

## Progetto: University

### Descrizione del progetto

University nasce con l'idea di creare un sistema che emuli la vita universitaria e che sia quindi in grado di riprodurre gli elementi caratteristici.

### Obiettivi

Gli obiettivi generali del sistema sono:

- Possibilità di iscrizione per i nuovi studenti
- Creazione di gruppi di lavoro per gli studenti
- Caricamento di un file di qualsiasi formato (per gruppo di lavoro)
- Possibilità di effettuare una discussione con il docente del gruppo

### Analisi dei requisiti

Dati gli obiettivi sopra esposti, l'analisi dei requisiti produce una lista di funzionalità necessarie e utili per lo sviluppo del sistema.

Tra i requisiti principali del sistema, vi sono la possibilità di effettuare la registrazione di nuovi studenti e successiva login per accesso ai sistemi e ai dati personali. La presenza di questi due requisiti impone: l'esposizione di due servizi, uno per l'accesso e uno per la registrazione e la presenza di una tabella degli utenti nel DB.

L'utente in fase di registrazione deve indicare il proprio username e la propria password per i successivi accessi.

La gestione delle autenticazioni e autorizzazioni richiede che l'accesso ad ogni servizio deve essere autorizzato, ossia in ogni richiesta deve essere specificata una chiave. Per accedere a tutti i servizi, al di fuori dei servizi di login e registrazione dove è richiesta solo l'autenticazione, è necessaria anche l'autorizzazione: la chiave di accesso, infatti, permetterà anche di individuare l'utente che ha effettuato la richiesta.

La possibilità di creare gruppi di lavoro e di caricare dei file impone la realizzazione di un servizio e la presenza di diverse tabelle per relazionare i gruppi e gli utenti.

È necessario, infine, un ultimo servizio che permetta ai diversi gruppi di effettuare una discussione con il docente del gruppo.

### Funzionalità

Sono presenti tre ruoli: guest (ospite), studente e professore.

Le funzionalità del guest sono:

- Login: si divide in login studenti e login professori
- Register: registrazione per gli studenti, i professori vengono inseriti manualmente sul database dall'amministratore
- Forgot: ricezione e-mail con una nuova password

Uno studente può:

- Visualizzare:
  - Profilo utente
  - Tutti gli studenti
  - Tutti i gruppi
  - File di un gruppo (tramite id gruppo/file)
  - Tutti i professori
  - Tutte le discussioni di un professore (tramite id professore)
  - Una discussione (tramite id discussione/gruppo)
  - Una prenotazione (tramite id prenotazione/gruppo)
- Modificare la password
- Creare un gruppo (diventandone l'admin)
- Unirsi ad uno o più gruppi (diventandone membro)

Un admin può:

- Modificare il nome del gruppo
- Eliminare il gruppo
- Eliminare un membro del gruppo
- Eliminare il file del gruppo
- Caricare il file del gruppo
- Creare una prenotazione
- Eliminare una prenotazione
- Modificare una prenotazione

Un fellow può:

- Lasciare il gruppo

Un professore può:

- Visualizzare:
  - Profilo utente
  - Tutti gli studenti
  - Tutti i gruppi
  - File di un gruppo (tramite id gruppo/file)
  - Tutti i professori
  - Tutte le discussioni di un professore (tramite id professore)
  - Una discussione (tramite id discussione/gruppo)
  - Una prenotazione (tramite id prenotazione/gruppo)
- Modificare la password
- Creare una discussione

- Eliminare una discussione
- Modificare una discussione
- Eliminare una prenotazione

### **Versionamento del codice e repository**

Il versionamento del codice sorgente è gestito attraverso il software di controllo di versione GIT e il codice è salvato nello spazio offerto da GitHub.

