

Documentazione Progetto Basi di Dati Mod. 2

Valutazione di Progetti di Ricerca

Leonardo Mirza, Lodovico Karroca, Alberto Cestaro

1. Introduzione

Questo progetto consiste in una web application per la valutazione interna dei progetti di ricerca da sottoporre per finanziamento all'Unione Europea. La web app si interfaccia con un database relazionale.

Il backend del progetto è stato sviluppato in Python, utilizzando le librerie Flask e Flask-SQLAlchemy. Il DBMS utilizzato è [SQLite](#).

Il frontend è stato realizzato con HTML e CSS, utilizzando Bootstrap come Framework.

2. Funzionalità principali

Ci sono due tipi di utenti che si interfacciano alla Web App: i ricercatori e i valutatori. L'utente si registra sulla piattaforma tramite una schermata di registrazione, dove indica se intende creare un account come ricercatore o come valutatore.

I ricercatori possono creare, visualizzare, modificare e cancellare dei progetti di ricerca.

I progetti vengono visualizzati nella Homepage del ricercatore in una tabella e hanno 5 stati possibili:

- Bozza
- Pubblicato in attesa di valutazione
- Richiesta di modifiche
- Approvato
- Rifiutato

Ogni progetto consiste in una descrizione testuale e un insieme di documenti (files) di diverso tipo, che saranno sottoposti per la valutazione.

I valutatori accedono alla piattaforma e visualizzano i progetti nella loro Homepage (tranne quelli in bozza, che sono visibili solo ai ricercatori).

La finestra di tempo per la valutazione dei progetti è di un mese dalla data di pubblicazione.

I documenti dei progetti in attesa di valutazione (pending) possono essere visualizzati e valutati dai ricercatori, creando un report di valutazione per ciascuno di essi.

Il report consiste in un voto in 30esimi e una descrizione testuale.

Una volta creato il report, il valutatore può decidere se usarlo per richiedere delle modifiche al ricercatore o se valutare definitivamente il progetto.

La navigazione all'interno del sito avviene tramite un menu posizionato nella parte alta della pagina.

Se l'utente ha effettuato l'accesso vengono visualizzati i tasti "Home" e "Logout".

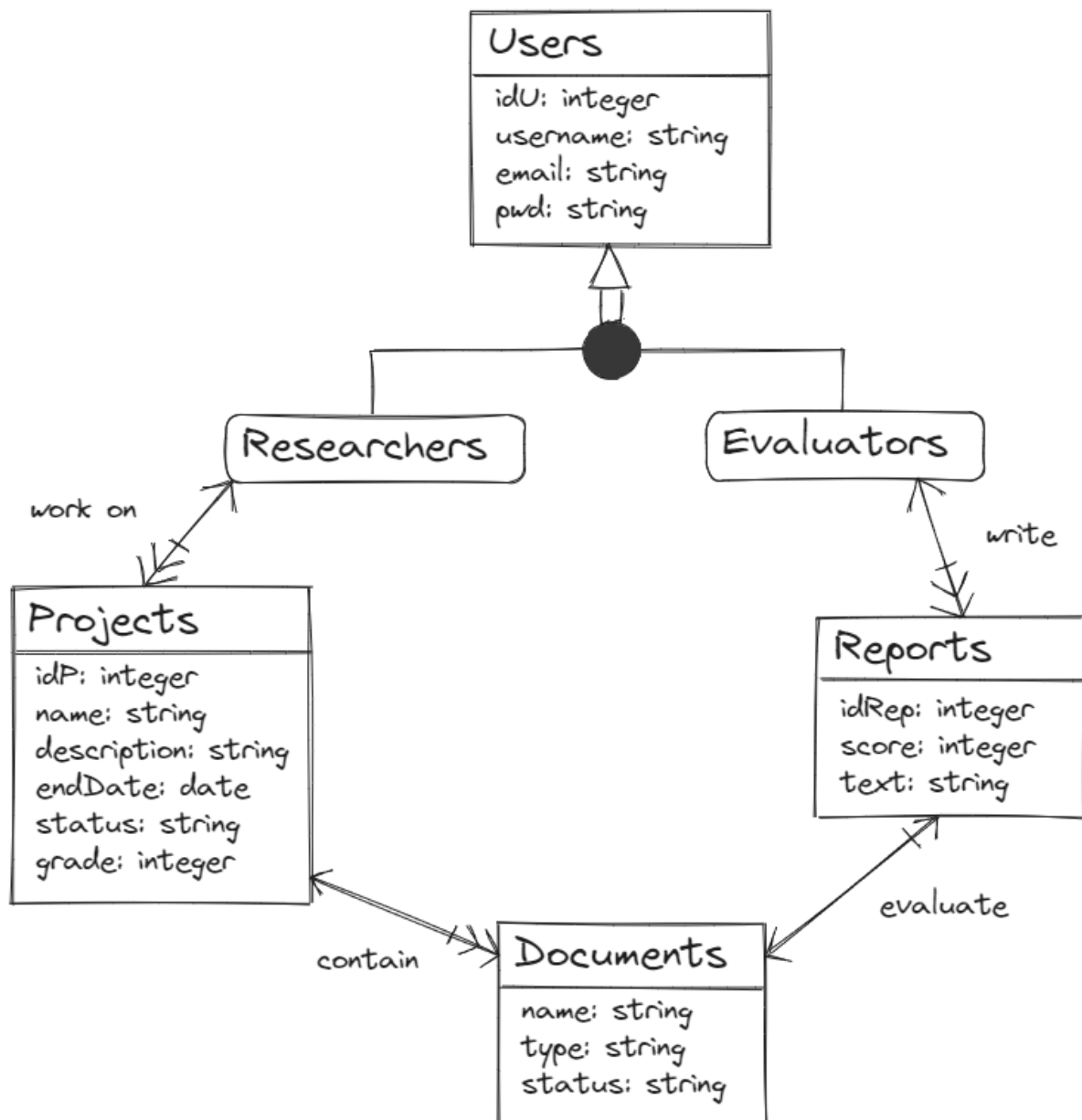
Se invece l'utente deve ancora effettuare l'accesso i tasti saranno "Login" e "Register".

