# Progetto Ingegneria del Software 2023/2024

#### Obiettivo:

Sviluppo di un sistema di login e di profilazione utente lato frontend e backend.

### Distribuzione dei ruoli

- Lavoro svolto individualmente;
- Sviluppo completo lato frontend e backend;
- Mi sono anche occupato della fase di testing.

#### Evidenza delle cerimonie

- Lavorando da solo non ho avuto modo di avere grandi confronti se non con me stesso;
- Ogni mattina ho fatto quindi un briefing sulle cose da fare.
- Per fare ciò ho usato Hypernotes, un'alternativa a notion.
  Progetto Ingegnetia SW

Progetto Ingegneria SW

1. Organizzazione Task

01/09

• Obertitet

[// Implementare login backend (Spring Security )

[// Implementare login backend (Spring Security )

[// Implementare logica di registrazione (endipoint API per registrazione)

[// Initializzare il database (configuration de Mr/SQL)

Prifessioni del giormo.

Il backend è stato configuration de Mr/SQL)

Prifessioni del giorno.

Il backend è stato configuratione. Il database è stato correttamente inizializzato.

02/09

Obertivit.

[// Iniziare login frontend (Wue je)

DC (Tester es el login funciona correttamente con il backend

[// Creare som di registrazione sul frontend

Rifessioni del giorno.

Il login fontend è stato avvisito, ma c'è un problema nella comunicazione tra frontend e backend. La form di registrazione e funcionante.

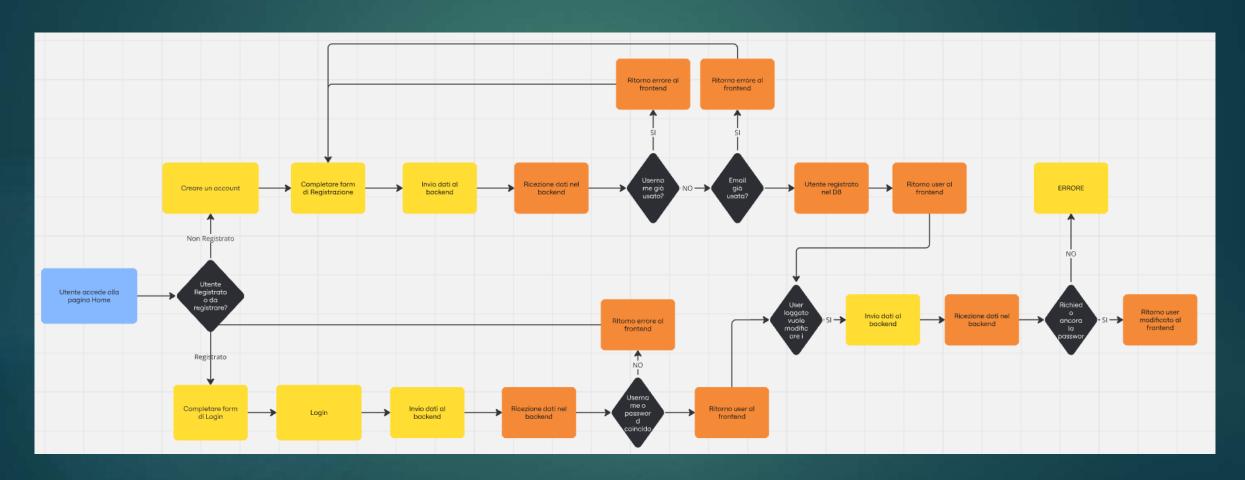
03/09

Obertivit

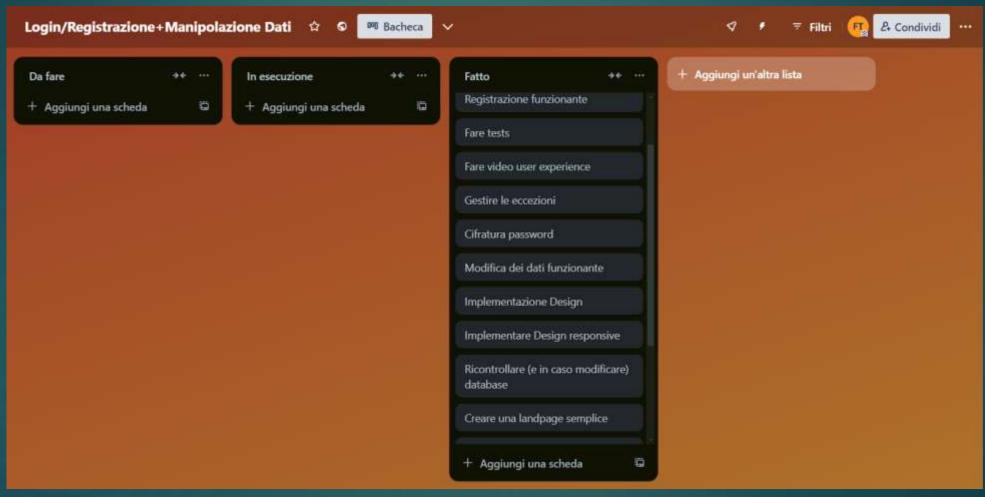
[// Risolvere problemi di autenticazione tra frontend e backend

[// Implementare validazione delle oredenziali sul frontend

### Estrazione disegno funzionale



### Screenshot di Trello



Tonin Francesco

#### Test Cases

- Sono stati fatti diversi test cases, ad esempio:
  - ► Login con credenziali errate;
  - Registrazione account utente;
  - ▶ Logout;
  - Cambiare sia username sia la password mettendo un username già in uso;
  - ▶ Inserire una mail già usata in fase di registrazione;
  - Provare ad accedere a pagine senza autorizzazione necessaria.

### Disegno architetturale scelto

- ▶ Ho preferito usare una struttura separata:
  - ▶ Per il frontend:Vuejs;
  - ▶ Per il backend: Springboot.
- ▶ Ho scelto questo approccio perché:
  - ▶ Più flessibile e scalabile rispetto ad una architettura monolitica;
  - ▶ Trovo giusto separare backend da frontend per una questione di ordine e manutenzione.

## Funzionalità implementata

- L'implementazione più complicata è stata quella della modifica dei dati dell'utente.
- ▶ Infatti l'utente manda i dati all'endpoint /api/user-update.
- ► Il backend dopo aver effettuato vari controlli (se username è già usato ad esempio), modifica i dati nel database e ritorna al frontend l'utente modificato.

### Feedback dell'esperienza

- Sicuramente questo progetto mi ha insegnato l'importanza di fare una documentazione del codice massiccia, sia per facilitare la manutenzione sia per la fase di testing.
- Inoltre ho imparato «sul campo» che il testing è fondamentale e che si deve prima progettare di lanciarsi a capofitto sul codice.
- Penso anche che mi abbia insegnato tanto anche sull'importanza di github e di avere sempre un prototipo funzionante in remoto.