione, modifica o eliminazione di un frame, fornisce anche la possibilità di visualizzare il video a cui il frame appartiene.

### 5.5 Interfaccia SoggettoDAO

L'interfaccia SoggettoDAO garantisce la creazione, modifica o eliminazione di un soggetto e fornisce anche la possibilità di visualizzare la fotografia in cui il soggetto è raffigurato.

## 5.6 Interfaccia UtenteDAO

L'interfaccia UtenteDAO si occupa della creazione, modifica o eliminazione di un utente dell'applicativo. Inoltre, offre anche la possibilità di visualizzare le fotografie scattate dall'utente, i video di sua proprietà e le collezioni da lui create

## 5.7 Interfaccia VideoDAO

L'interfaccia VideoDAO assicura la creazione, modifica o eliminazione di un video e fornisce anche la possibilità di visualizzare i singoli frame al suo interno.

## 5.8 Interfaccia LuogoDAO

L'interfaccia LuogoDAO assicura la creazione, modifica o eliminazione di un luogo e fornisce anche la classifica dei 3 luoghi più immortalati.

## 5.9 Package Diagram DAO

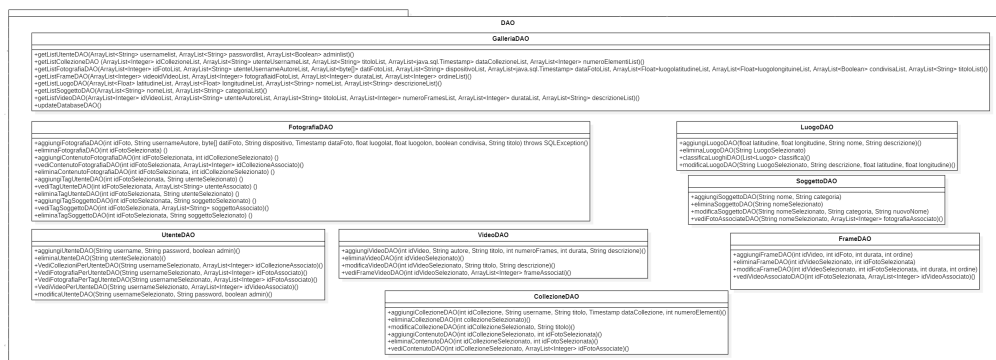


Figura 5.1: Diagramma Package DAO

## Capitolo 6

# Implementazioni del DAO

Le interfacce DAO definiscono i metodi per creare, leggere, aggiornare ed eliminare (Create, Read, Update, Delete) gli oggetti correlati a una specifica entità. Le classi di implementazione DAO forniscono l'implementazione concreta di questi metodi per un determinato tipo di database. Nel contesto delle implementazioni del DAO, vengono fornite le implementazioni specifiche delle interfacce DAO per un database Postgres nel package "ImplementazioniPostgresDAO". Queste implementazioni utilizzano le funzionalità offerte dal driver JDBC (Java Database Connectivity) per comunicare con il database. Attraverso l'implementazione del DAO, è possibile isolare la logica di accesso ai dati all'interno delle classi DAO, consentendo alle altre parti dell'applicazione di interagire con il database tramite un'interfaccia standardizzata. Ciò rende il codice più modulare, leggibile e facilmente manutenibile.

### 6.1 Exception handling

Nell'ambito della gestione delle eccezioni, è fondamentale affrontare i potenziali errori che possono verificarsi durante l'esecuzione dell'applicazione. In particolare, nel contesto del DAO, è importante gestire le eccezioni che possono essere sollevate dalle operazioni di accesso ai dati quando si verificano violazioni di vincoli o errori nel database.

L'obiettivo principale della gestione delle eccezioni è di catturare e gestire gli errori in modo appropriato, fornendo feedback all'utente sull'errore verificatosi e consentendo di prendere le azioni necessarie per correggerlo.

Nel caso delle eccezioni provenienti dai vincoli imposti sui dati nel database, si è scelto di inviare gli errori alla GUI e quindi all'utente. Quando il database rileva una violazione di un vincolo, genera un'eccezione corrispondente con un messaggio di errore. Questa eccezione viene catturata all'interno del DAO e il messaggio di errore viene quindi visualizzato direttamente all'utente tramite l'interfaccia grafica. È compito dell'utente gestire l'errore e prendere le opportune azioni correttive, ad esempio modificando la data di assunzione in caso di violazione di un vincolo di data.



