31.06.2024

A logo of a building

Description automatically generated

El-Sayed Aimen, Markus Mayer, Aziz Ftaiti  
Matthias Werkl, Eler Wohlmuth

My Flat-Platform

Back-End

Table of Contents

[Description 2](#_Toc170748022)

[Tools & Technologies Used 3](#_Toc170748023)

[Description of Web Service APIs & Application 4](#_Toc170748024)

[Code Quality Assurance 4](#_Toc170748025)

# Description

**Project Summary**

This Software Requirements Specification (SRS) document outlines the requirements for the My Flat project, a web-based platform aimed at simplifying and modernizing communication and administration for multiple rental properties within a building. This document covers the initial release of My Flat, with the scope including all necessary features to facilitate improved communication pathways among tenants and with property management, to reduce administrative overhead related to general property management tasks, and to ensure proper document management pertaining to rental properties. The SRS will detail the Greenfield development of this platform, which is to be co-designed by students with potential for further development into a master's thesis project. It is intended that the platform will be accessible both as a web application, with a design focus on ease of use, intuitive interaction, and compliance with GDPR through anonymization features where required.

# Tools & Technologies Used

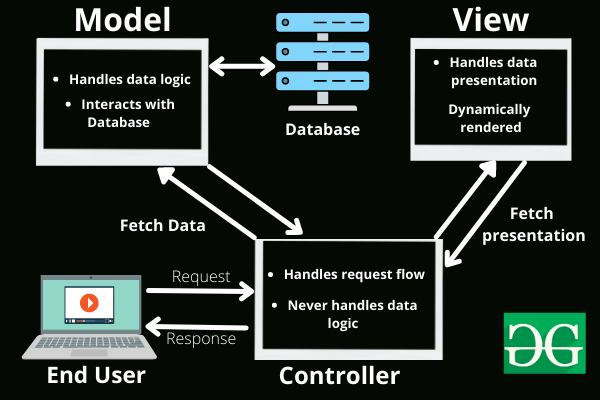
* **Frontend**: React.js, JavaScript
* **Backend**: Spring Boot, Java, Maven, Lombok
* **Database**: MySQL, phpMyAdmin, Hostinger as Host
* **Security**: JWT for authentication, HTTPS for secure communication
* **Document:**  Postman
* **Test:** Mockito, Postman
* **Clean code:** Snorlint
* **IDE:** IntelliJ, GitHub Desktop, Docker
* **Others**: CORS Handling, REST APIs, Git pilot, ChatGPT

# REST APIHostinger - WikipediaAdminer vs phpMyAdmin: Know the Key Differences!Announces Postman Flows ...Git – WikipediaThe Build Tool: Maven. Hey, | by Sarvar ...Spring boot is an app development ...MariaDB full logo transparent PNG - StickPNGSonarLint | excentiaProject Outline: My-Flat Application

# Design of data model

## MVC

## Model-View-Controller is a software architectural pattern that separates an application into three main logical components: the Model, responsible for managing data and business logic; the View, responsible for user interface and presentation logic; and the Controller, responsible for handling user input and updating the Model and View accordingly. This separation enhances code maintainability, scalability, and reusability.



# Back-End

1. Maven Dependency
2. Entities
3. Dtos
4. Repositories
5. Services
6. Controllers
7. Security

Description of Web Service APIs & Application

* **AuthController.java**: Manages user authentication (login, registration).
* **PropertyManagementController.java**: Handles property management operations.
* **TenantController.java**: Manages tenant interactions and data access.
* **PropertyManagementCommunicationController.java**: Facilitates communication between property managers and tenants.
* **JwtAuthenticationFilter.java**: Validates JWT tokens.
* **SimpleCorsFilter.java**: Manages CORS settings.

# Code Quality Assurance

* **Code Reviews**: Regular peer reviews for ensuring best practices.
* **Unit Testing**: Comprehensive unit tests using JUnit.
* **Static Analysis**: Tools like SonarQube for code quality and security checks.
* **Continuous Integration**: Automated builds and tests with tools like Jenkins or GitHub Actions.

# Leesons Lernend

1. **Security**: Importance of securing APIs with JWT and HTTPS.
2. **Scalability**: Designing a scalable architecture from the beginning.
3. **User Feedback**: Incorporating user feedback early to refine features.
4. **Collaboration**: Effective team collaboration and communication are crucial.