|  |  |
| --- | --- |
|  |  |
| **Project plan** | Looking for a study programme in the south of the Netherlands | Fontys  University |
|  |  |
|  |  |

Contents

[1](#_Toc160010173)

[Description 2](#_Toc160010174)

[Mock-up 2](#_Toc160010175)

[FRs 3](#_Toc160010176)

[Non-FRs 4](#_Toc160010177)

[Stack 5](#_Toc160010178)

[Time plan 6](#_Toc160010179)

[Research plan 6](#_Toc160010180)

[Sprint 0 6](#_Toc160010181)

Project name: HouseHunters

### Description

HouseHunters is a real estate application aimed towards both agencies and private sellers. It resembles a marketplace where properties can be listed by their owners (or on behalf of them) and interested buyers can bid on those properties. Once the listing expires, the highest bidder has the right to buy the property.

### Mock-up

A screenshot of a computer

Description automatically generated

### FRs

|  |  |
| --- | --- |
| **Functional requirement** | **Description** |
| **Login/Register** | Users will be able to register and login in the application |
| **(CRUD) Listings** | Users will be able to manage their listings only. Administrators can modify any listing. |
| **Bids** | Users who are not the owners of their listing can place bids. Once the bidding is over, the highest bidder is concidered the winner. |
| **Comments** | Users will be able to post comments for listings. |
| **Administrators** | Administrators are not limited to any functionality within the application |

### Non-FRs

|  |  |
| --- | --- |
| **Functional requirement** | **Description** |
| **Security** | Application needs to be secure with no vulnerabilities |
| **Stability** | Application must not contain any major bugs that degrade the user experience. A request to the client must not take longer than 1 second and the app can handle 100 000 concurrent requests/sec. |
| **Availability** | The application must be able to support many concurrent users (aiming for 1 000 000) at once. |
| **Downtime** | The application must experience minimal downtime in case of a technical issue |

### Potential stack

Frontend – React (Javascript + Typescript)

Backend – Microservices + Express + Typescript

Database – MongoDB (NoSQL)

Kubernetes

Google Cloud – Bucket + Deployment

### Time plan

This project is going to follow delivery dates (Sprints). There are 5 sprints in total.

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Start** | **End** |
| **Sprint 0** | 19 Feb 2024 | 2 March 2024 |
| **Sprint 1** | 2 March 2024 | 22 March 2024 |
| **Sprint 2** | 22 March 2024 | 12 April 2024 |
| **Sprint 3** | 12 April 2024 | 10 May 2024 |
| **Sprint 4** | 10 May 2024 | 31 May 2024 |
| **Sprint 5** | 31 May 2024 | 12 June 2024 |

### Research plan

For each sprint there will be topics to research. The main motivation is to gather new knowledge in order to move the project forward.

The order in which the topics are researched by is going to follow the recommended research default provided by Fontys

A diagram of a software development

Description automatically generated

Sprint 0

Sprint 0 will be about researching enterprise architecture and enterprise software platforms. This is a sensible way to start learning about how enterprise software is about.

The desired outcome it to be able deliver a technology choice document based on an evaluation matrix. The technologies described by the document will be used in the individual project to build an enterprise architecture

Deliverables: Project plan, Technology choice document

Sprint 1

Sprint 1 will be about researching how to shape the project‘s architecture, what kind of conventions to follow and how to assure the quality of the project.

The desired outcome it to be able choose an architecture style for the project and be able to argument why it is chosen and how it can be quality assured.

Deliverables: (Initial) Architecture document, partial walking skeleton

Sprint 2

Sprint 2 will be about researching message brokers and security. This is the last step towards being able to deliver a walking skeleton which can be further built afterwards.

The desired outcome it to be able choose an implement a message broker, as well as monitor security of the application. All of this will deliver towards a walking skeleton at the end of the sprint.

Deliverables: Project with implemented CRUD functionality and message brokers, ready for scaling.

Sprints 3 and 4

Sprint 3 and 4 will be about researching dev ops and cloud services. This is the next logical step towards having a complete CI/CD cycle and have the project deployed on a cloud service

The desired outcome it to be able choose a cloud provider and argument why said provider is chosen. Also, a near-complete looking CI/CD pipeline is a good objective.

Deliverables: A project running in a cloud environment with scalability

Sprint 5

The last sprint will explore GDPR regulations and observability and performance. This is the last step towards having a completed enterprise application running.

The desider outcome it to have an application that is GDPR compliant as well as being able to observe its performance.

Deliverables: GDPR compliant and observable system