## 6.2 Package Diagram Implementazione

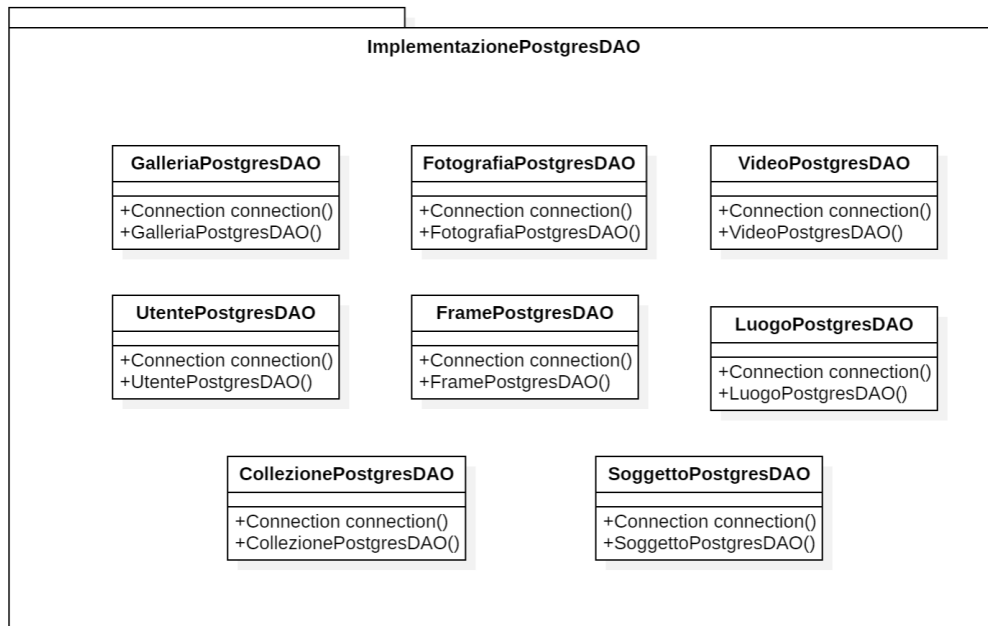


Figura 6.1: Diagramma ImplementazionePostgreDAO

## Capitolo 7

# Package Diagram DAO Completo

In questa sezione abbiamo il grafico completo che illustra in che modo è stata sviluppata l'architettura DAO all'interno del nostro applicativo

