



GRASSERISATE

Progetto per tecnologie web

FABIO ZANICHELLI

Mat. 142111

Sommario

Traccia del progetto	2
Descrizione del progetto dell'applicazione	2
Recommendation system	6
Tecnologie usate	2
Test.....	6
Risultati	7
Problemi riscontrati e miglioramenti	9

Traccia del progetto

L'idea del progetto nasce da una personale idea di utilizzo dei social network, che a mio parere tendono sempre di più a far prevalere sentimenti negativi quali tristezza o invidia, anziché riuscire a strappare un sorriso. Da qui l'idea di realizzare un sito che abbia come obiettivo quello di far ridere gli utenti che lo utilizzano.

Il progetto consiste quindi nel realizzare un social di battute/barzellette con le seguenti caratteristiche:

- Un utente autenticato può pubblicare delle proprie battute e visionare quelle degli altri; un utente anonimo potrà guardare ma non pubblicare
- Ogni battuta viene categorizzata dall'utente che la pubblica (esempi di categorie sono satira, giochi di parole, barzellette...)
- Funzionalità di filtraggio delle battute secondo le categorie descritte nel punto precedente
- Possibilità di cercare profili di utenti registrati
- Possibilità per gli utenti autenticati di poter recensire le battute con una valutazione da 1 a 10
- Possibilità di seguire altri utenti
- Feed personalizzato in base agli utenti seguiti
- Sistema di suggerimenti basato su recensioni e persone seguite
- Possibilità, per un utente registrato, di inserire una propria foto profilo e altri dati a lui correlati
- Pagina admin

Tecnologie usate

Per la realizzazione del progetto sono state utilizzate i seguenti strumenti:

- **Django**, requisito fondamentale per il progetto, sia nel back-end che nel front-end
- **Pycharm Educational**, IDE per linguaggio python
- **Bootstrap**, per dei componenti utilizzati nel front-end
- **Sqlite** per la gestione del database in quanto base di dati di default per Django.
- **Git/GitHub** per il versioning del progetto.

Descrizione del progetto dell'applicazione

All'interno del progetto sono stati implementati i seguenti modelli:

- **Battute** per la memorizzazione delle battute ed altri dati come l'autore (FK di User), il tipo di battuta e la data e l'ora in cui è stata pubblicata
- **Recensioni**: questo modello contiene le recensioni degli utenti tramite i campi voto, utente (FK di User) e battuta (FK di Battute).
- **ProfiloDettagliato**: modello che salva i dati degli utenti che non sono nel modello User, come ad esempio la foto profilo. La relazione con User è 1:1, ovvero per ogni utente corrisponde uno ed un solo profilo dettagliato e viceversa. La PK è l'id dell'utente.
- **Followers**: modello che memorizza le relazioni di follow tra gli utenti. I suoi campi sono due chiavi straniere (entrambe di User) che corrispondono a colui che segue e a colui che è seguito. I campi seguatore e seguito, assieme, devono essere unici al fine di evitare errori quando si deve contare il numero di follower

In tutti i modelli, ad eccezione di ProfiloDettagliato, la chiave primaria è quella impostata in automatico da Django.

