Software Engineering
Software Requirements Specification (SRS) Document

XJM

Oct 24, 2022

Version 1.3

By: Muizz Muhammad, Joshua Neely, Xavier Freeman

We abide by the UNCG Integrity Policy

Table of Contents

1. Introduction

1.1 Purpose:

The goal of our project is to create a service for landlords and tenants to be able to list houses for rent and tenant ability to rent the house from the tenant. All in one online application.

1.2 Document conventions:

The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements for the Real Estate Management System (REMS). Client-oriented requirements describe the system from the client's perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer's perspective. These requirements include a detailed description of functionality, data, performance, and other important requirements.

1.3 Definitions, Acronyms, and Abbreviations

Include any specialized terminology dictated by the application area or the product area.

Java	Programming language maintained by oracle
Spring Boot	Java framework is used to create microservices built by pivotal teams. Create production-ready standalone application
DBS	Abbreviation for Database Systems
SQL	Structured query language to query databases
REMS	Real Estate Management System
HTML/CSS	Markup language to create websites

1.4 Intended audience:

Real Estate Agents: Will be able to use the application to contact both tenants and clients about rental or purchasing the property.
Tenants and Clients: Will use the application to view the status of properties and can buy, sell, and rent a property.
Property Manager: Will have the ability to handle aspects of household properties.
Describe which part of the SRS document is intended for which reader. Include a list of all stakeholders of the project, developers, project managers, and users for better clarity.

1.5 Project Scope:

This project's software goals align perfectly with the business as this will bring more revenue to the business while creating standalone applications to sustain the business and further grow the business in new fields.

1.6 Technology Challenges:

References:

- Java Documentation
- developer.Mozilla
- spring documentation
- w3schools

- Phillip Webb, D. S. (n.d.). Spring Boot Reference Documentation. Retrieved September 19, 2022, from https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
- *Java™ Platform Standard Edition 18 Development kit*. JDK 18 Readme. (n.d.). Retrieved September 19, 2022, from https://www.oracle.com/java/technologies/javase/jdk18-readme-downloads.html
- *CSS: Cascading style sheets*. MDN. (n.d.). Retrieved September 19, 2022, from https://developer.mozilla.org/en-US/docs/Web/CSS

2.1 General Description

2.2 **Product perspective:**

This application was created to make an easier way of communicating between clients, tenants, and real estate agents. This one stand-alone application can help landlords and tenants by meeting each other's needs. Our software will be able to process rental applications and notify tenants if an application is accepted.

2.2 Product features:

Users will be able to rent houses, and also place houses on rent with ease. Tenants will be able to easily process a transaction between the landlord.
A high-level summary of the functions the software would perform and the features to be included.

2.3 User class and characteristics:

A categorization and profiling of the users the software are intended for and their classification into different user classes
Our website application does not expect our users to have any prior knowledge of a computer, apart from using a web browser, or knowledge of property value. Our website application has removed the need for them to have property management knowledge and can focus on observing the status of their viewed property.

2.4 Operating environment:

Specification of the environment in the software is being designed to operate in.
This application is created in a browser therefore can work in any environment that also has a browser or access to a browser.

2.5 Constraints:

Due to the use of real estate data, it was difficult to show data for which house the data was displayed so with the implementation of google API address we are able to get precise house addresses and data related to the house.

2.6 Assumptions and dependencies:

External: Slow running internet may cause the application to run slow thus preventing smooth operating functions within the application.

A list of all assumptions that you have made regarding the software product and the environment along with any external dependencies which may affect the project

3. Functional Requirements

Functional requirements:

Muizz Muhammad, Tennant.

- Browse houses to rent on a web app
- Make an application to the landlord to rent a house

Xavier Freeman, Landlord.

- Ability to inform customers on property sale status.
- Ability to create/display property information for purchase.

Joshua Neely , Admin/Property Manager.

- Ability to control the aspects of the house.
- Ability to update the house information as needed.

Statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.

3.1 Primary

All the requirements within the system or subsystem in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements, and constraints if any.

- FR1: The system will allow the user to look up houses to rent.
- FR2: The system will allow the user to view prices and other information about specified locations.
- FR3: The system will automatically fill in data fields using the housing information.
- FR4: The system will allow the user to update an application after it's been posted.
- FR5: The system will allow the user to delete the house information, update the house information, or add the house information.
- FR6: The system will keep the user's application information and the server's housing information database synchronized within a 24 hours time frame.

