**Project Plan**

***HUISTER***

|  |
| --- |
| **Date : 19/9/2023** |
| **Version : 1** |
| **State :** |
| **Author : Calvin Kwan** |

#### Version history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **Author(s)** | **Changes** | **State** |
| 1 | 19/9/2023 | Calvin Kwan |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Distribution**

|  |  |  |
| --- | --- | --- |
| **Version** | **Date** | **Receivers** |
|  |  |  |
|  |  |  |

Contents

[1. Project assignment 4](#_Toc146017323)

[1.1 Context 4](#_Toc146017324)

[1.2 Problem 4](#_Toc146017325)

[1.3 Goal of the project 4](#_Toc146017326)

[1.4 Scope and preconditions 4](#_Toc146017327)

[1.5 Strategy 4](#_Toc146017328)

[1.6 End products 5](#_Toc146017330)

[2. Activities and time plan 6](#_Toc146017331)

[2.1 Phases of the project 6](#_Toc146017332)

[2.2 Time plan and milestones 6](#_Toc146017333)

[3. Testing strategy and configuration management 7](#_Toc146017334)

[3.1 Testing strategy 7](#_Toc146017335)

[3.2 Test environment and required resources 7](#_Toc146017336)

[3.3 Configuration management 8](#_Toc146017337)

[4. Risk and Mitigation 9](#_Toc146017338)

# Project assignment

## Context

Huister is a web application which allow people to rent a property in Netherland. Huister also landlord to rent their own property. Both the landlord and the renter can also negotiate the contract so both party are able to feel satisfied with their agreement. On the other hand, both landlords and renters can also cancel or reject the contract offer so they can pursue a better offer for each of them.

**Key Feature :**

* User registration Authentication
* Property Listing
* Send a negotiated offer to landlord

**Target Audience:**

* Customers
* Property owners
* Admin

## Problem

In Netherland, housing is a quite problematic problem. It is hard for renters to get an accommodation during their stay in Netherland. Even if they are able to find a place to stay, the cost could be not suitable to their wish.

## Goal of the project

The purpose of this project is to help people renting a property with a suitable price. With being able to negotiate, both customers and property owner is able to achieve an agreement that will make both of them satisfied. This project also has another goal which is to help property owners to gain passive income using their unused property

## Scope and preconditions

|  |  |
| --- | --- |
| **Inside scope:** | **Outside scope:** |
| 1. User registration and login system | 1. Mobile Application development |
| 1. Property listing and searching functionality based on city | 1. Advance feature such as AR ( Augmented Reality ) integrations |
| 1. Contact and messaging system | 1. Support multiple language |

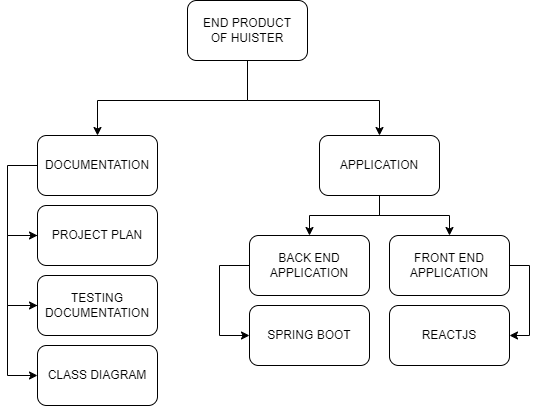
## Strategy

The strategy for the development of Huister is by utilizing scrum method. Even though scrum is mainly used for a team project, it also can be used in a one man project also. Here are the approach of scrum methodology in Huister development :

1. Defining the project
2. Set some milestones for each sprint that has to be accomplished during the sprint
3. At the end of the sprint, review the goal that has been done.
4. Reflecting on sprint performance and identify what can be improved.

The reason Scrum methodology is chosen as the approach is none other than the fact that with scrum it will be easy to create goals during each sprint so the workload will be divided equally during each sprint. Scrum also allows flexibility in the project so some feature can be added or cancelled according the situation

## End products



# Activities and time plan

## Phases of the project

**Initialization**

* Create Product Backlog
* Create Project Plan

**Sprint**

* **Sprint 1**
* **Sprint 2**
* **Sprint 3**
* **Sprint 4**
* **Sprint 5**
* **Sprint 6**

## Time plan and milestones

|  |  |  |  |
| --- | --- | --- | --- |
| **Phasing** | **Effort** | **Start date** | **Finish date** |
| 1. Initialization | Moderate | Week 1 | Week 3 |
| 1. Sprint 1 | Moderate | Week 1 | Week 3 |
| 1. Sprint 2 | Moderate | Week 4 | Week 6 |
| 1. Sprint 3 | Moderate | Week 7 | Week 9 |
| 1. Sprint 4 | High | Week 10 | Week 12 |
| 1. Sprint 5 | High | Week 13 | Week 15 |
| 1. Sprint 6 | High | Week 16 | Week 18 |

# Testing strategy and configuration management

## Testing strategy

There are several testing that will be done in the development process of this application. Those tests are listed below :

1. **Unit Testing**

Unit testing that will be done by utilizing Junit, which is a plugin for java that enable an automation in unit testing. This testing will be done for the back end software, which is using Spring Boot.

1. **Integration** **Testing**

this testing will be done using Postman. The purpose of this testing is to check whether the data, in the form of JSON, has the correct response according to the request.

1. **System Testing**

Tests that will be done in system testing will be using black box method. This type of testing will be able to show bugs that happen when interacting from front-end. All of the tests will cover around 70-90% of the main feature.

## Test environment and required resources

**Developer Environment**

* Developer’s local machine

**Testing Environment**

* Developer’s local machine
* GitLab
* Docker ( Future plan )

## Configuration management

**GIT Repository**

We will use Git for version control. A Git repository will be set up to track code changes. We will follow the GitFlow branching strategy, with feature branches for new development and separate branches for releases.

**Change Requests**

Change requests will be documented and tracked using a tool like Jira. The change management process will involve evaluating the impact of changes on project scope and schedule.

**Problem Reports**

Problem reports related to defects or issues discovered during testing or production use will be logged and tracked. The severity of each issue will be assessed, and a prioritization process will determine the order of resolution.

# Risk and Mitigation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Impact** | **Probability** | **Prevention activities** | **Mitigation activities** |
| 1. Requirement   Uncertainty | Low to Moderate | Moderate | Invest time in analyzing the requirements | Use of iterative development |
| 1. Technical Complexity | Low to Moderate | Low | Choose the appropriate technology and architectural pattern | Trying to simplify the software design |
| 1. Third Party Dependencies | High | Low | Choosing a stable API | Monitor the third party system update |