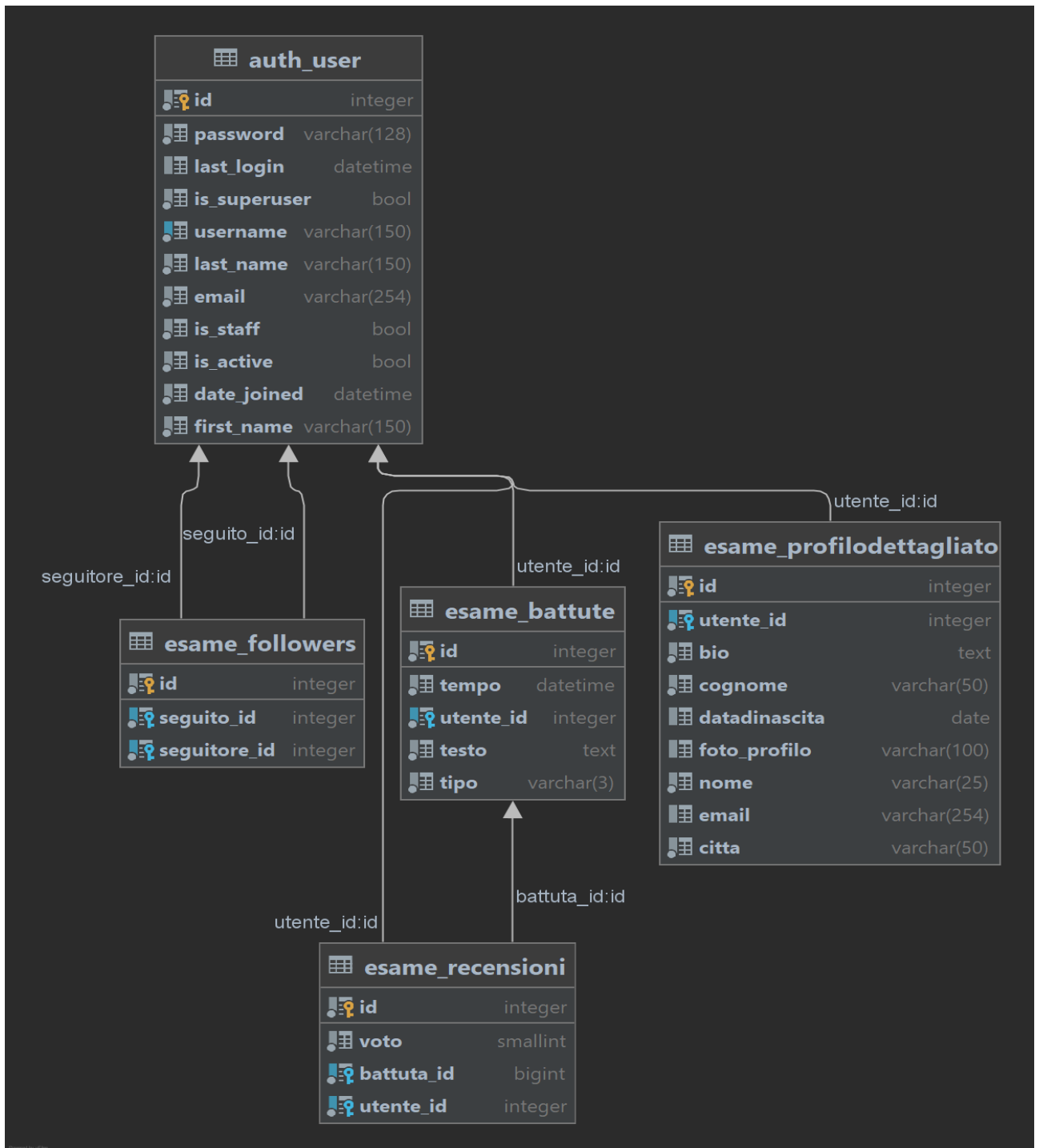


Figura 1: Diagramma dei modelli del database

All'interno del sito vengono distinti tre tipologie di utenti che presentano diversi privilegi:

- **Utente non autenticato:** l'utente in questione potrà esplorare il sito ma senza poter pubblicare nulla né recensire le battute altrui. Non può seguire gli altri utenti e, di conseguenza, non ha nemmeno accesso al feed. Può comunque cercare altri utenti e filtrare le battute.
- **Utente autenticato:** può pubblicare, seguire, recensire, cercare altri utenti, filtrare le battute per tipo, avere accesso al proprio feed e modificare il proprio profilo. Non può tuttavia cancellare le battute, le recensioni o modificare qualsiasi cosa riguardante gli altri profili.
- **Admin:** può visionare, cancellare e modificare profili, battute, recensioni e relazioni di follow. Per ovvie ragioni di sicurezza, non può vedere le password in quanto quest'ultime sono protette da hashing.