3. Progettazione concettuale e logica

Abbiamo individuato 6 entità:

- Utente
 - Ricercatore
 - Valutatore
- Progetto
- Documento
- Report

Lo schema concettuale / a oggetti:



Sono state fatte le seguenti assunzioni:

Un utente può essere un ricercatore o un valutatore.

I ricercatori possono lavorare su più progetti, e un progetto appartiene ad un solo ricercatore.
Un ricercatore potrebbe non aver ancora lavorato a nessun progetto.

I progetti possono contenere più documenti, e un documento è correlato necessariamente ad un progetto.

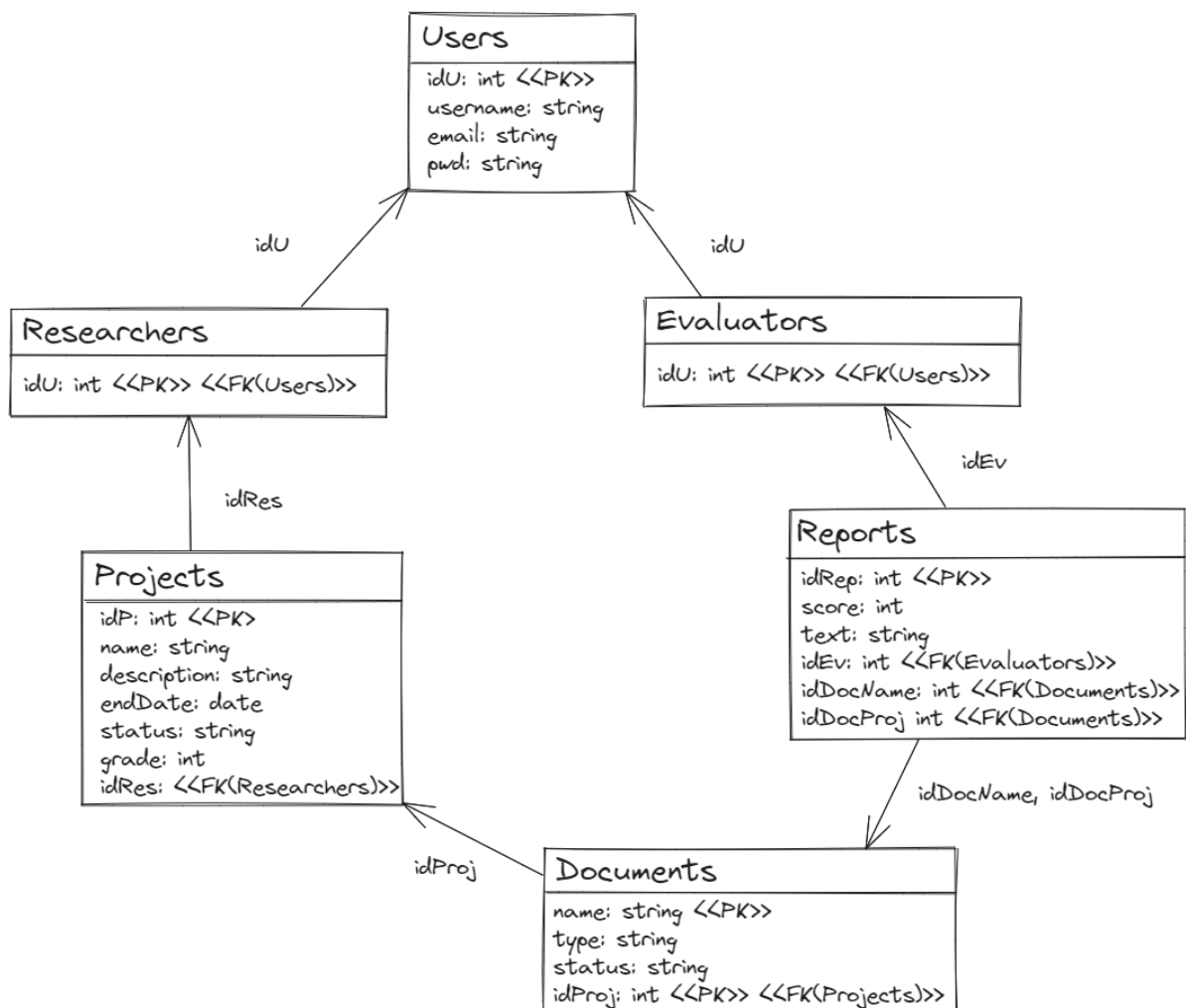
Un progetto potrebbe non avere documenti (in caso venissero cancellati per aggiungerne altri in seguito).

Un valutatore può scrivere più report, e un report è scritto da un solo valutatore.

Un valutatore potrebbe non aver scritto ancora nessun report.

Un report valuta un documento, ma un documento potrebbe non essere ancora stato valutato.

Lo schema relazionale:



4. Query principali

Un esempio di query importante è quella che controlla se l'utente è presente nel DB al momento del login:

```
user = User.query.filter_by(email=email_form).first()
```

Che si traduce in linguaggio SQL nel seguente modo:

```
SELECT *  
FROM User u  
WHERE u.email = email_form
```

Questa query invece trova tutte le coppie di report e documento che appartengono allo stesso progetto:

```
q = db.session.query(Report, Document).join(Report, Report.idDocProj  
== Document.idProj).filter(Document.idProj == p_id).all()
```

In SQL:

```
SELECT *  
FROM Report r JOIN Document d ON r.idDocProj = d.idProj  
WHERE d.idProj == p_id
```

5. Principali scelte progettuali

I controlli per le SQL-Injection vengono effettuati autonomamente dal DBMS SQLite

E' stato implementato (tramite una annotazione Python) un controllo per evitare che l'utente possa accedere ad altre route del sito cambiando l'url della barra di ricerca.

Prima di essere salvate nel DB, le password vengono criptate con il metodo di hashing "Scrypt"

6. Ulteriori informazioni

Strumenti utilizzati per la realizzazione del progetto (sviluppo, comunicazione e ricerca):

- Software:
 - Visual Studio Code
 - PyCharm
 - Discord
 - Git
- Websites:
 - [Trello](#)
 - WhatsApp
 - [Google Fonts](#)
 - GitHub
 - [Bootstrap](#)
 - Google
 - Youtube
 - [Stack Overflow](#)

7. Contributo al progetto

La maggior parte del lavoro è stata effettuata da remoto, ragionando e scrivendo il codice assieme.

Per comunicare abbiamo usato [Discord](#) e WhatsApp.

Per gestire le task abbiamo utilizzato [Trello](#), un sito dove è possibile creare delle liste di attività visibili e modificabili dai membri del progetto.

Leonardo Mirza: sviluppo (con particolare attenzione al frontend)

Lodovico Karroca: sviluppo (con particolare attenzione al backend)

Alberto Cestaro: testing e sviluppo