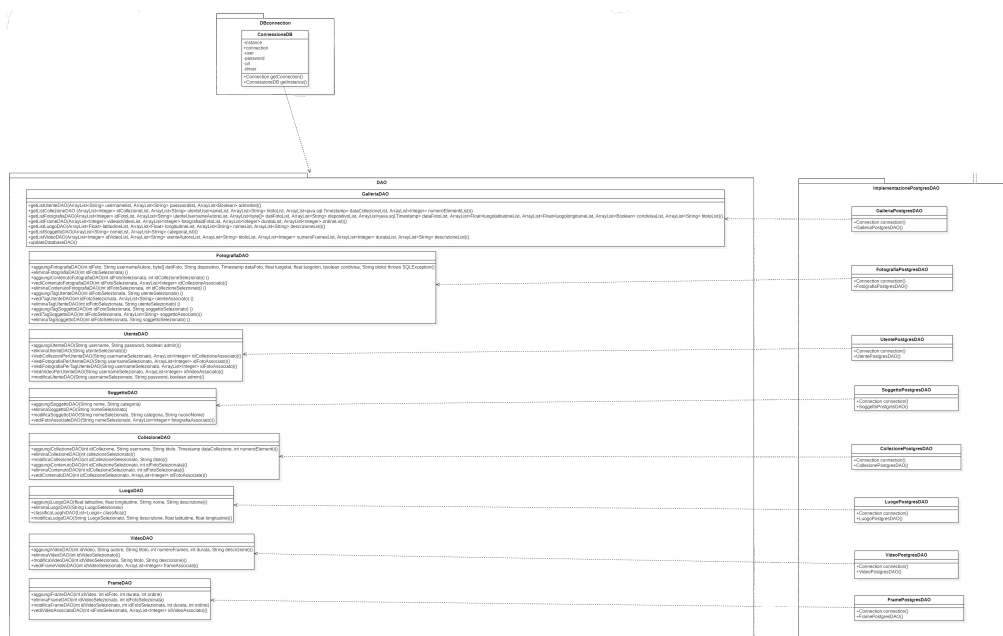


Figura 7.1: Diagramma Package DAO

## Capitolo 8

# Graphical User Interface

In questa sezione viene descritto il modo in cui l'interfaccia grafica è stata implementata. Inanzitutto è stato scelto come framework Swing, appartenente alle Java Foundation Classes (JFC). L'interfaccia utente è implementata nel package GUI e i file vengono gestiti attraverso questo pattern :

- **Login:** All'inizio, l'applicativo presenta all'utente una schermata di Login, dove l'utente è tenuto a inserire le proprie credenziali.
- **Menu Principale:** Questo è il menù centrale da cui l'utente può navigare verso le diverse sezioni dell'applicativo. Offre all'utente la possibilità di navigare verso vari sottomenù attraverso bottoni interattivi. Ogni sottomenù permette di visualizzare tabelle specifiche del database, a seconda dell'utente che ha effettuato l'accesso.
- **MenuUtenti:** Questo sottomenù permette di visualizzare, modificare, inserire ed eliminare gli utenti dell'applicativo.
  - **ProfiloUtente:** Questa sezione consente la modifica delle informazioni dell'utente, la visualizzazione di tutte le collezioni e i video creati da lui e mostra anche tutti i tag\_utente ricevuti nelle fotografie.
  - **InserimentoUtente:** Questa sezione offre la funzionalità di aggiungere un nuovo utente.
- **MenuCollezioni:** Questo sottomenù fornisce una vista di tutte le collezioni condivise presenti nel database, offrendo la possibilità di modificarle, inserirne di nuove ed eliminarle.
  - **ProfiloCollezione:** Questa sezione offre la possibilità di modificare una collezione, oltre a permettere l'aggiunta e la rimozione dei contenuti della galleria e a mostrare il contenuto stesso.
  - **InserimentoCollezione:** Questa sezione consente la creazione di una nuova collezione.
- **MenuFotografie:** Questo sottomenù presenta tutte le fotografie dell'utente attualmente loggato e tutte quelle condivise. Offre la possibilità di modificarle, inserirne di nuove ed eliminarle.
  - **ProfiloFotografia:** A In questa sezione, oltre alla possibilità di modificare una foto, è possibile aggiungere tag utente e tag soggetto.

- **InserimentoFotografia:** Questa sezione offre la possibilità di aggiungere una nuova fotografia al database.
- **MenuLuogo:** Questo sottomenù fornisce una vista di tutti i luoghi nel database, offrendo la possibilità di modificarli, inserirne di nuovi ed eliminarli.
  - **ProfiloLuogo:** In questa sezione, oltre alla possibilità di modificare un luogo, è possibile visualizzare tutte le fotografie associate a quel luogo.
  - **InserimentoLuogo:** Questa sezione permette l'aggiunta di un nuovo luogo all'applicativo.
- **ClassificaLuoghi:** : Questa sezione serve a rispondere a un'esigenza dell'utente specificata nella (SEZIONE 1.2).
- **MenuSoggetto:** Questo sottomenù mostra tutti i soggetti e offre la possibilità di modificarli, inserirne di nuovi ed eliminarli.
  - **ProfiloSoggetto:** In questa sezione, oltre alla possibilità di modificare un soggetto, è possibile visualizzare tutte le fotografie associate a quel soggetto.
  - **InserimentoSoggetto:** Questa sezione consente l'aggiunta di un nuovo soggetto all'applicativo.
- **MenuVideo:** Questo sottomenù fornisce una vista di tutti i video presenti nell'applicativo e offre la possibilità di modificarli, inserirne di nuovi ed eliminarli. Inoltre, consente di visualizzare un video come un insieme di frame e di eseguire le stesse operazioni su di essi.
  - **ProfiloVideo:** Questa sezione consente la visualizzazione e la modifica del video.
  - **InserimentoVideo:** Questa sezione consente l'aggiunta di un nuovo video all'applicativo.
  - **MenuFrame:** Questo sottomenù mostra tutti i frame di un determinato video e consente di modificarli, inserirne di nuovi ed eliminarli.
    - \* **ModificaFrame:** Questa sezione consente la modifica delle informazioni di un particolare frame.
    - \* **InserisciFrame:** Questa sezione consente l'aggiunta di un nuovo frame a un video.