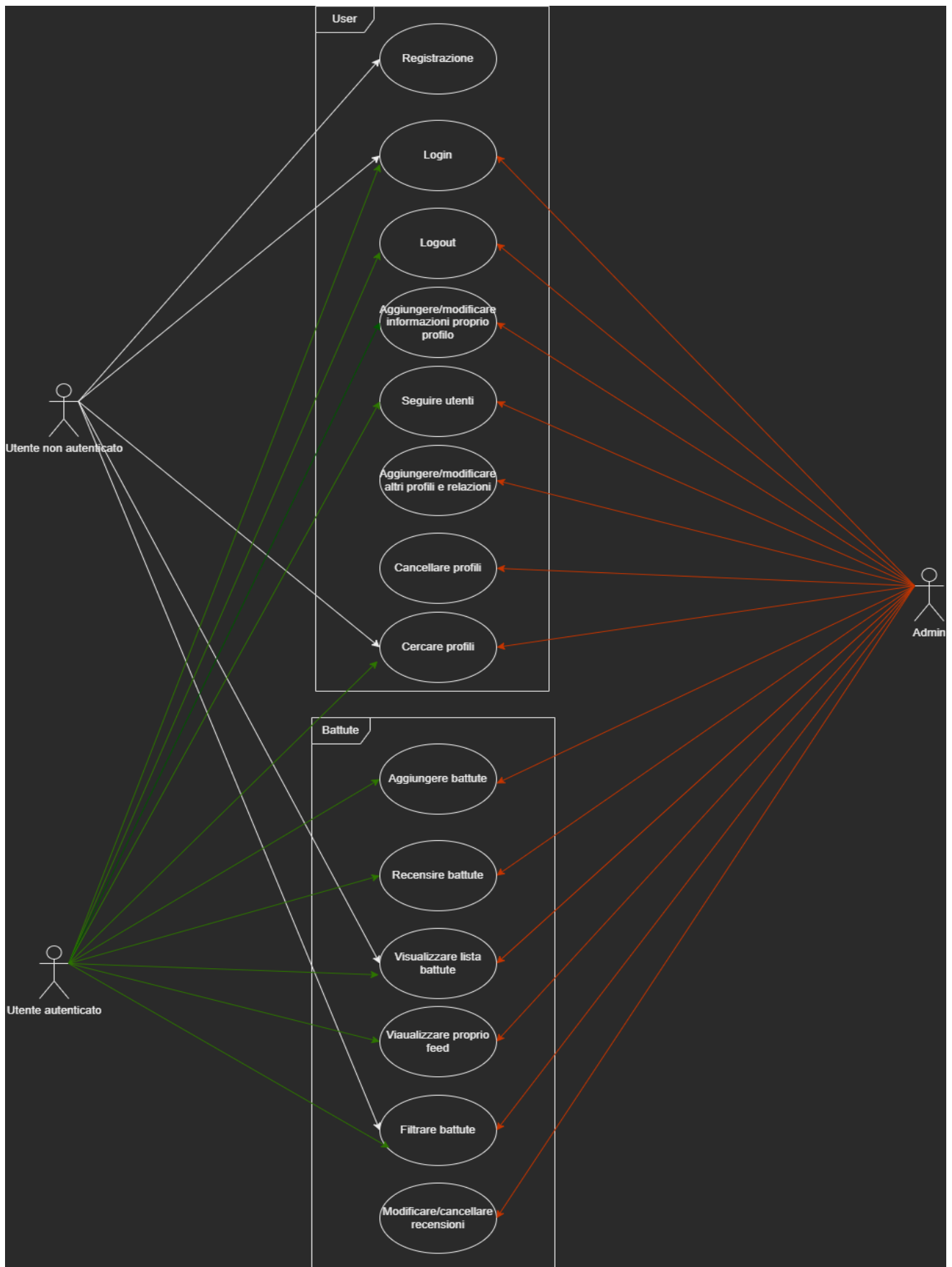


Figura 2: diagramma casi d'uso

Recommendation system

All'interno del progetto vi sono due sistemi di raccomandazioni, entrambi visionabili dal feed e quindi solo da utenti autenticati.

Il primo consiste nel creare una classifica tra gli utenti in base alla media di tutte le recensioni che questi hanno ricevuto, e permettere all'utente di poter visionare il podio in cima al feed. Questo tipo di suggerimento non è ovviamente personalizzato, e potrebbe succedere che quegli utenti siano già seguiti oppure che sia l'utente stesso a comparire nel podio. Rimane tuttavia uno strumento utile per chi si affaccia sul sito per la prima volta, per scoprire gli utenti più divertenti in assoluto.

Il secondo sistema è invece più personalizzato e si basa su chi non si conosce. Viene consigliato, infatti, il miglior utente che non si segue. Ovviamente, il sistema prevede la possibilità che sia l'utente stesso il migliore tra i non seguiti (è infatti vietata la relazione di "auto follow"); in tal caso, viene mostrato il secondo tra i non seguiti.

Test

All'interno della web-app sono presenti tre test:

1. **Test sulla view della home:** il test simula una richiesta GET al server e controlla se il codice di risposta è 200 e se il template utilizzato è quello corretto.
2. **Test di un profilo inesistente:** in questo test viene controllata la richiesta ad url che richiede la PK di un utente. Questo controllo, così come avviene per tutti gli altri, viene effettuato utilizzando un database differente creato appositamente per scopi di testing e pertanto inizialmente è vuoto. Ciò che ci si aspetta, quindi, è un codice di errore 404.
3. **Test sul modello Battute:** il test consiste nel creare un utente e una battuta ad esso associato. Utilizzando il metodo `__str__` del modello si può verificare che esso sia stato salvato correttamente.

