

Sprint 1 – Report

Reseller

Introduction:

Reseller is an application for selling used clothes with an Instagram type interface, each user having it's own profile with their items for sale and a system of followers. Our motivation is striving to offer a more sustainable alternative to staying trendy, encouraging people to buy fewer new clothes.

The technologies used are Spring Boot (Backend), Angular (Frontend), MySQL (Database), Firebase (Storage), Python (Scripting).

We started with the login process since it is one of the most important features of an account-based application.

Backlog:

- Register
- Register page
- Login
- Login page
- Secure login
- Database creation
- Testing database and forms
- Populate database

Tasks Report:

- Register - **DONE**

Register functionality in backend is done with JPA Repository and Hibernate. Each table has it's own class in the Model folder and describes an object and its

properties. For security reasons, every time an account is created an email is sent to the user's address for him to confirm and enable his account. We achieved that with a unique token. For the same reasons, we also crypted the passwords in the database.

The user class is supposed to only be used for authentication and for functionality inside the app, we created a Profile class.

- Register page - **DONE**

Created a form with validators which sends the data to the backend using CRUD operations.

- Login – **NEXT SPRINT**

There were a few problems and bugs with the login functionality when we first started. The spring security package had a recent update and we struggled to find resources. So we moved it to the next sprint.

- Login page – **NEXT SPRINT**

Login form using validators and CRUD operations. Here we struggled with parsing the JWT Token.

- Secure login – **NEXT SPRINT**

Same with the login task.

- Testing database and forms – **NEXT SPRINT**

We couldn't test the login so we moved it to the next sprint. Register working fine.

- Populate database – **DONE**

Added users to the database.