## 8.1 Package GUI diagram

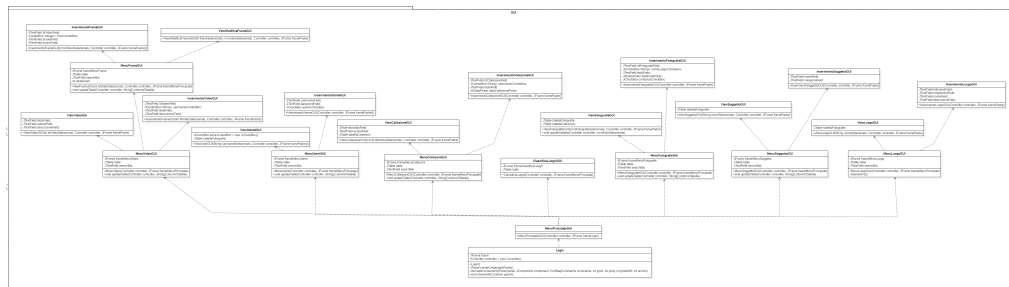


Figura 8.1: Diagramma Package GUI

## Capitolo 9

# Design pattern BCED

Prima di procedere con il dominio della soluzione, è necessario identificare il pattern utilizzato per la creazione dell'applicativo Java. Abbiamo deciso di adottare una versione estesa del design pattern BCE (Boundary - Controller - Entity) chiamata BCED (Boundary - Controller - Entity - Database). Poiché nel modello originale non è prevista la memorizzazione permanente dei dati, abbiamo dovuto aggiungere la parte mancante relativa al database.

- **Boundary** : Sta ad indicare l'applicativo grafico con il quale l'utente interagisce, il tutto è esplicitato all'interno del package GUI. (SEZIONE 8)
- **Controller** : Svolge il ruolo coordinatore tra l'interfaccia utente (Boundary), la logica del business (Entity) e il database. Gestisce l'input dell'utente, esegue operazioni sulle Entity e interagisce con il database in base a tale input. Successivamente, comunica i risultati al Boundary per aggiornare l'interfaccia utente. In questo modo, il Controller svolge un ruolo fondamentale nel mantenere separati i dettagli dell'interfaccia utente, la logica del business e la gestione dei dati nel database, promuovendo un design più pulito, modulare e mantenibile. L'implementazione del controller è esplicitata nel diagramma del Package Controller.(SEZIONE 9.1)
- **Entity**: Sono l'insieme di classi che riproducono il diagramma del dominio, affrontato durante la creazione del package MODEL. (SEZIONE 3)
- **Database** : Sta ad indicare la base di dati utilizzata e la sua implementazione, esplicitata durante la stesura del package DAO. (SEZIONE 5)

## 9.1 Package Controller



Figura 9.1: Diagramma Package Controller

## Capitolo 10

# Dominio della Soluzione

A seguito delle considerazioni effettuate nei capitoli precedenti, possiamo definire il dominio della soluzione adottata.

### 10.1 Solution Diagram



Figura 10.1: Diagramma Soluzione



## Capitolo 11

# Sequence Diagram

In quest'ultima sezione andremo ad utilizzare come esempio due diagrammi di sequenza per descrivere il flusso di computazione di due operazioni molto frequenti all'interno dell'applicativo.

### 11.1 Sequence Diagram Inserimento Fotografia

Il primo Sequence Diagram illustra cosa succede all'interno dell'applicativo quando viene inserita una nuova fotografia.