3.2 Secondary: Some functions that are used to support the primary requirements.

4.Technical Requirements

4.1 Operating System & Compatibility

4.2Interface requirements

4.2.1 User Interfaces

The logic behind the interactions between the users and the software. This includes the sample screen layout, buttons and functions that would appear on every screen, messages to be displayed on each screen and the style guides to be used.

4.2.2 Hardware Interfaces

All the hardware-software interactions with the list of supported devices on which the software is intended to run, the network requirements along with the list of communication protocols to be used.

- **Browser** - **Computer** - **HTTPS**

4.2.3 Communications Interfaces

Determination of all the communication standards to be utilized by the software as a part of the project

4.2.4 Software Interfaces

The interaction of the software to be developed with other software components such as the frontend and the backend framework to be used, the database management system and libraries describing the need and the purpose behind each of them.

5. Non-Functional Requirements

Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.

5.1 Performance requirements

For Example
performance requirements need to be specified for every functional requirement. The rationale behind it also needs to be elaborated upon.

- NFR1(R): The system (including the local copy of the vehicle violation database) will consume less than 50MB of memory
- NFR2(R): The novice user will be able to add and post houses in less than 10 minutes.
- NFR3(R): The expert user will be able to add and post houses in less than 5 minutes.

5.2 Safety requirements

Secure Network:

Not using a secure network while accessing the application may lead to unwanted leaks of secure and personal information. Always consider choosing a strong safe password when creating accounts on the websites. Always log out when done using services.

5.3 Security requirements

Privacy and data protection regulations that need to be adhered to while designing the product. For Example:

- NFR4(R): The system will only be usable by authorized users.

5.4 Software quality attributes

5.4.1. Availability
5.4.2. Correctness
5.4.3. Maintainability
5.4.4. Reusability
5.4.5. Efficiency

Detailing the different qualities that need to be incorporated within the software like maintainability, adaptability, flexibility, usability, reliability, portability, etc.

5.5 Process Requirements

5.5.1. Development Process Used: **Agile methodologies.**
5.5.2. Time Constraints: **The enclosed amount of time to complete the project**.
5.5.3. Cost and Delivery Date **Cost: of the project is $0. Date: TBA.**

