Roommate App with Spring Boot and MongoDB

Eric Moore

CST-451 Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Professor Mark Reha

Revision: v0.0.1

Date: 11/29/20

**ABSTRACT**

Students at universities are often grouped with other students who make for incompatible roommates. Due to this incompatibility, the potential for roommate conflict is consistent at a level that requires the university to set aside resources—time, money, and personnel—to resolve these conflicts as they arise. Despite some student housing portals having a roommate preference questionnaire for students to fill out, this application lacks any convenient opportunities for students to select compatible roommates for on-campus housing.

This project seeks to solve this problem by allowing users to log into this app, fill out the questionnaire from the housing portal, and view each other’s answers and contact information much quicker than on the student portal. This will allow students to develop groups with more compatible roommates much faster and more conveniently than through the student portal, and allow them to contact each other outside of the portal and bond as well as discuss plans for moving in. This reduces the potential for roommate conflict and allows the university to dedicate these resources to other projects, which will ultimately save the university money and allow for greater expansion rate.

This application will be a dynamic web application programmed with Thymeleaf and Bootstrap for the frontend, Spring Boot for the backend, and MongoDB Atlas for the database server. The logical structure of the application will follow several common design patterns including MVC, N-Layer, Façade, and DAO design patterns. It will be deployed as a Heroku web application.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  |  | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Introduction 5

Detailed High-Level Solution Design 6

Detailed Technical Design 9

Appendix A – Technical Issue and Risk Log 20

Design Introduction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | Data Dictionary | The Data Dictionary contains an explanation of each collection in the MongoDB database. |  |  |  |

Detailed High-Level Solution Design

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1 – Build POC application with Spring Boot and MongoDB | To determine if Spring Boot and MongoDB were the right choice for my project | Spring Boot and MongoDB works well, and so I am using these. |
| 2 – Build POC application with Log4J and SLF4J | To determine if Log4J and SLF4J are in the application’s technical scope. | To be determined. |

|  |
| --- |
| Hardware and Software Technologies |
| 1 – MongoDB Atlas (MongoDB v4.2.10) |
| 2 – Spring Boot v2.3.5 |
| 3 – Apache Maven v4.0.0 |
| 4 – GitHub |
| 5 – Heroku v18 |
| 6 – Java Development Kit v11.0.8 |
| 7 – Bootstrap v4.5.2 |
| 8 – Thymeleaf v3.0.11 |
| 9 – Visual Studio Code v1.49.3 |

**Logical Solution Design:**

Diagram

Description automatically generated

**Physical Solution Design:**

Diagram

Description automatically generated

Detailed Technical Design

**General Technical Approach:**

The point of this application is to allow the user to create and sign into an account, create and update their profile, fill out the roommate preferences questionnaire, and view the profiles and questionnaire responses of other users on the website. This is done with a focus on the backend, using MVC and N-Layer design patterns to form the structure of the application. The user data will be contained in a MongoDB Cluster using MongoDB Atlas, an online application which serves as a cloud database service. Inversion of Control and Dependency Injection are also used in this application to allow for increase code interchangeability. The most significant concern in the making of this application is providing a simple, efficient design for managing full functionality without the unnecessary sacrifice of valuable resources.

**Key Technical Design Decisions:**

The framework chosen for this application is Java Spring Boot v2.3.5. This is the framework I am most familiar with, which allows for quicker completion of production while still maintaining quality performance. Spring Boot is also very common in the industry, so there is great incentive for me to use this framework. The database technology, MongoDB Atlas v4.2.10, allows for fast and flexible documenting of data created by users in the application. The Proofs of Concept necessary to test these frameworks and technologies have fully passed, and this project seeks to bear the fruit of it.

**Database ER Diagram:**

Diagram

Description automatically generated

**Database DDL Scripts:**



**Sitemap Diagram:**

Diagram

Description automatically generated

**User Interface Diagrams:**Diagram

Description automatically generatedTable

Description automatically generatedDiagram, table

Description automatically generated

**UML Diagrams:**

Message Functionality Class Diagram

Diagram

Description automatically generated

Profile Functionality Class Diagram

Diagram

Description automatically generated

QuestionnaireResponse Functionality Class Diagram

**Diagram

Description automatically generated**

User Functionality Class Diagram

**Diagram

Description automatically generated**

**Sequence Diagram**

**Diagram

Description automatically generated**

**NFR’s (Security Design, etc.):**

One non-functional requirement of this application is to provide security by preventing users from accessing pages while not signed into a legitimate account. This security is provided through Spring Security in the form (primarily) of the WebSecurityConfigurerAdapter. When an instance of this adapter is configured properly, it will redirect any user attempting to access a secured page to the application’s login page. Only after the user signs in will they have access to the secured pages of the website.

**Operational Support Design:**

To support monitoring and logging, this application will use the Log4J logging façade in conjunction with the SLF4J logging framework. Relying on the Interceptor design pattern, these logging technologies will be used to track users’ navigation within the site as well as the transfer of data between the logical layers of the application. These logs will be kept in Heroku’s application logs.

Appendix A – Technical Issue and Risk Log

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| R | Unfamiliar with Spring Boot | Project Stagnation | Learn and develop proof of concept for Spring Boot. | Eric Moore | Low | 09/27/20 | 10/04/20 | 10/04/20 |
| R | Unfamiliar with React Native | Project Stagnation | Switch to web browser application. | Eric Moore | Low | 09/27/20 | 10/04/20 | 10/04/20 |
| R | Unfamiliar with MongoDB | Project Stagnation | Learn and develop proof of concept for MongoDB. | Eric Moore | Low | 09/27/20 | 10/04/20 | 10/04/20 |
| R | Unfamiliar with MaterialUI | Project Stagnation | Switch to Bootstrap. | Eric Moore | Low | 09/27/20 | 10/04/20 | 10/04/20 |
| R | Unfamiliar with Log4J | Lacking monitoring and logging. | Learn and develop proof of concept for Log4J. | Eric Moore | Medium | 11/29/20 | 12/06/20 |  |
| R | Unfamiliar with SLF4J | Lacking monitoring and logging. | Learn and develop proof of concept for SLF4J. | Eric Moore | Medium | 11/29/20 | 12/06/20 |  |
|  |  |  |  |  |  |  |  |  |