Risultati

The screenshot shows the GRASSERISATE homepage for non-authenticated users. At the top, there is a dark header with the GRASSERISATE logo. Below the header, the main heading reads "Benvenuto* su GRASSERISATE !". Underneath, a subheading says "Ecco le ultime battute pubblicate dagli utenti!". There are two buttons: "Filtra" and "Cerca profilo". To the right, there are links for "Non hai ancora un account? Registrati" and "Hai già un account? Login". The main content area displays three jokes, each with a user link, the joke text, a red warning message, and the average rating.

GRASSERISATE

Benvenuto* su GRASSERISATE !

Ecco le ultime battute pubblicate dagli utenti!

[Filtra](#) [Cerca profilo](#)

[Non hai ancora un account? Registrati](#) [Hai già un account? Login](#)

[Flem](#)

Battuta squallida
C'è una mela ed una arancia che giocano a carte, la mela dice all'arancia: "Forza, pesca!"
Non puoi recensire perchè non hai effettuato il login
Media recensioni: 3,5

[ilcastor](#)

Notizia divertente
Chiude una lavanderia. Faceva affari sporchi.
Non puoi recensire perchè non hai effettuato il login
Media recensioni: 7,5

[ilcastor](#)

Figura 3: home del sito per utenti non autenticati

The screenshot shows the GRASSERISATE homepage for authenticated users. At the top, there is a dark header with the GRASSERISATE logo. Below the header, the main heading reads "Benvenuto* su GRASSERISATE , Fabio!". Underneath, a subheading says "Ecco le ultime battute pubblicate dagli utenti!". There are several buttons: "Il tuo feed", "Aggiungi una battuta!", "Il tuo profilo", "Filtra", "Cerca profilo", and "Logout". The main content area displays three jokes, each with a user link, the joke text, a "Recensisci!" button, and the average rating.

GRASSERISATE

Benvenuto* su GRASSERISATE , Fabio!

Ecco le ultime battute pubblicate dagli utenti!

[Il tuo feed](#) [Aggiungi una battuta!](#) [Il tuo profilo](#) [Filtra](#) [Cerca profilo](#) [Logout](#)

[Flem](#)

Battuta squallida
C'è una mela ed una arancia che giocano a carte, la mela dice all'arancia: "Forza, pesca!"
[Recensisci!](#) Media recensioni: 3,5

[ilcastor](#)

Notizia divertente
Chiude una lavanderia. Faceva affari sporchi.
[Recensisci!](#) Media recensioni: 7,5

[ilcastor](#)

Notizia divertente

Figura 4: schermata home per utenti autenticati

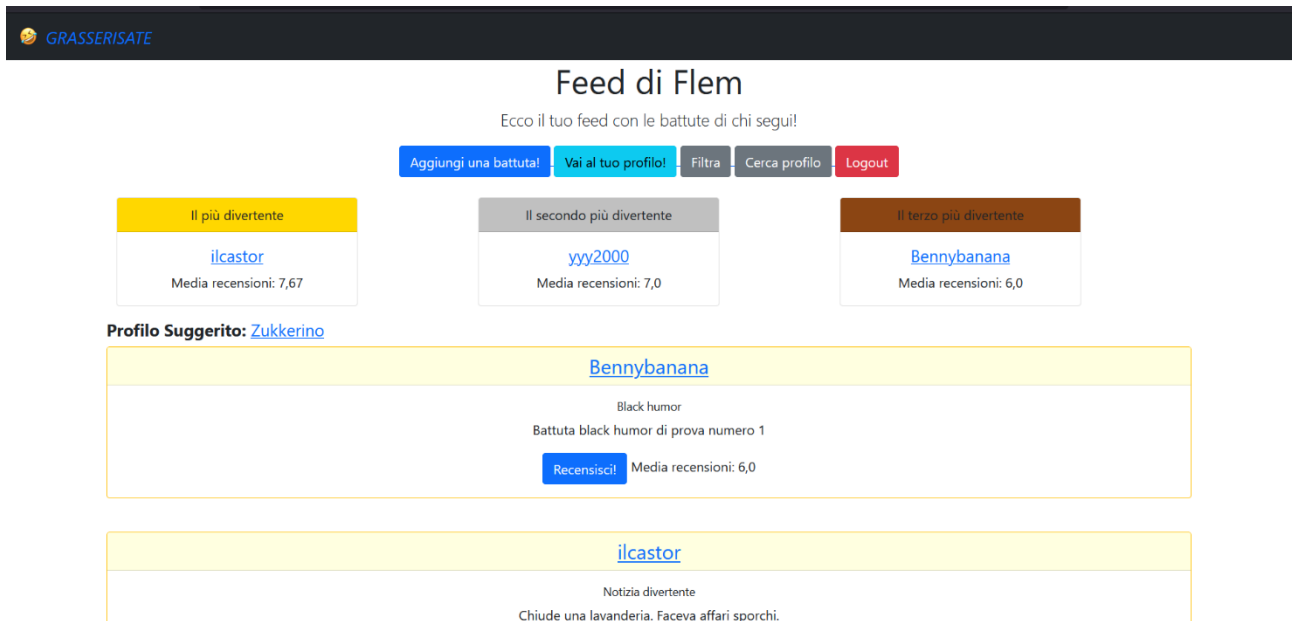


Figura 4: schermata feed

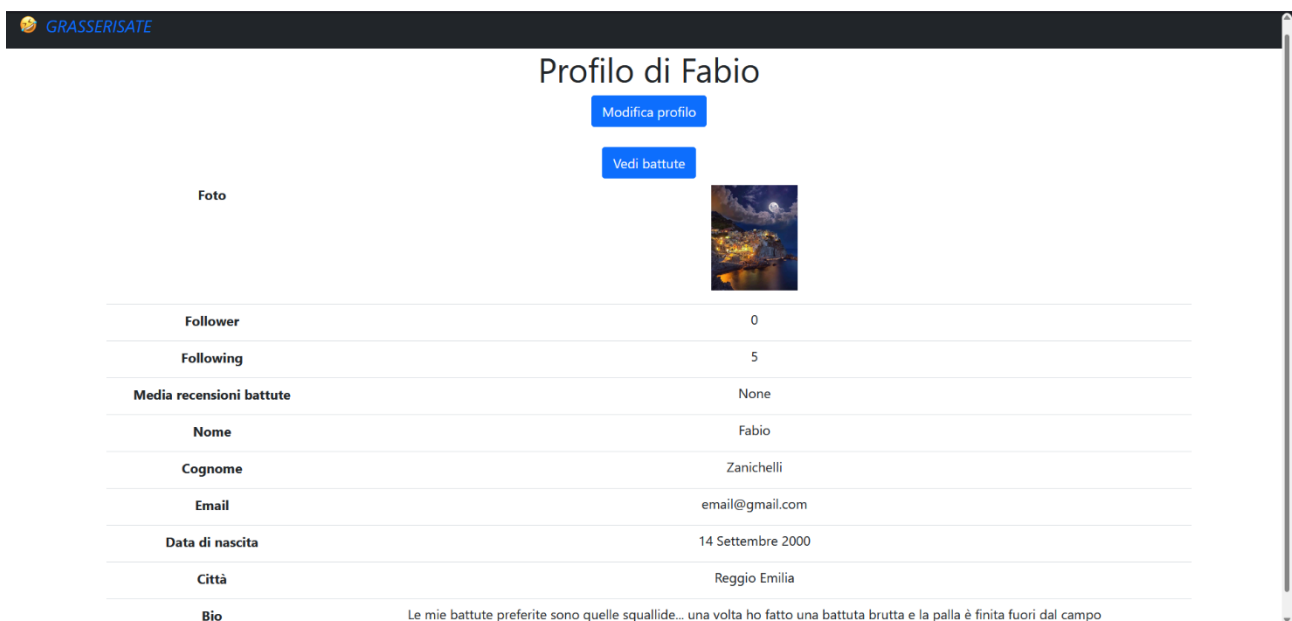


Figura 5: schermata di visualizzazione profilo

Problemi riscontrati e miglioramenti

I principali problemi sono stati riscontrati per quanto riguarda il calcolo della media delle recensioni dei profili e in generale il model `ProfiloDettagliato`.

Questo problema era dovuto al fatto che la tupla non venisse aggiunta automaticamente alla registrazione di un utente, nonostante l'utilizzo del tipo `AutoOneToOneField`. Il problema è stato successivamente risolto con una funzione implementata nel modello di `ProfiloDettagliato`.

Alcune idee di miglioramenti sono le seguenti:

- Possibilità di segnalazione delle battute; l'admin potrà vedere le segnalazioni ed eventualmente eliminare le battute in questione
- Possibilità di caricare immagini (meme) oltre le battute testuali
- Pubblicare la "battuta del giorno" nel feed, ovvero la battuta che nelle ultime 24 ore ha riscosso maggior successo con le recensioni
- Miglioramento del recommendation system, in modo da tenere in considerazione anche il tipo di battute che piace di più all'utente (es. a `utente1` piacciono i giochi di parole, `utente2` ne carica tanti, allora ad `utente1` consiglio `utente2`)