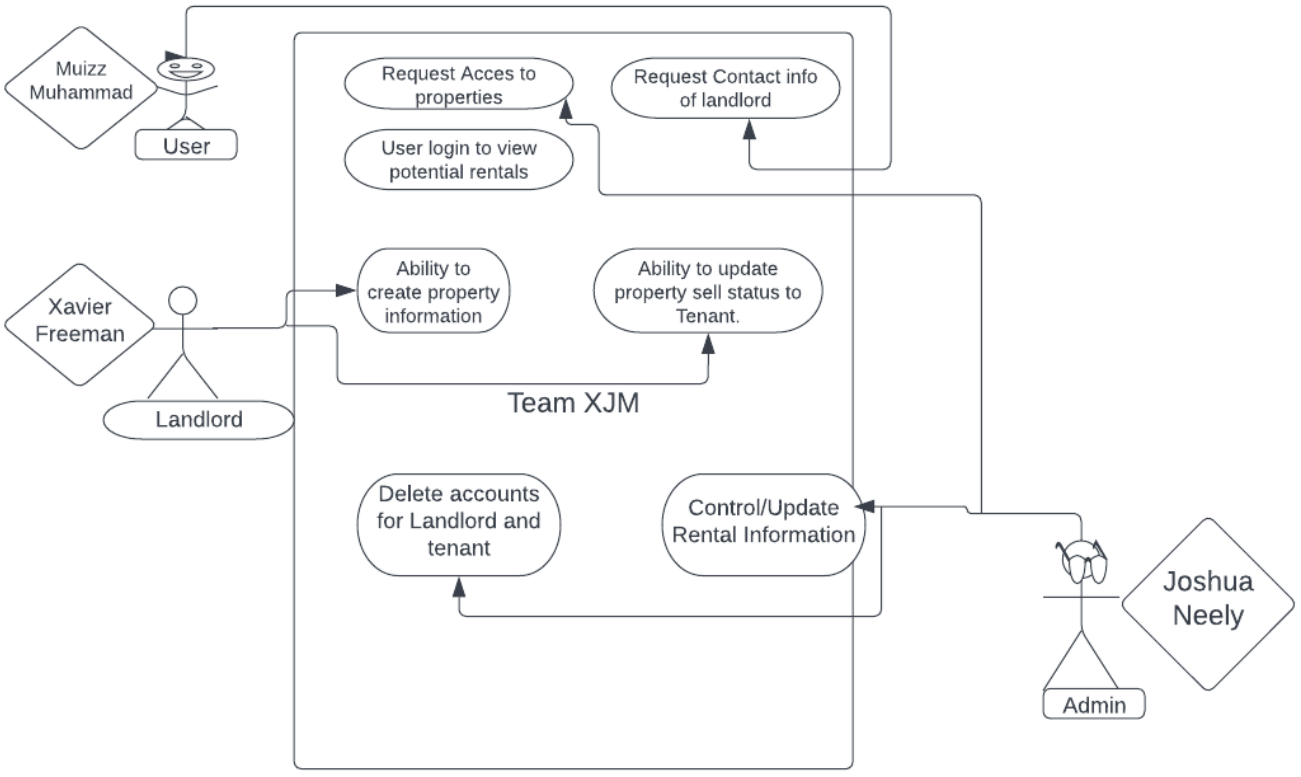
5.6 Other requirements

These may include legal requirements, resource utilizations, future updates, etc.

- NFR4(R): The system will conform to FERPA guidelines to maintain users' privacy.

6. Use Case Model

6.1 Use Case Model Diagram:



6.2 User case description

- **Request Access to website**
- The admin provides the landlord and tenant with login information for them to have private access.
- **control update information**
  -The admin has the ability to assign tasks for the landlords and tenants.

6.3 Use Case Description - Scenarios

- a**.Request Access to properties:**

  i. Initial Assumption: The admin has a login account.

  ii. Normal: The landlord and tenant can log in to view information

  iii. What can go wrong: The landlord or tenant credentials do not match because the username or password is not the same as they are in the database.