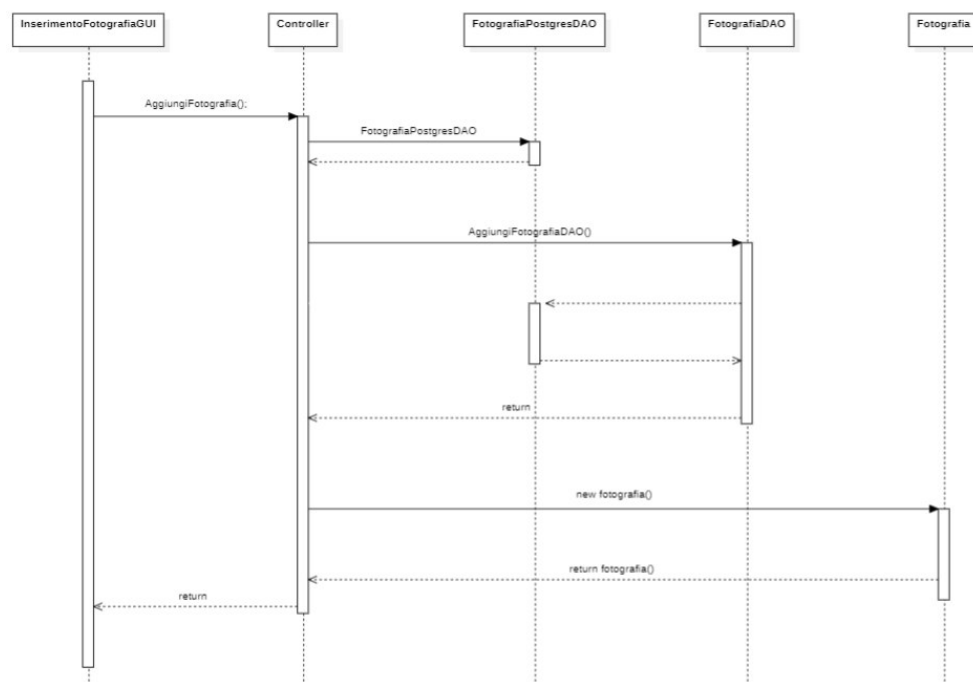


Figura 11.1: Diagramma Di Sequenza Fotografia

### 11.2 Sequence Diagram Aggiungi Foto Collezione

Il secondo Sequence Diagram fa vedere cosa accade ogni volta che una Fotografia viene aggiunta all'interno di una Collezione

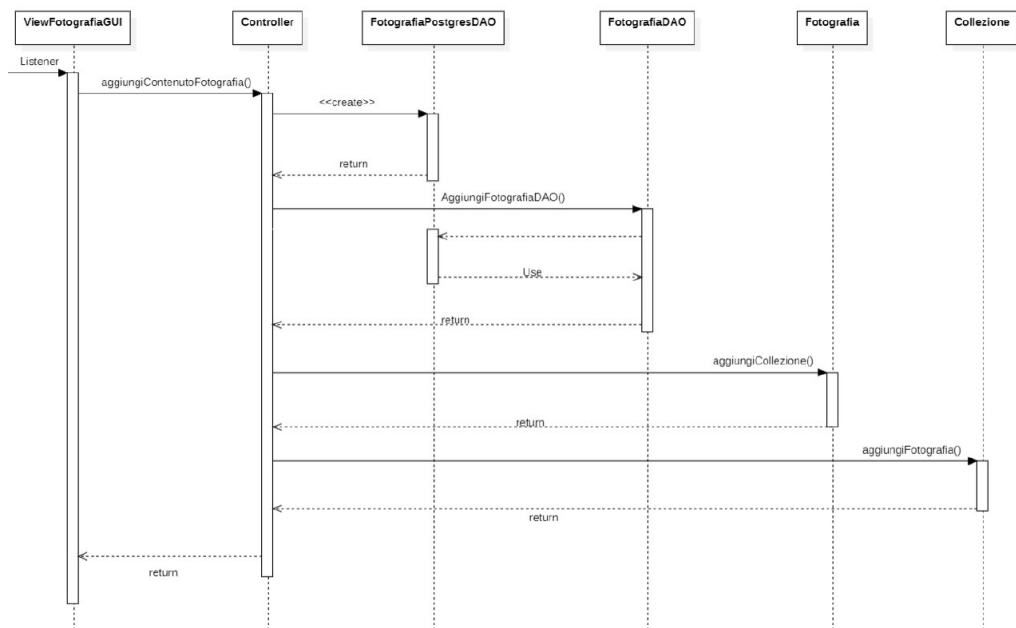


Figura 11.2: Diagramma Di Sequenza Contenuto