

UNIVERSITÀ DEGLI STUDI DI MODENA E  
REGGIO EMILIA

DIPARTIMENTO DI INGEGNERIA "ENZO FERRARI"

## Tesina Ingegneria di Tecnologie Web

Autore:

**Centanni Samuele**

---

Anno Accademico 2023/2024

# Indice

<b>1</b>	<b>Descrizione del Progetto</b>	<b>3</b>
1.1	Dashboard . . . . .	3
1.2	Immagini . . . . .	4
1.2.1	Immagini nel dettaglio . . . . .	4
1.3	Utenti . . . . .	5
1.3.1	Utenti nel dettaglio . . . . .	6
1.4	Immagini più viste . . . . .	6
1.5	Le tue immagini . . . . .	7
1.6	Modifica Profilo, Password e Visibilità . . . . .	7
<b>2</b>	<b>UML</b>	<b>8</b>
2.1	Use Case . . . . .	8
2.2	Class Diagram . . . . .	9
<b>3</b>	<b>Organizzazione Progetto</b>	<b>10</b>
3.1	Organizzazione App . . . . .	10
3.1.1	Account . . . . .	10
3.1.2	Images . . . . .	10
3.1.3	Actions . . . . .	10
<b>4</b>	<b>Test Fatti</b>	<b>11</b>
4.1	Test della View . . . . .	11
4.2	Test del codice applicativo . . . . .	11
<b>5</b>	<b>Descrizione Tecnologie Usate</b>	<b>13</b>
5.1	Redis . . . . .	13
5.2	Pillow e Sorl-Thumbnail . . . . .	14
5.3	Requirements e Dipendenze . . . . .	14

# Capitolo 1

## Descrizione del Progetto

Il mio progetto si basa su sito web che emula le caratteristiche principali di Instagram. In particolare, gli utenti possono decidere di registrarsi al sito: in fase di registrazione sono chiesti i dati necessari ad identificare quell'utente all'interno del sito web.

Ogni utente, all'interno della sua area, può anzitutto modificare le proprie informazioni, come ad esempio aggiornare la propria foto profilo (gliene viene assegnata una di default in fase di registrazione), e la propria password.

Dopodiché ha a disposizione 5 sezioni:

- **Dashboard**
- **Immagini**
- **Utenti**
- **Immagini più viste**
- **Le tue immagini**

### 1.1 Dashboard

In questa sezione l'utente è in grado sia di caricare le proprie immagini, con indicazione di quante immagini ha caricato fino a quel momento, sia di poter caricare immagini da siti esterni. Può inoltre vedere le azioni recenti dei profili che segue. Ogni utente infatti può compiere una serie di azioni che tutti i profili che lo seguono potranno vedere:

- utente X ha messo like ad immagine Y
- utente X ha creato account
- utente X ha iniziato a seguire utente Z
- utente X ha commentato immagine Y