La repository è raggiungibile tramite il seguente link: <https://github.com/mmedici0/University>

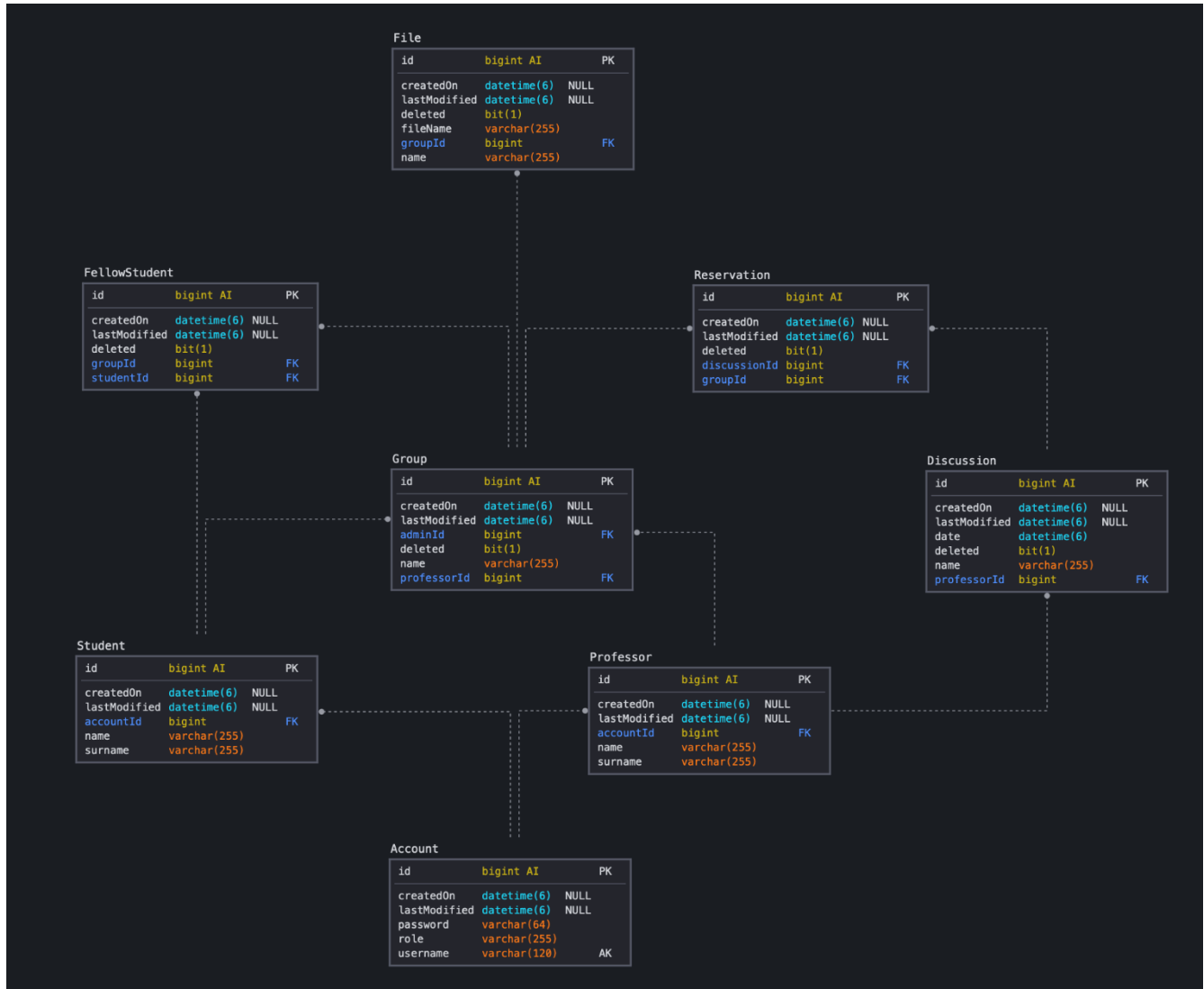
### **Tecnologie utilizzate**

Il linguaggio di programmazione scelto è Java 11, con il framework Spring Boot.

I servizi e i dati esposti sfruttano l'interfaccia di programmazione delle applicazioni REST API, con lo scambio di dati in formato JSON.

Viene inoltre impiegato MySQL 8 come servizio di database e Swagger per la documentazione.

