Internal System for Lending Hardware - Objects (ISLHO)

Degree programme : BSc in Computer Science | Specialisation : Digital Business Systems
Thesis advisor : Prof. Rolf Jufer
Expert : Dr. Mirjam Hofer

The BFH has numerous objects which are used for the Learning Process. These objects are often used for a short period of time during the semester and the rest of the time they will be stored untouched in a closet, archive etc. The goal of this project is to make it possible for BFH students and its employees to borrow these objects and conduct their own experiments with them.

Project Overview

The BFH have acquired an extensive arsenal of objects that are used for learning purposes, such as the robots for the one-week block. Students were given one robot for each group, which they must program within a week. These and other objects are often only used for a short period of time during the semester and the rest of the time they are left unused in the closet, archives, etc.
From here on came the idea of creating an internal web application that helps BFH students and employees to gain access of these objects and thereby draw the maximum benefit from them. The main goal of this project was to build a web application that gives an overview of the available objects. Objects will be displayed on the platform with a short description, object quantity and a pickup location.
The interested students and employees can make a reservation for one or more of these objects by choosing them and specifying both a pickup and return date. The project was done in multiple steps (Iterations), following the scrum methodology for project management.

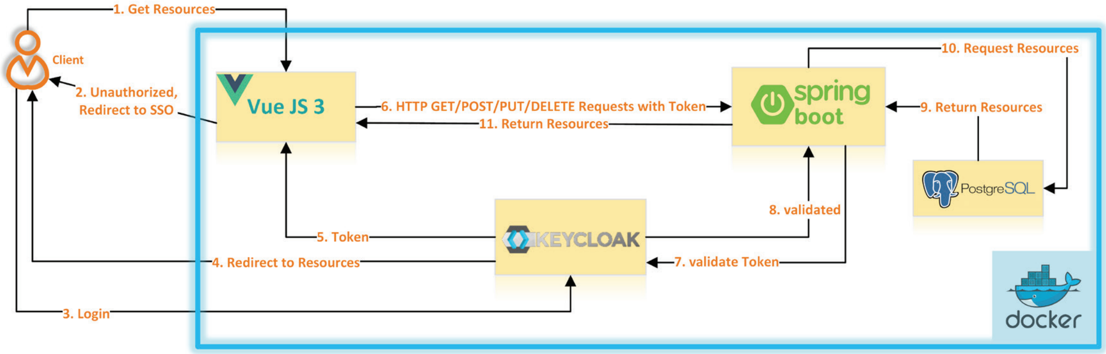
The Application

The application consists of both a server (backend) and client side (frontend). The frontend was developed using the Framework Vue JS and is made up of multiple components and routes, alongside the services which are responsible for sending HTTP requests to the backend.
Spring Boot was used to develop the backend and is based on a three-tier architecture in which the codebase is divided into three separate layers with differentiate responsibilities.
The Controller Layer is responsible for communication with the frontend, and the business logic is done in the Service Layer as well as the Repository Layer, which is responsible for the interaction with the database to save and restore application data.
Keycloak works as an identity provider and is responsible for authorization and authentication.

Conclusion

The main Goals of this Project were met, and some of the optional Objectives were also Implemented. However, there are certainly many other optional features that could be implemented in the future.

Mohammed Ali
078 210 02 40
mohammed-h.ali@hotmail.com



Project Dynamic Model