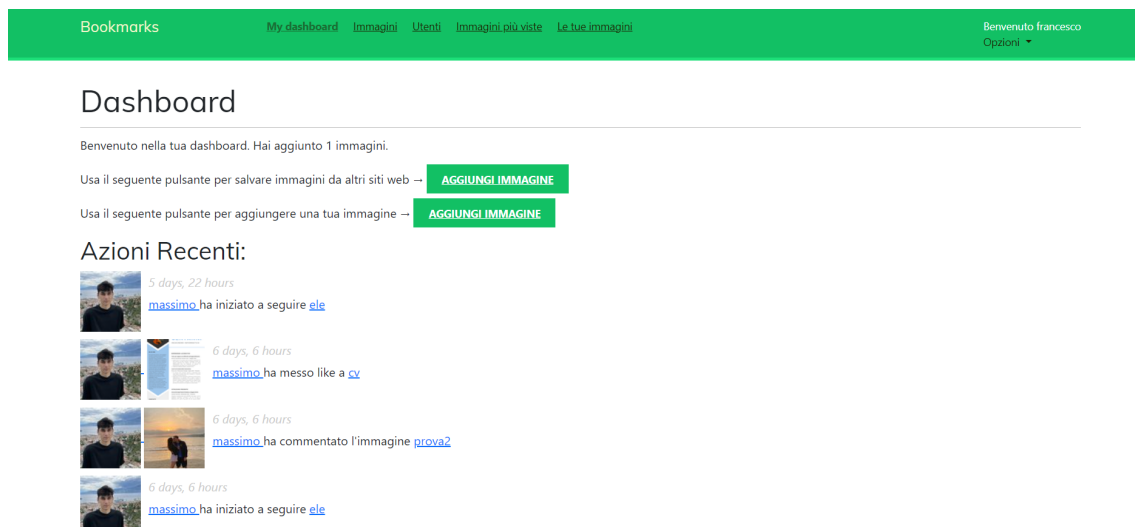


Figura 1.1: Dashboard

## 1.2 Immagini

In questa sezione l'utente è in grado di vedere **tutte le immagini globali** caricate da **tutti gli utenti con profilo pubblico**. Se un profilo è privato, un altro utente potrà vedere le immagini di quel profilo privato in questa sezione se e solo segue quell'utente con profilo privato.

### Immagini Caricate



Figura 1.2: Immagini Caricate

### 1.2.1 Immagini nel dettaglio

Per ogni immagine, se è visualizzabile, l'utente può:

- mettere like o togliere like
- commentare l'immagine e leggere eventuali commenti

- vedere quanti like ha una immagine e quante volte è stata visualizzata
- vedere le foto profilo degli utenti che hanno messo like a quell'immagine

Se l'utente sta visualizzando una propria immagine, è in grado anche di poterla eliminare.

CV

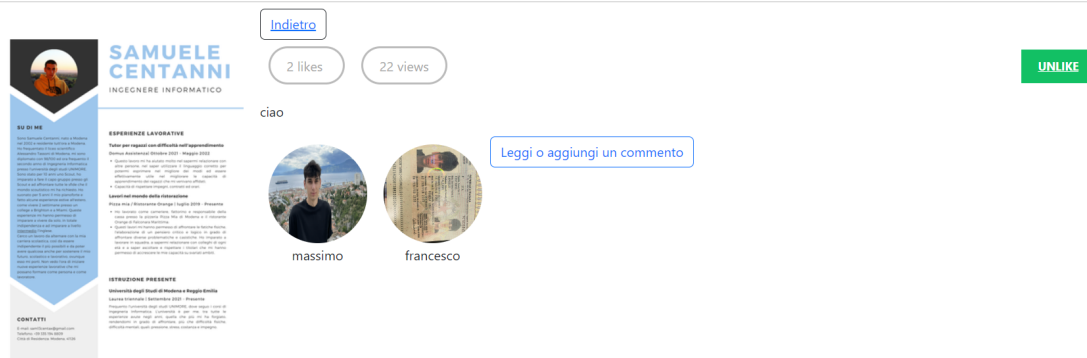


Figura 1.3: Dettaglio Immagine

## 1.3 Utenti

In questa sezione l'utente può vedere l'elenco di tutte gli utenti registrati alla piattaforma, con possibilità di cercare l'utente per nome o username.

Utenti

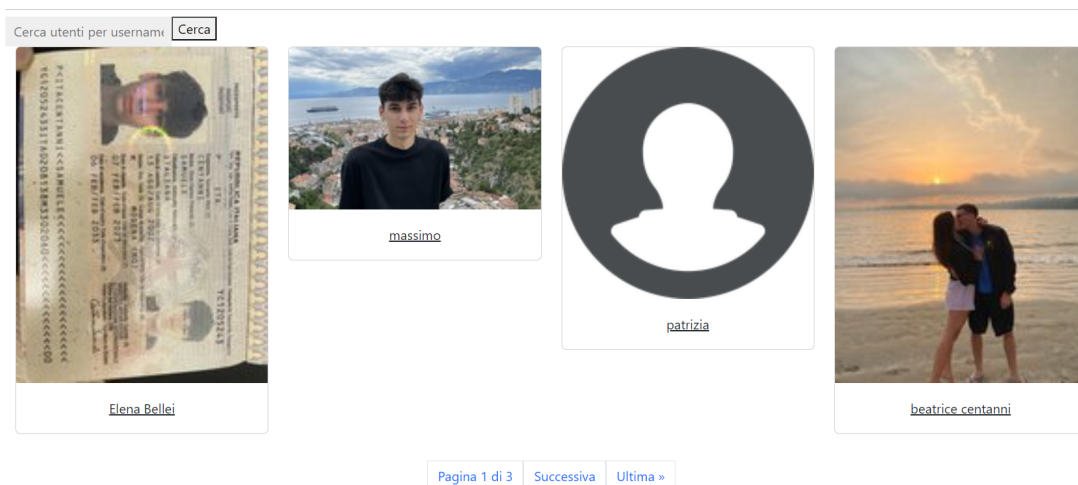


Figura 1.4: Utenti

### 1.3.1 Utenti nel dettaglio

Per ogni utente, se esso è privato non verrà mostrata nessuna informazione, ma solamente un tasto "Follow" per iniziarlo a seguire. Se l'utente invece è pubblico, oppure fa parte degli utenti che seguo, potrò vedere:

- le immagini che ha caricato
- quante persone lo seguono



Figura 1.5: Dettaglio Profilo Pubblico

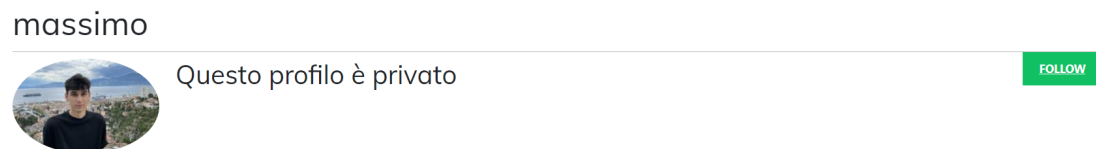


Figura 1.6: Dettaglio Profilo Privato

## 1.4 Immagini più viste

Classifica delle immagini più visualizzate.

## Immagini più viste

---

1. [Django and Duke](#)
2. [prova2](#)
3. [prova](#)
4. [provaBea](#)
5. [cv](#)
6. [Django and Duke2](#)

Figura 1.7: Immagini più viste

## 1.5 Le tue immagini

Sezione in cui l'utente può visualizzare unicamente le immagini che ha pubblicato.

### Le Tue Immagini

---

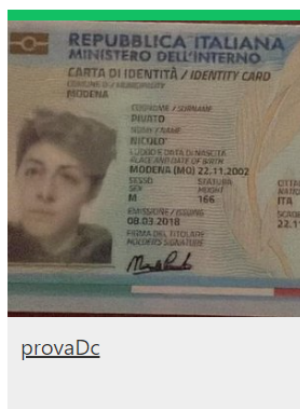


Figura 1.8: Le Tue Immagini Caricate

## 1.6 Modifica Profilo, Password e Visibilità

L'utente, come anticipato, ha la possibilità, tramite un menù a tendina, di poter modificare il proprio profilo, la propria password e di poter impostare il proprio profilo come privato o pubblico.

# Capitolo 2

## UML

### 2.1 Use Case

A livello di progettazione del software, lo Use Case rappresenta gli utenti del mio sito e le loro possibili azioni. Per il mio progetto, lo Use Case è molto semplice, infatti ho solamente un attore: l'utente registrato.

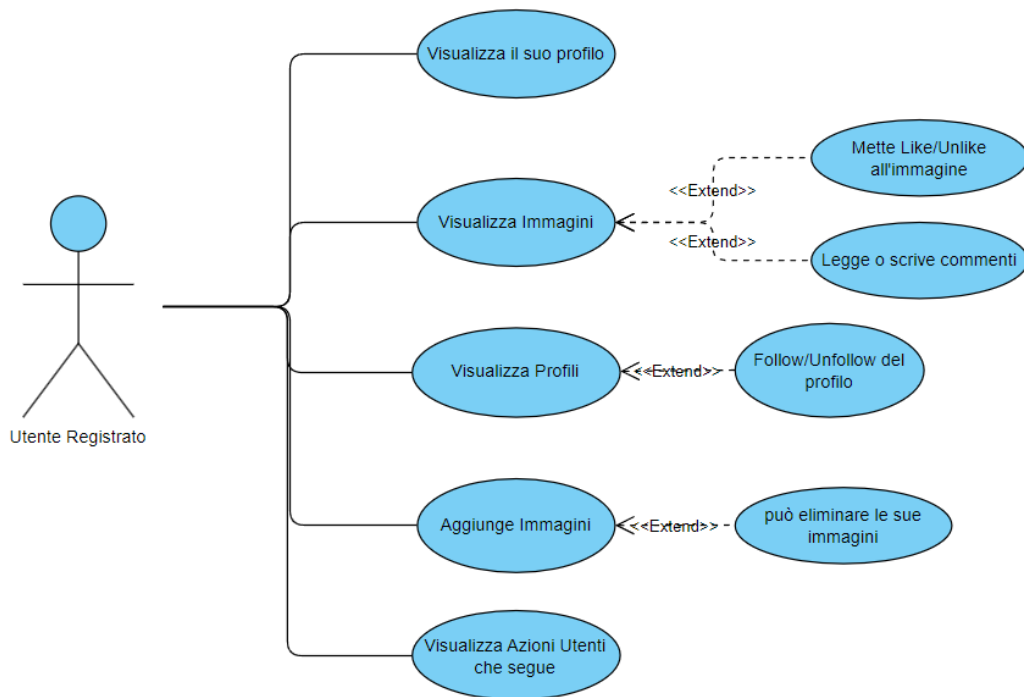


Figura 2.1: Use Case



## 2.2 Class Diagram

Per quanto riguarda l'organizzazione del progetto, fare riferimento al capitolo 3.

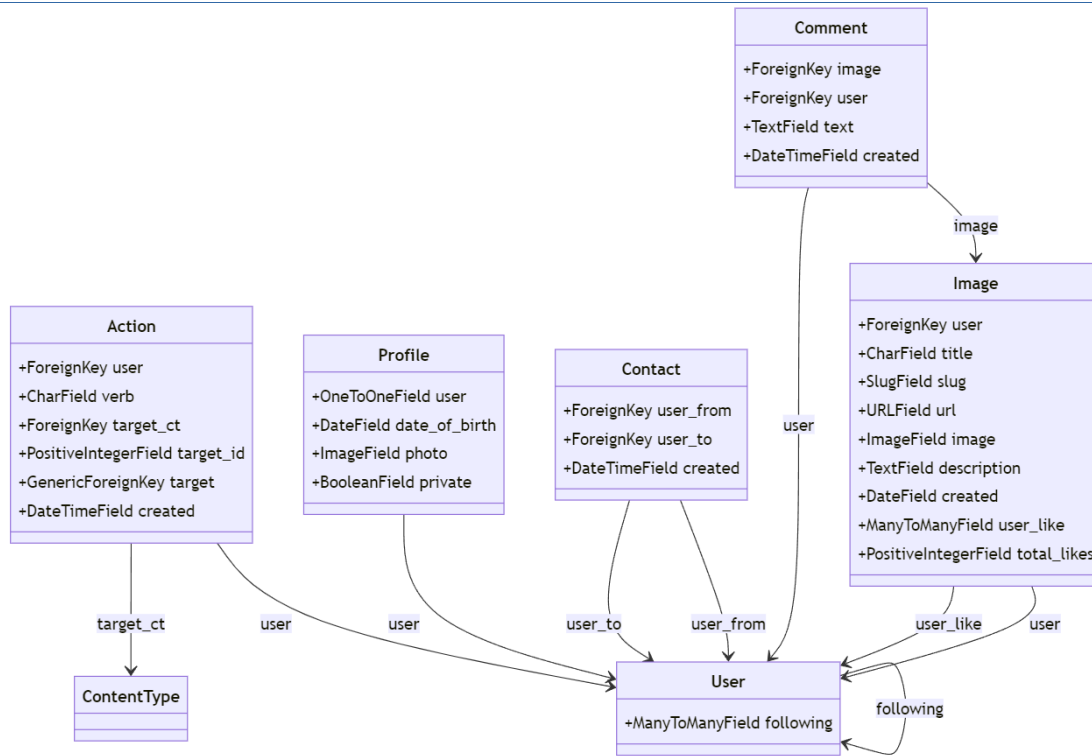


Figura 2.2: Class Diagram

# Capitolo 3

## Organizzazione Progetto

### 3.1 Organizzazione App

Ho deciso di realizzare 3 app:

1. app Actions,
2. app Account,
3. app Images.

Ho deciso di realizzare queste 3 app in modo tale da poter gestire in modo autonomo le immagini e gli account.

#### 3.1.1 Account

Gestisco gli utenti, con le loro informazioni e foto profilo, e le interazioni tra utenti (follow e unfollow degli utenti).

#### 3.1.2 Images

L'applicazione Images invece gestisce le immagini caricate dagli utenti, con i likes, i commenti e la visualizzazione.

#### 3.1.3 Actions

L'applicazione Actions serve invece per poter gestire le azioni degli utenti, in modo tale da poter gestire:

1. chi ha compiuto l'azione,
2. su cosa è stata compiuta l'azione,
3. quando è stata compiuta l'azione.

# Capitolo 4

## Test Fatti

Per quanto riguarda i testing fatti, ho deciso di testare una view ed una funzione di codice applicativo.