  iv. Other activities: The landlord or tenant can reset their password using the forgotten password

  v. System state on completion: The admin is logged in and can see the information of the landlord and tenant.

- b.**control update information:**

  i. Initial Assumption: The administrator logs in and creates a login for potential tenants and landlords so each party could post information on the website which saves it to our database.

  ii. Normal: Admin logs in to the system. Does daily tasks for updating the system and removing properties that are no longer active in the system. Also provides login info for landlords and tenants so they can have smooth access to the system.

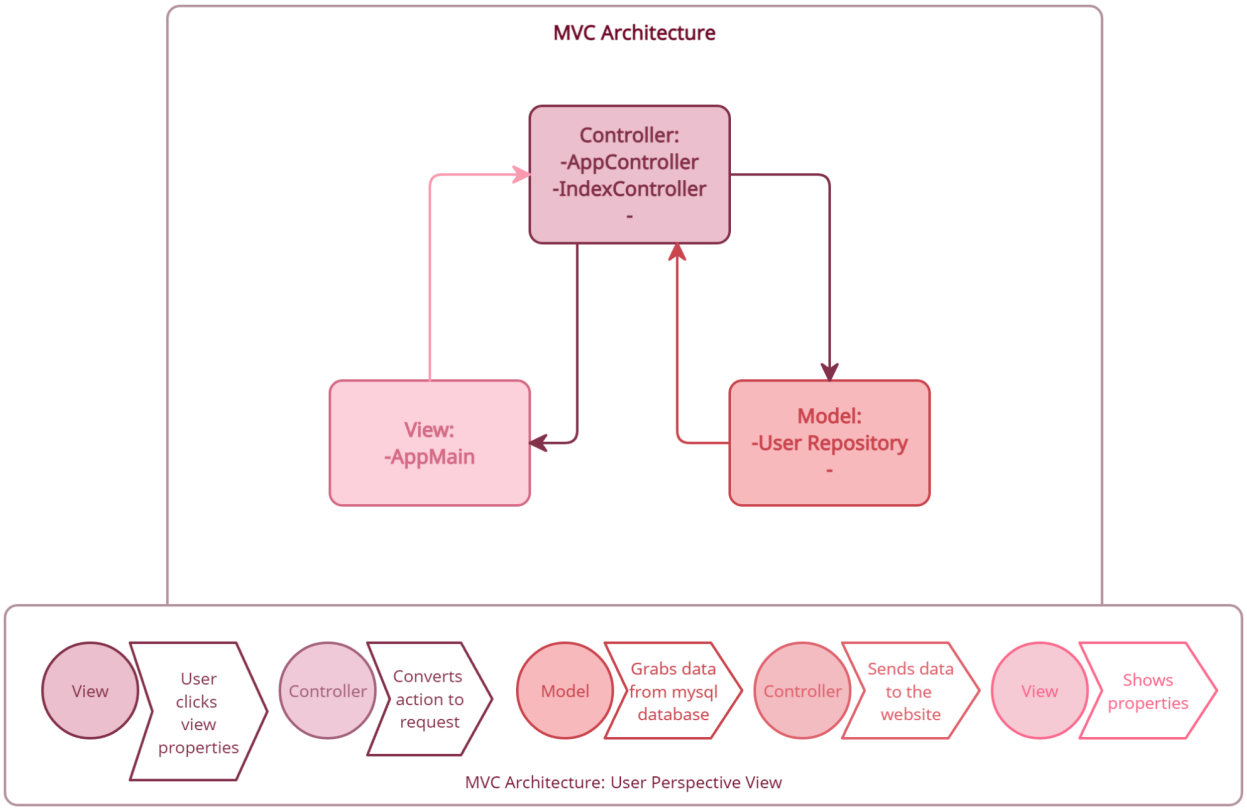
  iii. What can go wrong: Admin providing access to login info can provide tenants with landlords' info and vice versa.