## Database

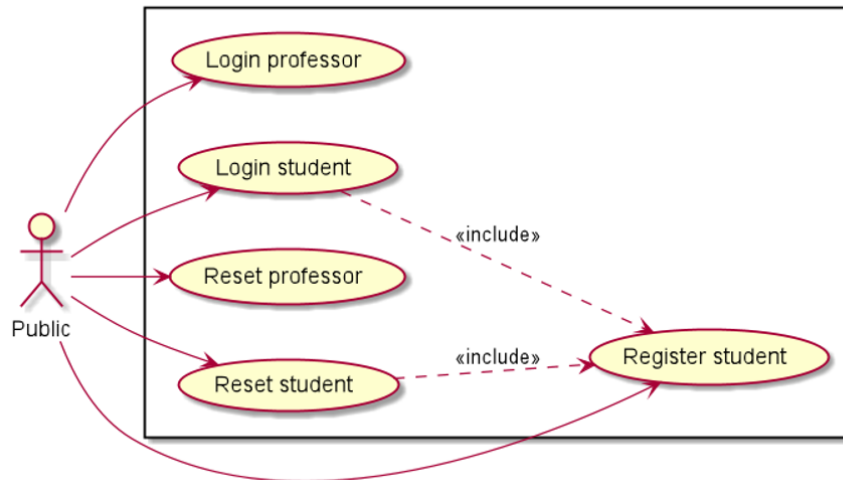


- Account: raccoglie le informazioni di ogni utente, in particolare le credenziali e il ruolo
- Student: raccoglie tutte le informazioni di ogni studente
- Professor: raccoglie tutte le informazioni di ogni professore
- Group: raccoglie le informazioni di ogni gruppo, tra cui l'id dell'admin e l'id del professore
- FellowStudent: raccoglie le informazioni e relaziona i membri con i gruppi di appartenenza
- File: raccoglie tutti i file
- Discussion: raccoglie tutte le discussioni create dal professore
- Reservation: raccoglie tutte le prenotazioni alle discussioni

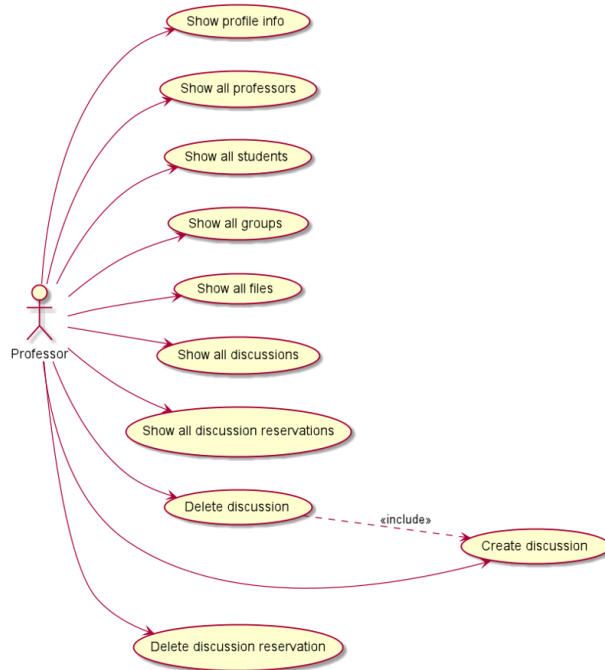
## Use case Diagram

Vengono descritti i seguenti casi d'uso:

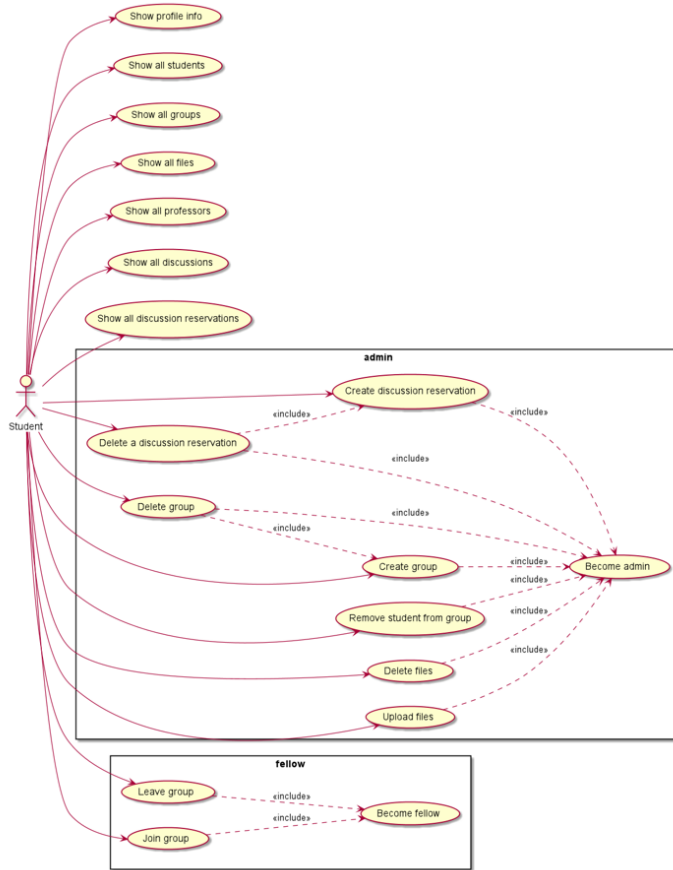
- funzionalità public



- funzionalità professore



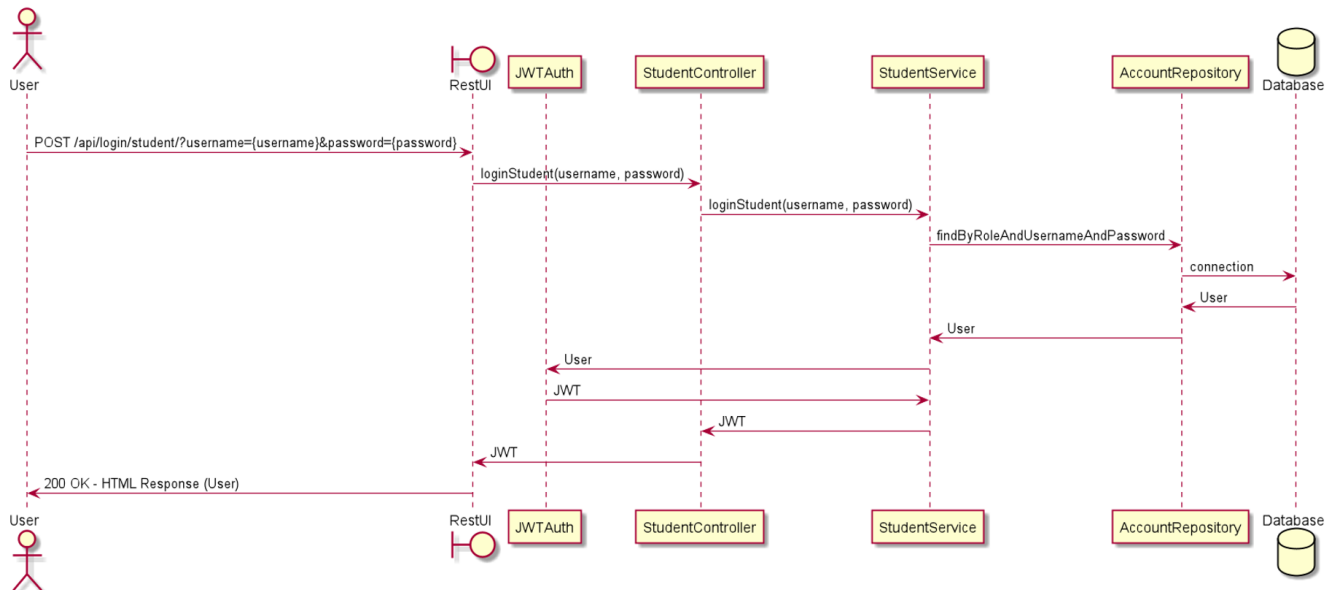
- funzionalità studente



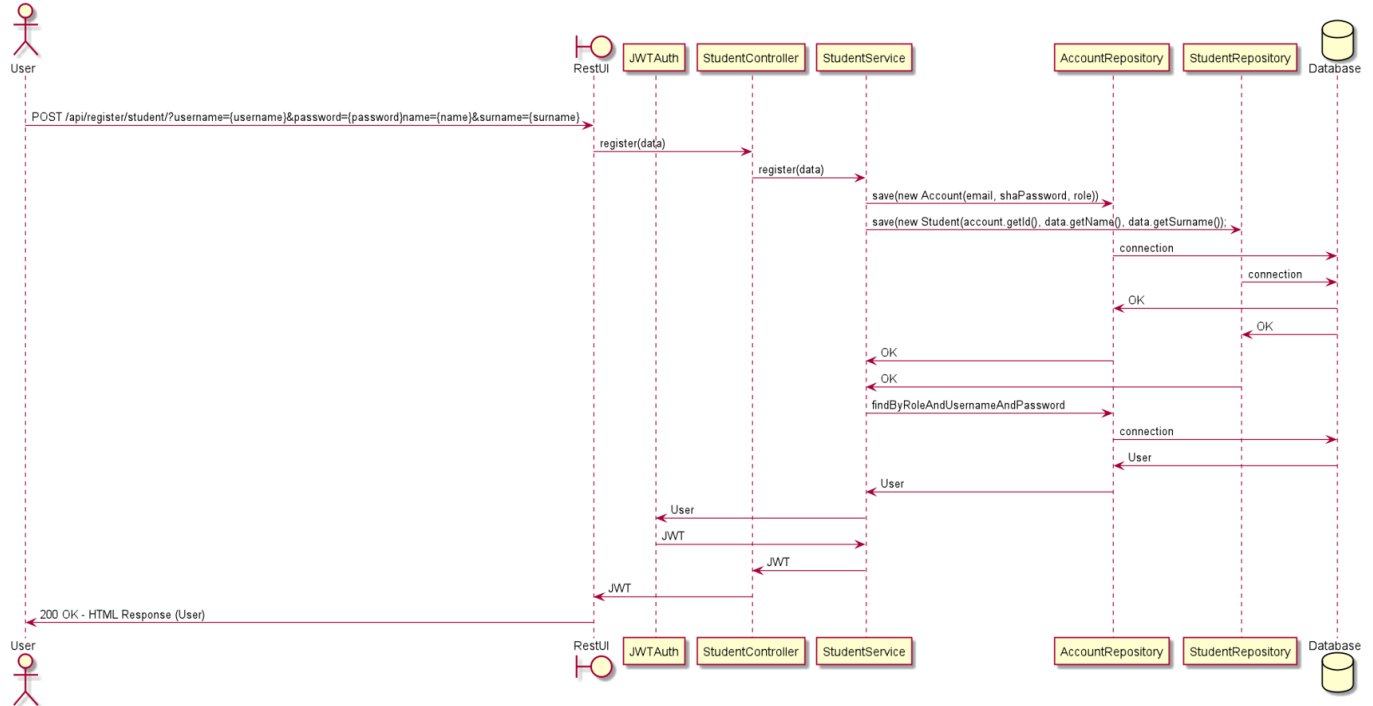
## Sequence Diagram

Vengono descritti i seguenti sequence diagrams:

- login



- registrazione



## Piano di test

Per strutturare i test vengono utilizzati i controller e i service.

I controller sono test che vengono utilizzati per verificare che l'api sia funzionante ed effettuano questa prova utilizzando un mock. I test di servizio servono invece a testare le operazioni svolte, ad esempio: chiamo la register e verifico che sul database ci sia la nuova utenza.

I test per i controller utilizzati sono:

- ProfessorApiControllerTest
- PublicApiControllerTest
- StudentApiControllerTest

I test per i servizi utilizzati sono:

- ProfessorServiceTest
- PublicServiceTest
- StudentServiceTest

## Rapporto sull'andamento dei test

I risultati del test sono visibili al seguente [link](#).

## Future migliorie

Una possibile miglioria per un perfezionamento dell'esperienza degli utenti è l'implementazione di un servizio di chat tra gli utenti.