**Rentler**

**Software Requirements**

**Specification**

**Version 1.0**

**Prepared by Kyryl Halmiz, Anrii Pyzh,**

**Kateryna Lehka, Marta Kostetska**

**10 November 2022**

**1. Introduction**

* 1. **Purpose**

The purpose of this document is to present a detailed description and outline the requirements for the Rental web application. It will explain the purpose and product features, the interfaces of the system, what the system will do and constraints under which it must operate.

* 1. **Intended Audience and Reading Suggestions**

The intended audience of this document includes all stakeholders of Rentler project who are supposed to review and sign-off this document. Tre primary stakeholders of the system include customers. It will be useful for those who want to rent an apartment and also for rental providers.

* 1. **Project Scope**

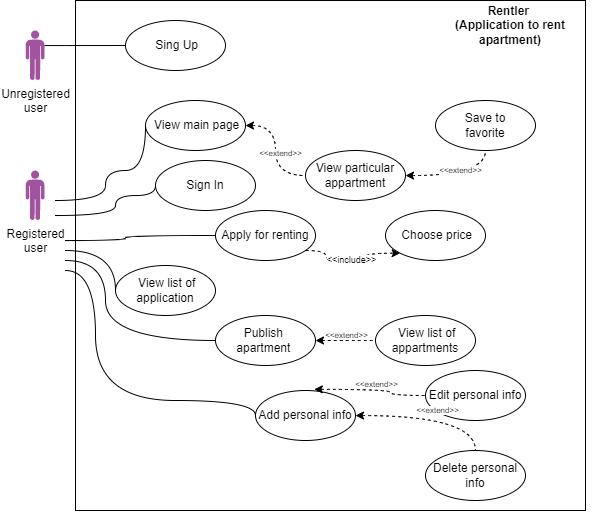
The system will provide all the functionality so that the tenant can find the right real estate, view all the details about the object, make offer to rent and contact the rental provider. On the landlord's side, will be opportunities to publish ads and provide all the necessary information, view all rental offers and reject them, or approve them. Also, users receive notifications by e-mail regarding submitted applications for renting apartment.

**2. Overall Description**

**2.1 Product Features**

* User profile management (create, edit, delete).
* Property application management (create, edit, delete).
* Displaying detail information about the property.
* Displaying all rental applications.
* Rental request management.
* Displaying application status.

**2.2 User Classes and Characteristics**

User classes and characteristics can be demonstrated by Use Case UML diagram:

**Use Case Description:**

|  |  |  |
| --- | --- | --- |
| **№** | **Name** | **Description** |
| 1 | Sing Up | Sing Up gives the possibility to enter name and password to become a registered user. |
| 2 | View main page | Main page is the page when user can see list of all apartments on site. |
| 3 | View apartment | On the main page, the user can select a particular apartment and view all information about the apartment. This information is about the number of rooms, several photos, and price. |
| 4 | Save to favorite | Users can save a particular apartment to list of favorite apartments to view them later. |
| 5 | Sign In | Sign In gives the possibility to enter the site. |
| 6 | Apply for renting | When the user selects a particular apartment, it is possible to apply for rent. |
| 7 | Choose price | In order to apply for rent it is needed to choose the price of the apartment. |
| 8 | View list of application | Then the user can view a list of applications which were created. So, users can view the list of apartments which they would like to rent. |
| 9 | Publish apartment | User can publish apartment and find the renter. |
| 10 | View list of apartments | Users can view a list of published apartments available to rent. |
| 11 | Add personal info | Users can add personal information about them. |
| 12 | Edit personal info | Users can edit personal information. |
| 13 | Delete personal info | Users can delete personal information. |

**2.4 Operating Environment**

Operating environment for the rent system is as listed below:

* Databases: PostgreSQL;
* Client/server system;
* Microservices architecture;
* Operating system: Linux/Docker;
* Platform: Java;

**2.5 Assumptions and Dependencies**

The system will be developed by using Java language. Other dependencies will also include:

* Spring Framework
* Spring Boot
* Spring Cloud
* Docker
* React
* PostgreSQL
* Google Cloud
* Kubernetes

The system will be available and usable in Chrome, Opera, Mozilla Firefox, Safari browsers and Microsoft Edge. React framework will let the system work as a native app without reloading the page.

**3. Requirements specifications**

**3.1 Functional Requirements**

* The system should allow users to log into their account by entering user email and password.
* Rental provider can delete the application by pressing “delete” button.
* Registered user can create rental application
* Registered user can edit all the fields of previously created application
* Registered user can view detailed information of all application
* Registered user can view all rental applications.
* Rental provider can reject/accept rental requests.
* Registered user can see the status of each rental request
* Registered user can save apartment to list of favorite apartments.
* Registered user can add personal info on Personal info page.
* Registered user can edit personal info on Personal info page.
* Registered user can delete personal info on Personal info page.

**3.2 Non-functional Requirements**

**Security:**

1. Password requirements - length, special, characters.
2. Password should be hashed.
3. Authentication and authorization according to the standard.
4. Command and Query Responsibility Segregation(CQRS) pattern should be followed.

**Performance:**

1. The application must load withing 2 seconds.
2. The system must be able to support 1000 users simultaneously.

**Reliability:**

1. The system must work even if errors have occurred.
2. The system must be available 95% time per month.
3. Downtime after a critical failure must not exceed 1 hours.

**Usability:**

1. The system must work as a native app without reloading the page.
2. The interface shall be easy to learn without a tutorial and allow users to accomplish their goals without errors.

**Supportability:**

1. The system shall be available and usable in Google Chrome, Opera, Mozilla Firefox, and Safari browsers.

**4. Other Requirements**

**4.1 User Interfaces**

The following mock-ups describe most features of user interface.

Graphical user interface, application

Description automatically generated

Sign Up Page

Graphical user interface, website

Description automatically generated

Page with detailed information about the property

Graphical user interface, website

Description automatically generated

Rental ads Page

Graphical user interface

Description automatically generated

Log in Page

Graphical user interface, application, Teams

Description automatically generated

Page where rental providers can create rental applications.

Graphical user interface, application

Description automatically generated

Page where rental providers can view all rental applications.

Graphical user interface, website

Description automatically generated

Page where rental provider can view his published property

Graphical user interface, application

Description automatically generated

Page where user can see the status of his applications.

Graphical user interface

Description automatically generated

Property publishing Page

Graphical user interface, application

Description automatically generated

Page where user can edit his personal information

Graphical user interface, application, website

Description automatically generated

Page where can view information about the user and his applications

Graphical user interface, application

Description automatically generated

Page where users can approve/reject rental request