  i**v. Other Activities:** Assign the task to upload house pics on the system and also notify tenants of new updated listings nearby.

  v. System state on completion: The tasks are assigned successfully. Updated and shows the assignee and the assignee being able to view their tasks.

7. Software Architecture

MVC Architecture:



-

Schema:

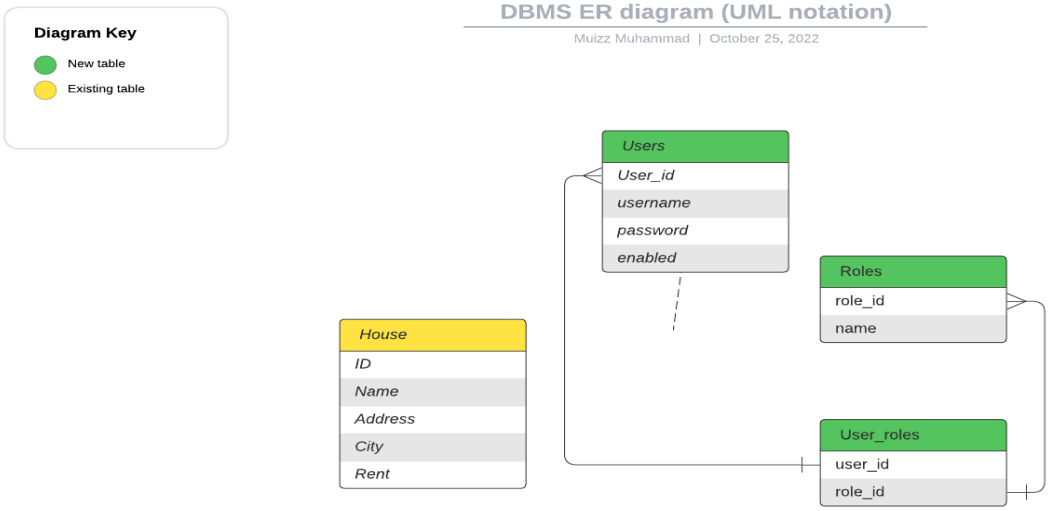
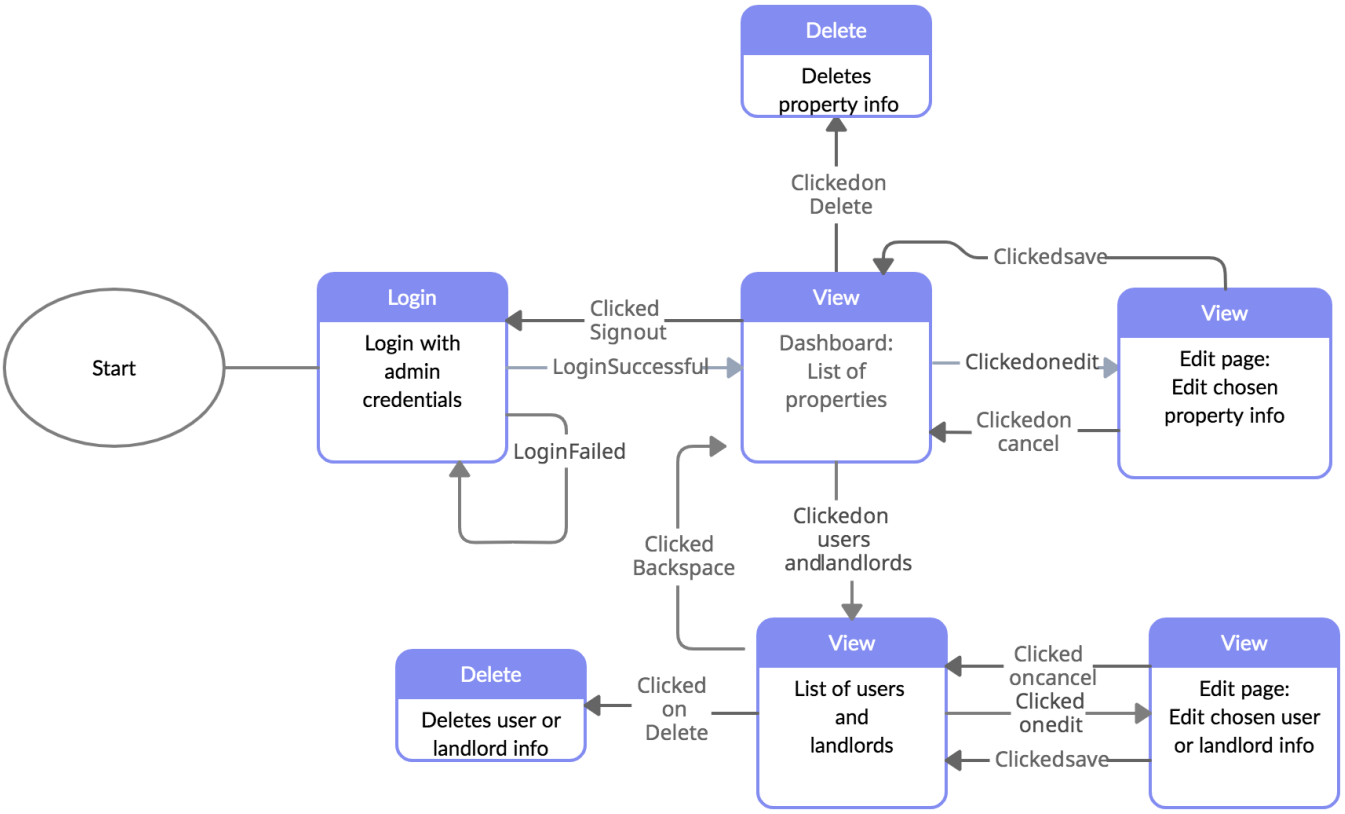


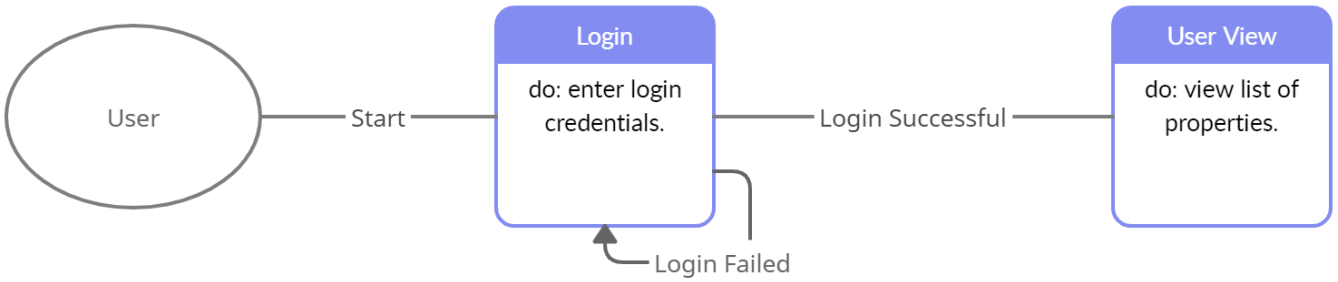
Diagram Key
New table
Existing table
DBMS ER diagram (UML notation)
Muizz Muhammad | October 25, 2022
Users
User_id
username
password
enabled
Roles
role_id
name
House
ID
Name
Address
City
Rent
User_roles
user_id
role_id