### 4.1 Test della View

Ho testato la porzione di codice che si occupa di gestire la `DetailView` di ogni utente. Nel mio progetto infatti gli utenti possono avere profili pubblici e privati, e quest'ultimi possono essere visualizzabili:

- da utenti che seguono quel profilo,
- dall'utente proprietario di quel profilo privato.

Se un profilo che non non segue quell'utente vuole visualizzarne la sua `DetailView`, visualizzerà un messaggio del tipo "Questo profilo è privato".

Il testing mira a:

1. verificare che un utente non loggato non possa visualizzare nulla e venga rediretto alla pagina di Login,
2. verificare che un utente correttamente loggato e che segue il nostro profilo PRIVATO, possa vedere tutto,
3. verificare che un utente correttamente loggato e che NON segue il nostro profilo PRIVATO, possa vedere il profilo ma veda solamente la scritta "Questo profilo è privato".

### 4.2 Test del codice applicativo

Ho testato il metodo `'create_action()'`, che si occupa di gestire le azioni degli utenti e di occuparsi di salvare solo le azioni non ripetute più volte dallo stesso utente.

In particolare, il testing si occupa di verificare che:

1. l'azione venga correttamente creata,
2. se lo stesso utente ha fatto la stessa azione in meno di un minuto, allora non deve crearsi nessuna azione,
3. se però lo stesso utente ha fatto la stessa azione dopo un minuto, allora deve crearsi un'azione.

# Capitolo 5

## Descrizione Tecnologie Usate

### 5.1 Redis

Ho deciso di utilizzare Redis per gestire quante volte un'immagine viene visualizzata, ed usare questa informazione per gestire l'elenco di immagini più viste nella mia piattaforma.

La scelta di Redis è stata fatta per due motivi:

1. **Performance e Scalabilità:** A differenza di un database relazionale che utilizzerebbe operazioni di aggiornamento SQL per ogni visualizzazione di immagine, Redis offre un'alternativa significativamente più veloce. Con Redis infatti, è sufficiente incrementare un contatore in memoria per ogni visualizzazione, eliminando la necessità di costose operazioni di scrittura su disco. Questa architettura in memoria di Redis si traduce in una risposta più rapida alle richieste degli utenti, migliorando l'esperienza complessiva sulla piattaforma.
2. **Approccio Alternativo:** Optare per Redis ha rappresentato anche una scelta per aggiungere complessità e diversità tecnologica al mio progetto. Rispetto alle soluzioni apprese durante il corso, l'utilizzo di Redis mi ha permesso di esplorare un nuovo strumento potente e versatile. Questa decisione ha quindi contribuito ad ottimizzare la gestione delle informazioni dinamiche all'interno della webapp.

Infatti, mentre SQL è ideale per lo storage dati persistente definito dallo schema, Redis offre numerosi vantaggi quando si tratta di dati che cambiano rapidamente, o quando è necessaria una cache veloce, come nel caso di aggiornare rapidamente il conteggio di quante volte viene visualizzata un'immagine.

Se memorizzassimo il numero totale di volte in cui un'immagine viene visualizzata usando l'ORM di Django, ciò comporterebbe l'esecuzione di una dichiarazione SQL di aggiornamento ogni volta che un'immagine viene visualizzata. Utilizzando invece Redis, ci limitiamo ad incrementare un contatore memorizzato in memoria, ottenendo così prestazioni nettamente migliori.

## 5.2 Pillow e Sorl-Thumbnail

Ho utilizzato la libreria python Pillow per poter gestire le immagini: ogni profilo ha infatti la possibilità di caricare immagini, oltre che la possibilità di avere una foto profilo.

L'utilizzo di Sorl-Thumbnail, una applicazione di Django, serve invece per gestire le thumbnails.

Un thumbnail è una miniatura di un'immagine originale, ridimensionata e talvolta compressa per adattarsi a una dimensione più piccola e occupare meno spazio di archiviazione. Tipicamente, un thumbnail conserva l'aspetto proporzionale dell'immagine originale, ma è ottimizzato per una visualizzazione rapida e efficiente.

## 5.3 Requirements e Dipendenze

Per quanto riguarda l'utilizzo del progetto, fare riferimento al mio <https://github.com/SamueleCentanni/DjangoProject> ed in particolar modo leggere attentamente il file <https://github.com/SamueleCentanni/DjangoProject/blob/master/requirements.txt>