8. Software Design

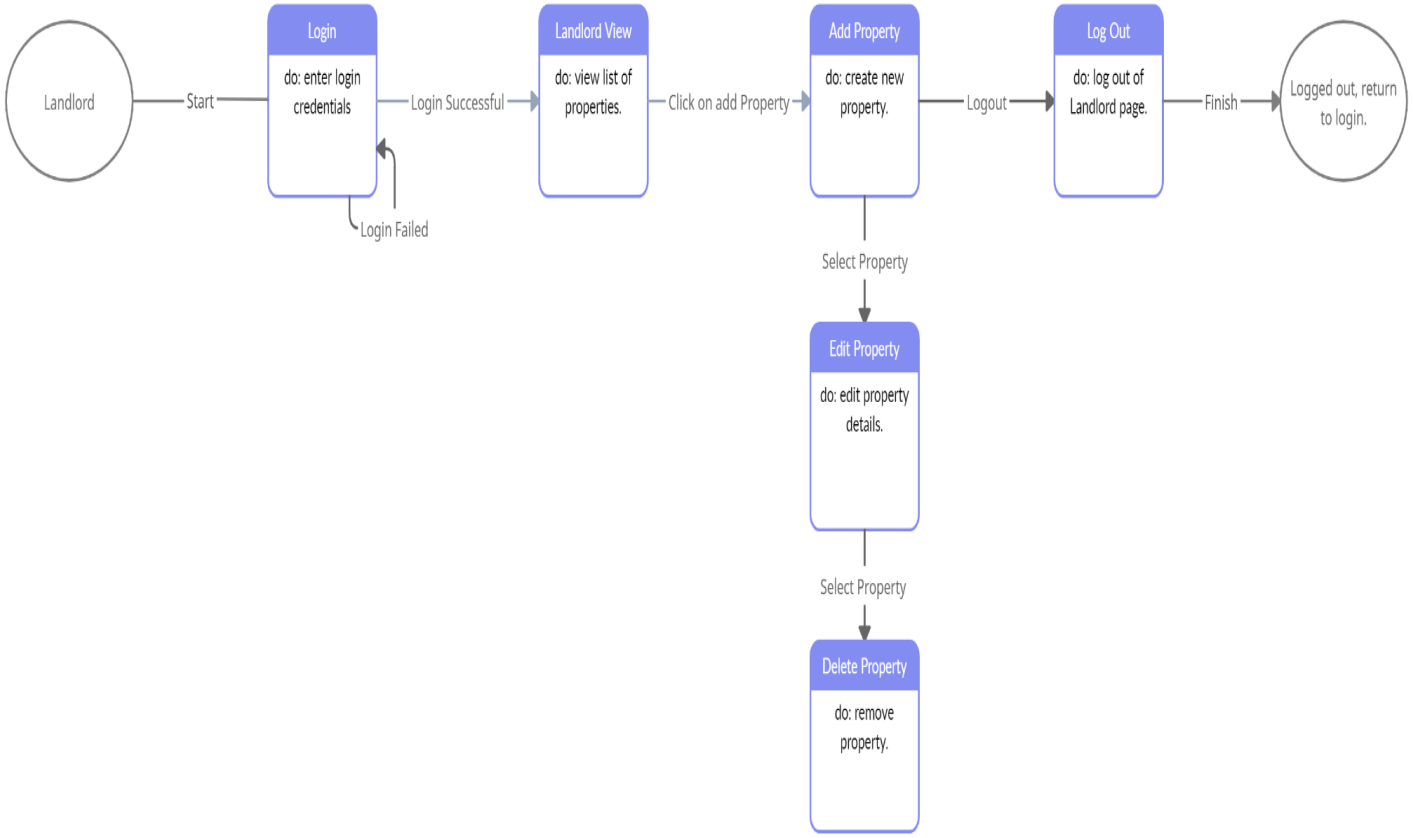
Admin State Diagram:



User State Diagram:



Landlord State Diagram:



UML Class Diagram: