

CS383 GROUP PROJECT

SOFTWARE DESIGN DOCUMENT(SDD)



Report By :

Sadeem Al-resaini

ID:421202041

Reef Alkhudhairy

ID: 421202087

Leen Alsheeb

ID:421202006

Rana Alnughaimshi

ID:392207728

Latifah Alamer

ID: 421201944

Lina Altuwayjiri

ID: 421201911

Raghad Alshammari

ID: 421202094

Supervisor: Dr.Delel

Rhouma

CONTENT

1. Introduction

1.1. Purpose

1.2. Scope

1.3. References

1.4. Software Process Activity Model

1.5. Structure

2. System Overview

2.1 User Registration.

2.2 Property Listing and Management.

2.3 Booking and Reservation.

2.4 Search and Filtering.

2.5 Communication and Direct Messaging.

2.6 Secure Payment Integration.

3. System Architecture and Components Design

3.1. Architecture Description

3.2. Component Decomposition Description

3.3. Detail Component Description

3.4. Design Rationale

4. Data Design

4.1. Database Description

4.2. Data Structure

4.3. Data-flow Diagram (DFD)

5. Design Details

5.1. Class Diagrams

5.2. State Diagrams

5.3. Activity Diagrams

5.4. Sequence Diagrams

6. Human Interface Design

6.1. Overview of the User Interface

6.2. Detail Design of User Interface

INTRODUCTION

1.1 Purpose:

The Software Design Document (SDD) acts as a precise technical blueprint for the property rental application. Its primary focus is on delineating the system architecture, data model, user interface, and essential modules/components needed to enable smooth property management, rental processes, and property search functionalities. By providing clear guidelines, the SDD empowers the development team to construct an efficient and all-encompassing platform that streamlines property management for both owners and tenants. It guarantees adherence to the initial requirements, fosters collaboration among stakeholders, and serves as an invaluable reference throughout the entirety of the development and implementation phases.

1.2 Scope:

The property rental application's scope includes system architecture, data model, user interface design, modules/components, guidelines, and collaboration. Its objective is to guide the development team in implementing a comprehensive and efficient property management platform. By defining the system's structure, data representation, UI, and components, the application ensures a cohesive and well-structured approach, fostering collaboration and meeting project requirements. The ultimate goal is to deliver a reliable and scalable solution that optimizes property management processes, catering to the needs of property owners and tenants. Additionally, the application prioritizes enhancing the user experience through intuitive and user-friendly interfaces, simplifying property management tasks, streamlining rental processes, and facilitating property search, resulting in a positive and satisfactory experience for both parties involved.

INTRODUCTION

1.3 References:

- CS383 lectures
- [UML tutorial. tutorialspoint.](#)
- [draw.io](#)
- [smart draw](#)

1.4 Software Process Activity Model

We chose the waterfall model for our rental property app because it offers a clear and sequential approach that suits our well-defined project requirements. The stages in the waterfall model align with the distinct features of our app, making it easy to navigate through tasks like user authentication, property search, and payment processing. Its simplicity and lack of extensive iterations contribute to a straightforward development process. The focus on documentation ensures clarity at each step. We aim to follow a systematic path, completing one stage before moving to the next, ultimately delivering a stable and well-executed rental property app.

1.5 Structures:

1.5.1 Introduction:

This document provides a comprehensive overview of the software system, including its requirements, purpose, and scope. It outlines the various features and functionalities of the system in a general sense.

1.5.2 System Overview:

In this document, we will provide a comprehensive overview of the software system, covering its functionality, components, and underlying architecture. We will also explore the product perspective by discussing its features and the capabilities it offers to our users.

1.5.3 System Architecture and Components Design:

This document conducts a thorough analysis of the components comprising the software system, providing insights into its overall structure and design. Additionally, it delves into the design rationale behind the system, explaining the reasoning and considerations that influenced its architecture.

INTRODUCTION

1.5.4 Data Design :

This document outlines the approach to structuring, storing, and managing data within the software. It provides a detailed description of the data management system, including how data will be organized, stored, and accessed by the software.

1.5.5 Design Details:

This document outlines a comprehensive plan for the development, testing, and implementation of software features, functions, and interfaces in our rental app. To aid in understanding and implementation, we have utilized visual models such as class diagrams, state diagrams, activity diagrams, and sequence diagrams to depict the app's design, showcasing its structure, behavior, and flow. This plan ensures a systematic approach to building the rental app, resulting in a well-tested and smoothly integrated software solution.

1.5.6 Human Interface Design :

This document prioritizes the user's experience and interaction with the software system, with a specific focus on layout, navigation, and accessibility. It outlines design considerations aimed at enhancing usability and ensuring a user-friendly interface. The document covers aspects such as visual presentation, element organization, and intuitive navigation. Additionally, it addresses accessibility features to ensure inclusivity for users with diverse needs. Overall, the document aims to provide a seamless and user-centric experience within the software system.

SYSTEM OVERVIEW

The Property Rental Application is a platform designed to assist users in finding and renting properties for various occasions. The application provides a user-friendly interface and a comprehensive set of features to simplify the property rental process.

2.1 User Registration:

- Users can create an account in the Property Rental Application to access its functionalities.
- Registration involves providing basic personal information and creating login credentials.

2.2 Property Listing and Management:

- The application allows property owners to list their properties for rent.
- Users can search and browse through the available properties based on their preferences and requirements.
- Property owners can manage their listings, update property details, and set rental terms.

2.3 Booking and Reservation:

- Users can make property reservations through the application.
- They can specify the desired dates, duration, and location for their rental.
- The application checks for property availability and facilitates the booking process.

2.4 Search and Filtering:

- The application offers search and filtering options to help users find properties that meet their specific criteria.
- Users can filter properties based on location, price, rooms, and other relevant factors.
- The search functionality allows users to quickly find properties that match their preferences.

2.5 Communication and Direct Messaging:

- The Property Rental Application enables direct communication between users and property owners.
- Users can send inquiries, ask questions, and negotiate rental terms through the application's messaging system.
- This feature ensures effective and transparent communication between parties involved in the rental process.

2.6 Secure Payment Integration:

- The Property Rental Application integrates secure payment gateways to facilitate online transactions.
- Users can make rental payments through the application, ensuring a safe and convenient payment process.
- The application may support various payment methods and implement encryption and data protection measures.

SYSTEM ARCHITECTURE AND, COMPONENTS DESIGN

3.1. Architecture Description:

Within property rental applications, a robust software architecture is pivotal in shaping a user-friendly interface for both tenants and property owners. This architecture revolves around core components: efficient user management for registrations and profiles, seamless geolocation integration enabling visual property exploration, an intuitive search and filtering system to cater to user preferences, detailed property displays, a communication module fostering easy interaction, and a secure payment gateway ensuring safe transactions. Prioritizing user ease, security, and effective communication between tenants and property owners, this architecture aims to simplify the rental process while maintaining a user-friendly interface and safeguarding data integrity .

as this component:

1. User Management:

- Handles user registration, login, and profile management.

2. Geolocation Integration:

- Connects with maps to show property locations and enable location-based searches.

3. add Property

- Allow the owner to add his property, write details, and create a contract.

4. Search and Filtering:

- Allows users to filter properties based on preferences like price, location, and others.

5. Property Details Display:

- Shows detailed property information, including descriptions, images, and availability.

SYSTEM ARCHITECTURE AND COMPONENTS DESIGN:

6. Communication Module:

- Facilitates messaging between tenants and property owners.

7. Secure Payments:

- Authorizes and processes secure payments between users and property owners.

8. User Feedback Incorporation:

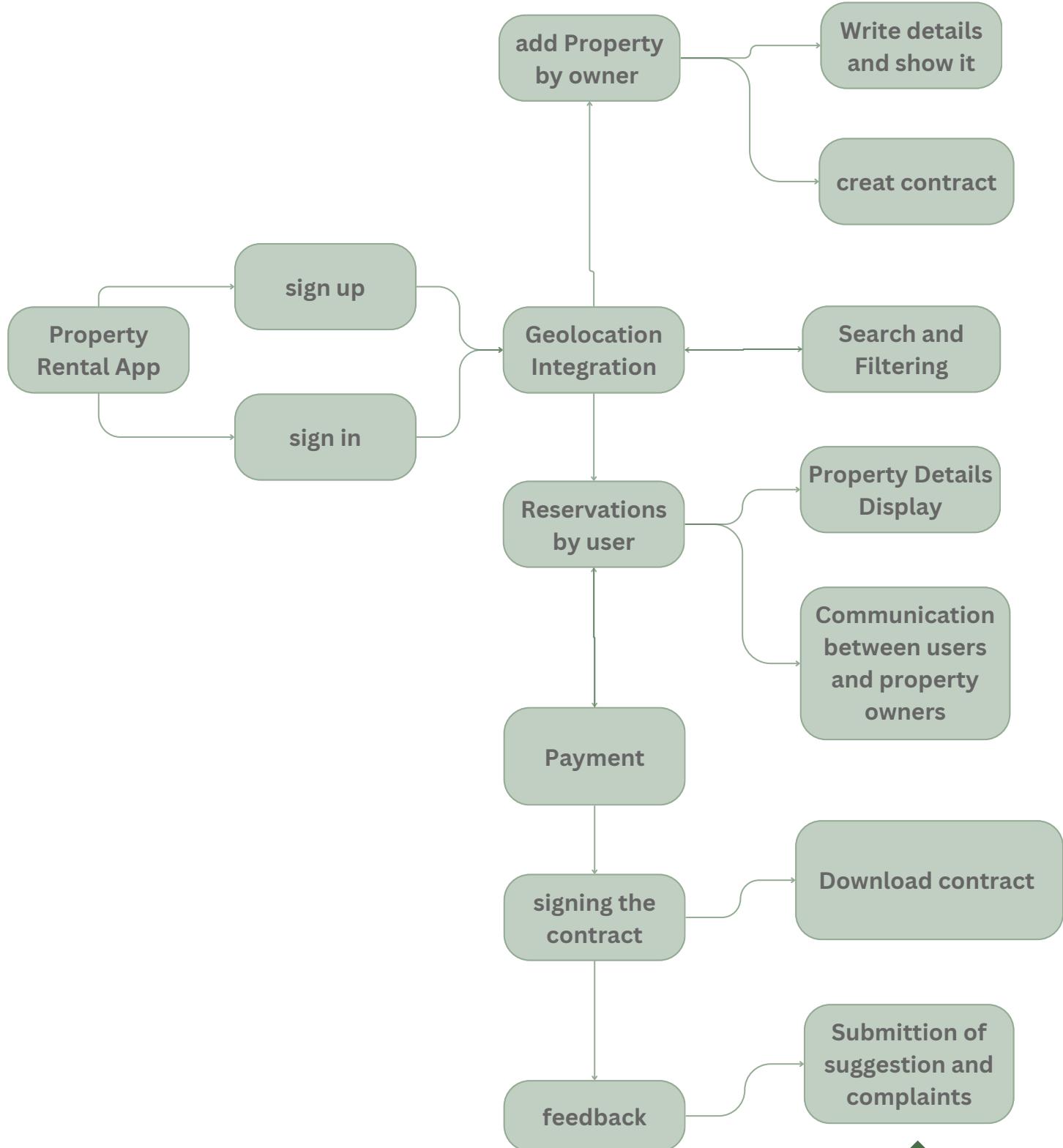
- Gathers and integrates user feedback systematically to refine the platform's usability, functionality, and features based on user suggestions.

9. contract :

- after the user signs the contract he can download it and keep it, and back anytime in his profile.

SYSTEM ARCHITECTURE AND COMPONENTS DESIGN:

3.2. Component Decomposition Description



SYSTEM ARCHITECTURE AND COMPONENTS DESIGN:

3.3. Detail Component Description

1- Registration Component:

This component allows users to create a new account by providing their personal information, such as name, email, and password. It includes a sign-up form where users can enter their details and register as new users of the app.

2- Sign-In Component:

The sign-in component enables registered users to log in to their accounts using their email and password. It provides a secure authentication mechanism to verify the user's credentials and grant access to their personalized account.

3- Geolocation Integration:

This component integrates geolocation services into the app, allowing users to determine their current location or search for properties in specific areas. It utilizes GPS or IP-based geolocation to provide accurate location data.

4- Search and Filtering Component:

The search and filtering component enables users to search for properties based on various criteria, such as location, property type, price range, number of rooms, and amenities. It provides a user-friendly interface for entering search parameters and displays relevant property listings based on the user's preferences.

5- Reservations Component:

The reservations component allows users to book properties they are interested in renting. It includes a booking form where users can select desired dates, specify the duration of stay, and review the rental terms and conditions. Users can also view their existing reservations, modify or cancel bookings if permitted.

SYSTEM ARCHITECTURE AND COMPONENTS DESIGN:

6- Property Details Display Component:

This component displays detailed information about a selected property, including property images, description, amenities, location map, rental price, availability calendar, and reviews/ratings from previous tenants. It provides an intuitive interface for users to explore property details before making a reservation.

7- Communication between Users and Property Owners Component:

This component facilitates communication between users and property owners. It may include features such as messaging/chat functionality, allowing users to inquire about property details, negotiate rental terms, and get responses from property owners in real-time.

8- Payment Component:

The payment component integrates secure payment gateways to enable users to make rental payments online. It supports various payment methods, such as credit/debit cards, digital wallets, or bank transfers. It ensures the privacy and security of users' financial transactions.

9- Contract:

This component generates the rental contract once the payment is completed. It includes the terms and conditions agreed upon by the property owner and the tenant . The contract can be viewed, downloaded, and digitally signed by both parties.

SYSTEM ARCHITECTURE AND COMPONENTS DESIGN:

10-Feedback Component:

This component allows users to provide feedback on their rental experiences, rate properties they have rented, and leave reviews/comments. It helps maintain transparency and trust within the app's community and assists other users in making informed decisions.

11- Submission of Suggestions and Complaints Component:

This component provides a platform for users to submit suggestions, complaints, or report any issues they encounter while using the app. It ensures that user feedback is heard and addressed appropriately, contributing to continuous app improvement.

12-Add Property:

This component enables property owners to add their properties to the platform and you can add from the home page . It consists of two sub-components

-Write Details and Show It -then Create Contract

These components collectively create a comprehensive property rental app that offers a seamless user experience, efficient property search and reservation functionalities, secure payments, and effective communication channels between users and property owners/agents.

SYSTEM ARCHITECTURE AND COMPONENTS DESIGN:

3.4 Design Rationale

User-Centric Experience: The app focuses on providing a seamless and user-friendly experience for searching and booking rental properties.

Simplified Property Search : Users can easily search for properties based on location, type, price, and other criteria.

Comprehensive Property Listings : Detailed property information, including photos, descriptions, amenities, and availability, helps users make informed decisions.

Easy Booking Process : The booking process is straightforward, allowing users to select dates, review terms, and make secure payments.

Communication Channel : Users can communicate directly with property owners through a built-in messaging system.

Ratings and Reviews : Users can leave feedback and ratings to help others make informed choices.

Personalized Recommendations : The app provides personalized property recommendations based on user preferences.

Responsive Design : The app is designed to be accessible and user-friendly across various devices.

By incorporating these elements, the app delivers a user-centric experience with simplified property search, easy bookings, efficient communication, and personalized recommendations.

Data design

4.1. Database Description

This database is architected to oversee a property rental system, which includes the management of users, their contact details, rental agreements, financial transactions, and real estate listings. The database comprises five core entities: User, Contract, Payment, Property, and ContactInfo. Below is a detailed description of each entity along with the relationships among them.

1. User:(`userID` (Primary Key): A unique identifier for each user. `username`: The user's chosen username for system access. `password`: The user's password for system access, which should be securely encrypted. `properties` (Foreign Key): A reference to the Property entity, indicating properties owned by the user. `contactInfo` (Foreign Key): A reference to the ContactInfo entity, containing the user's contact details.)

2. Contract:(`contractID` (Primary Key): A unique identifier for each rental agreement. `property` (Foreign Key): A reference to the Property entity, indicating the property involved in the rental. `tenant` (Foreign Key): A reference to the User entity, identifying the user who is renting the property. `owner` (Foreign Key): A reference to the User entity, identifying the user who owns the property. `startDate`: The date when the rental agreement commences. `endDate`: The date when the rental agreement concludes. `paymentStatus`: The current status of payments for the rental agreement (e.g., paid, overdue).)

3. Payment:(`paymentID` (Primary Key): A unique identifier for each payment transaction. `paymentDate`: The date the payment was made. `amount`: The monetary value of the payment. `status`: The status of the payment (e.g., completed, pending). `contract` (Foreign Key): A reference to the Contract entity, linking the payment to a specific rental agreement.)

4. Property:(`propertyID` (Primary Key): A unique identifier for each property. `address`: The physical address of the property. `size`: The size of the property (e.g., square footage, number of rooms). `rentAmount`: The amount of rent charged for the property. `status`: The current status of the property (e.g., available, rented). `owner` (Foreign Key): A reference to the User entity, indicating the user who owns the property.)

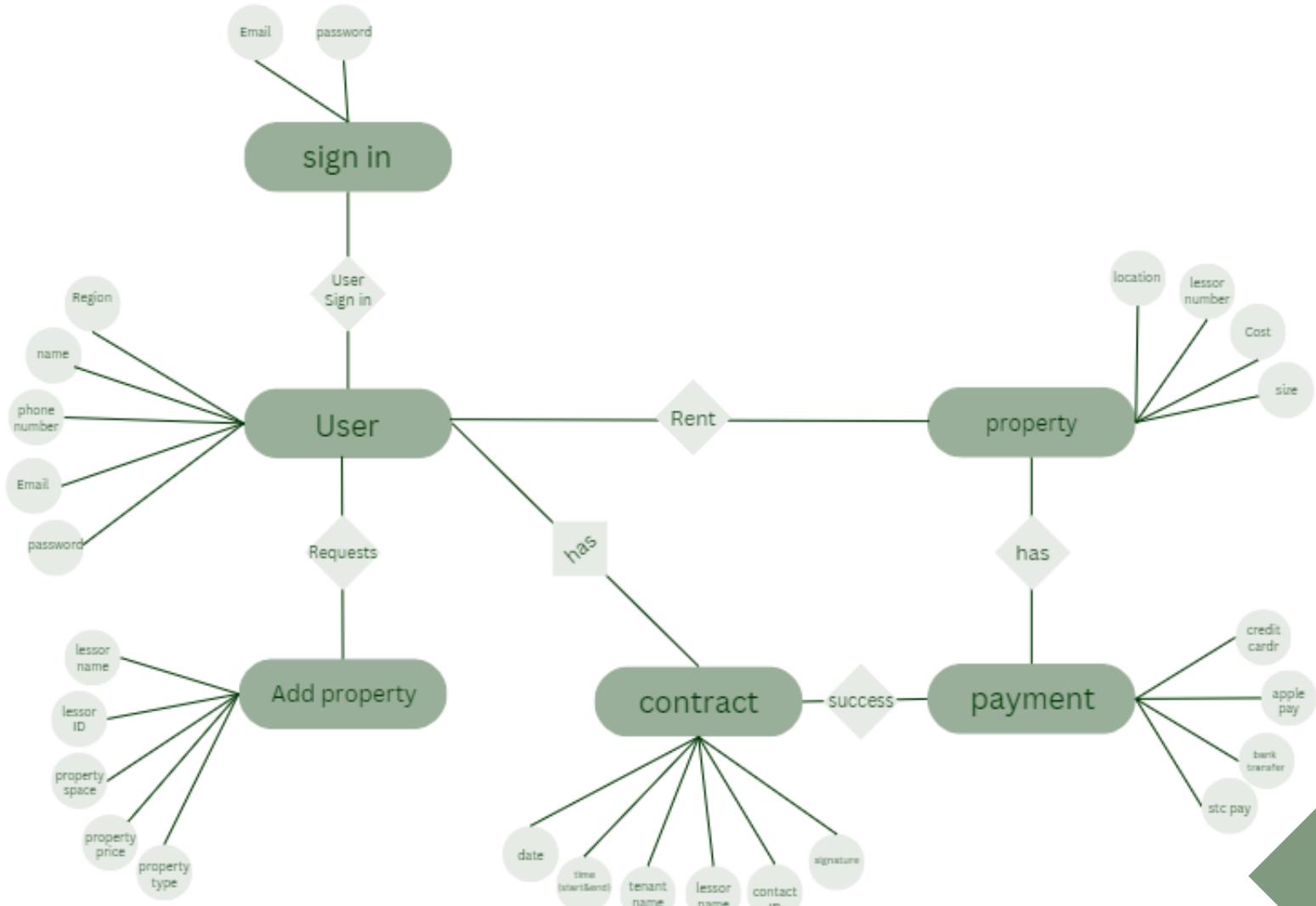
5. ContactInfo:(`contactInfoID` (Primary Key): A unique identifier for each set of contact information. `email`: The user's email address. `phone`: The user's phone number. `address`: The user's physical address.)

Data design

Relationships:

- The User entity is linked to the Payment entity, indicating that users are responsible for making payments.
- The User entity is linked to the ContactInfo entity, providing contact details for each user.
- The Payment entity is associated with the Contract entity via the contract foreign key, connecting payments to their corresponding rental agreements.
- The Property entity is linked to the Contract entity, denoting which property is tied to each rental agreement and who owns it.
- The User entity is connected to the Property entity in two capacities: users can be lessees under a rental agreement, and they can be proprietors of the properties.

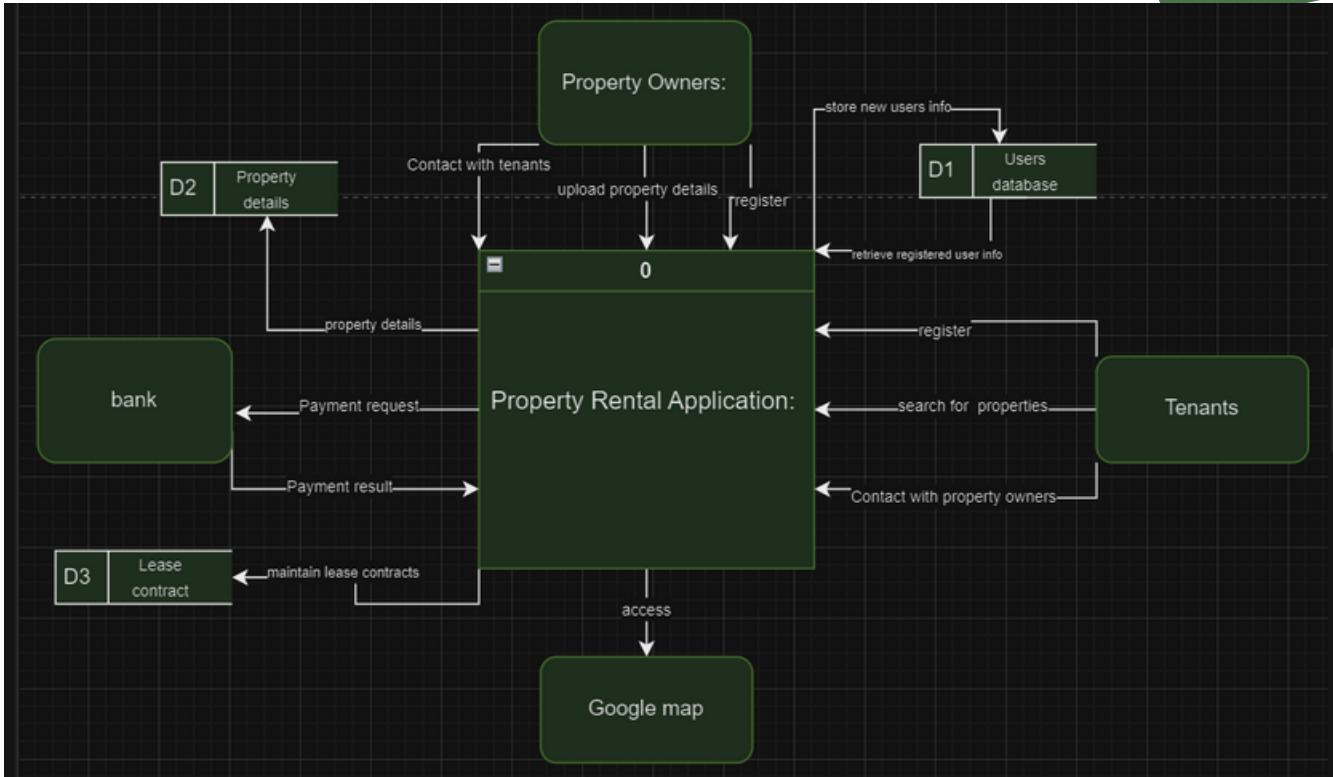
4.2 Data Structure



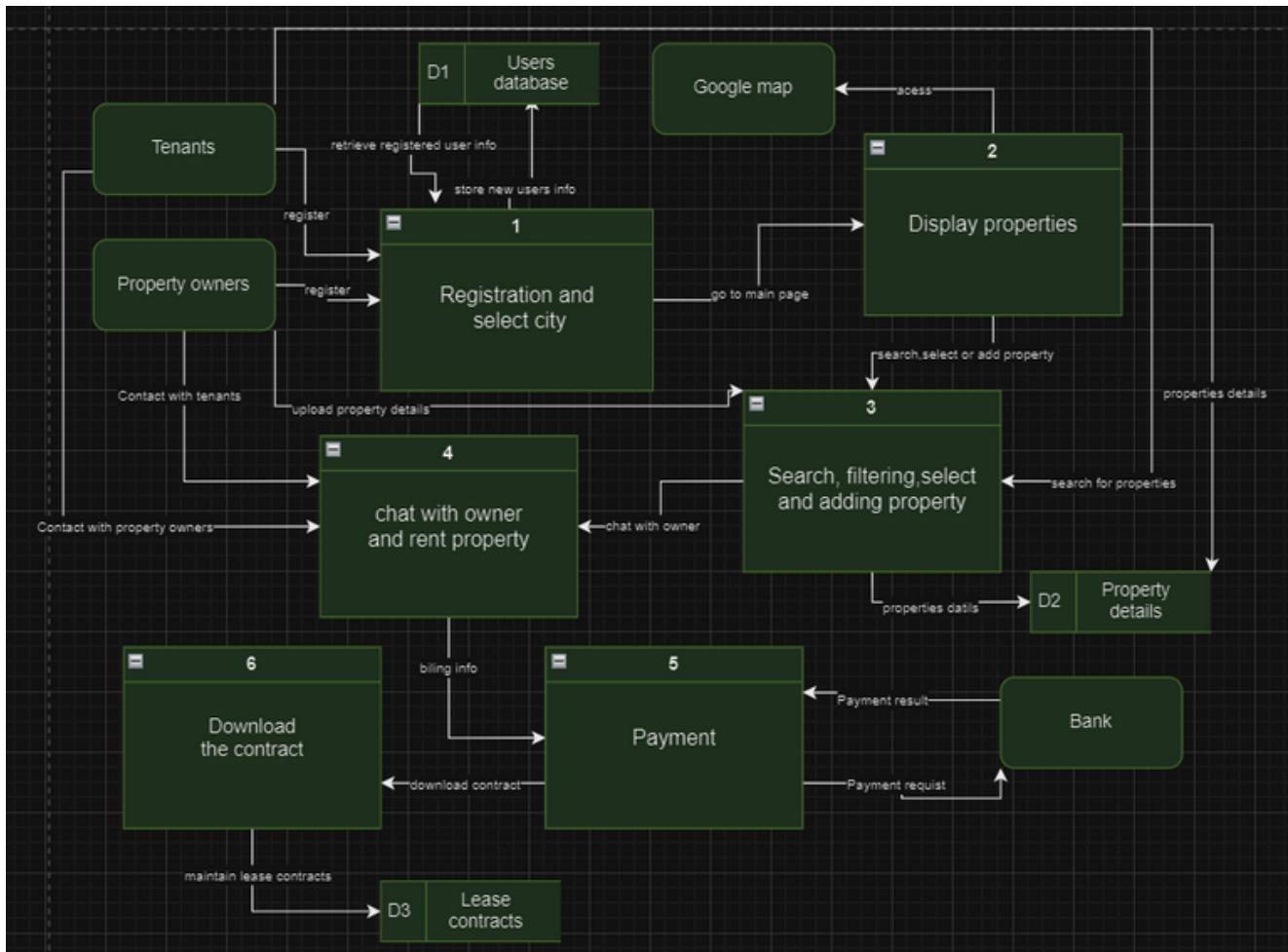
Data design

4.3. Data-flow Diagram (DFD)

1- Level 0 DFD:

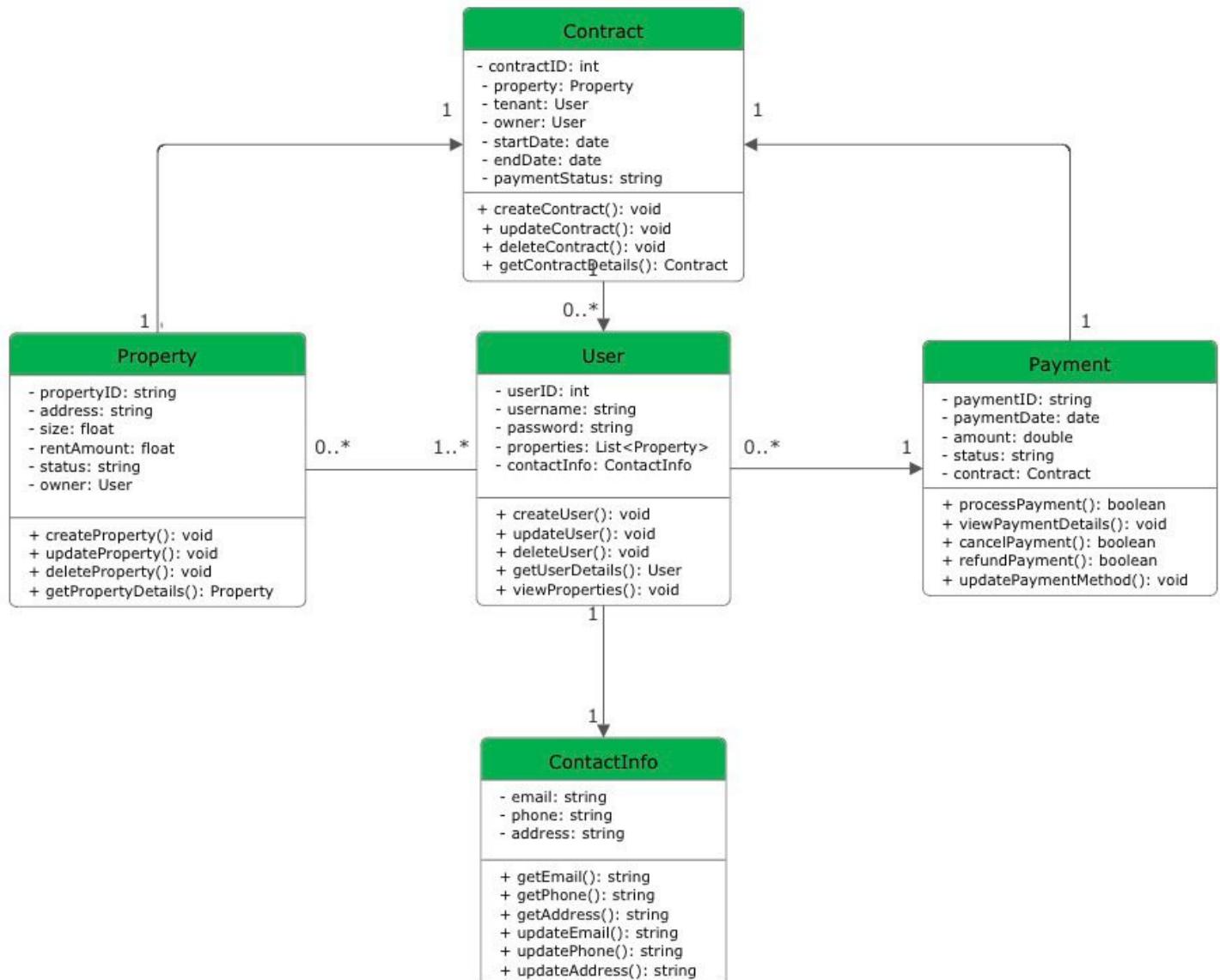


2- Level 1 DFD:



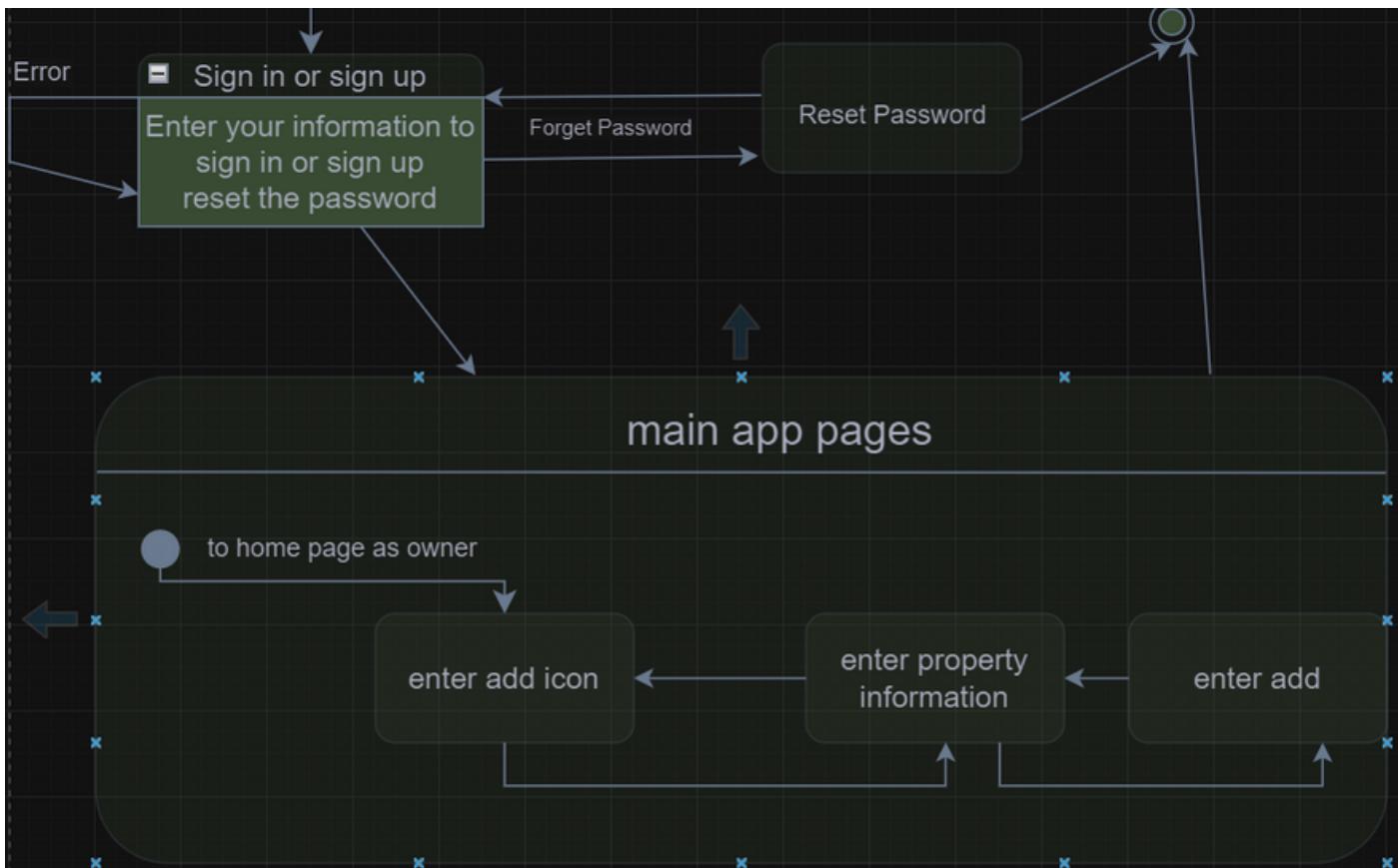
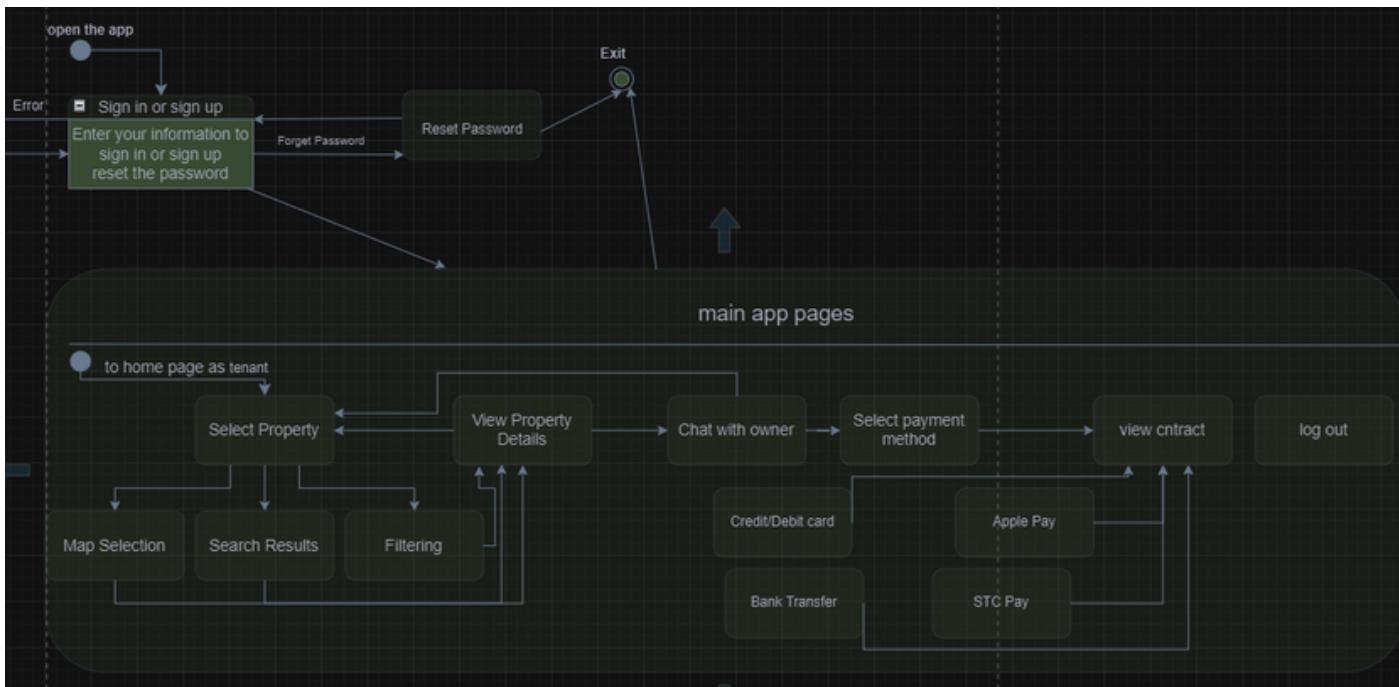
Design Details

5.1. Class Diagrams



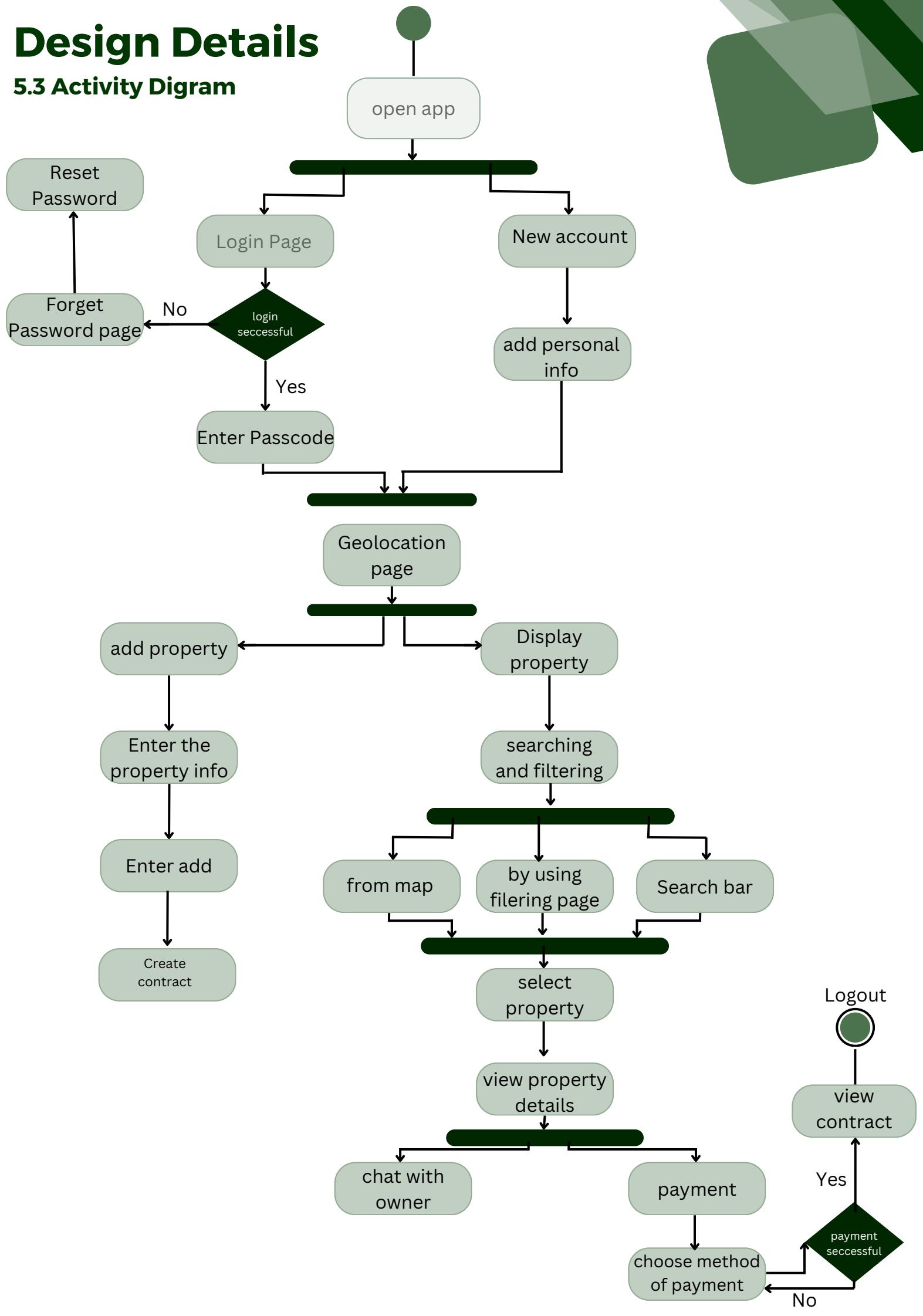
Design Details

5.2. State Daigram



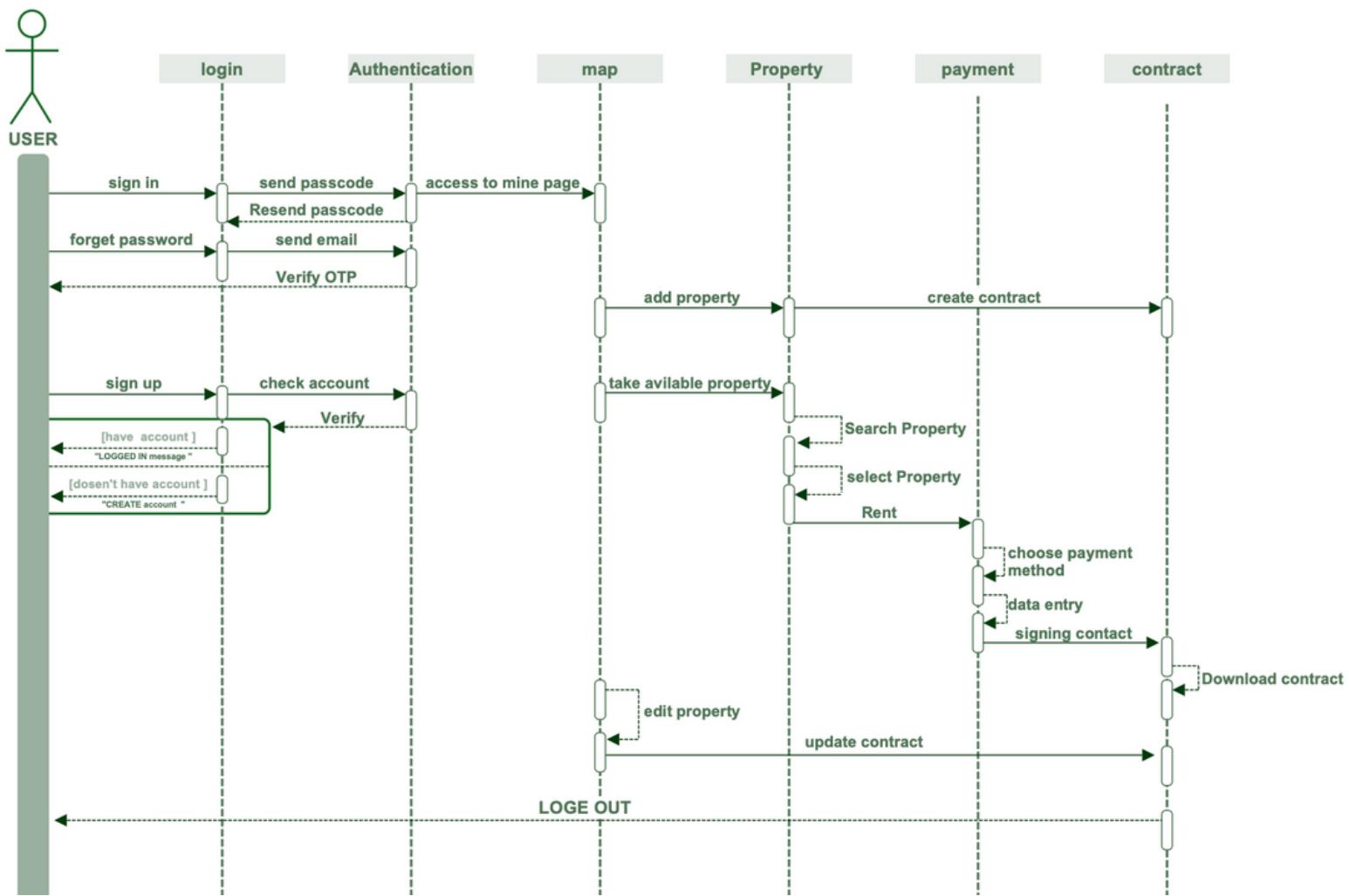
Design Details

5.3 Activity Diagram



Design Details

5.4 Sequence diagram

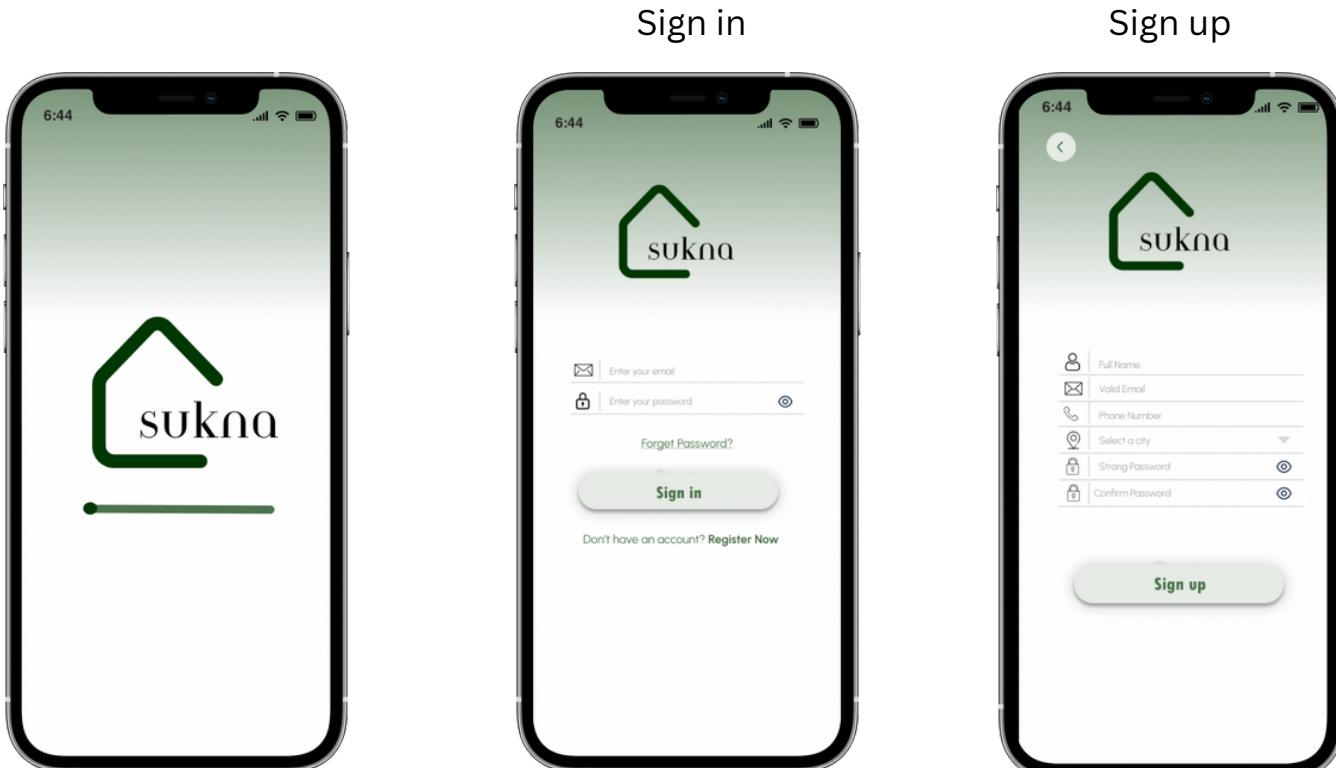


HUMAN INTERFACE DESIGN

6.1. Overview of the User Interface

Sukna's user interface is crafted for simplicity and visual appeal. Users are greeted with a login/register screen, leading to the main screen featuring an interactive map showcasing properties in proximity. Offering flexibility, users can switch between map and list views, aided by a search bar and filter options. The bottom bar hosts the user's profile, private messages for seamless communication with property owners, and a Add New Property button. Direct property searches streamline user experience, and upon selection, a detailed property page emerges. After confirmation, users seamlessly transition to the payment page, where a range of payment methods is presented. This allows users to effortlessly complete transactions. Post-service, tenant can provide feedback and download the contract, ensuring a comprehensive and user-friendly property browsing and management experience.

6.2. Detail Design of User Interface



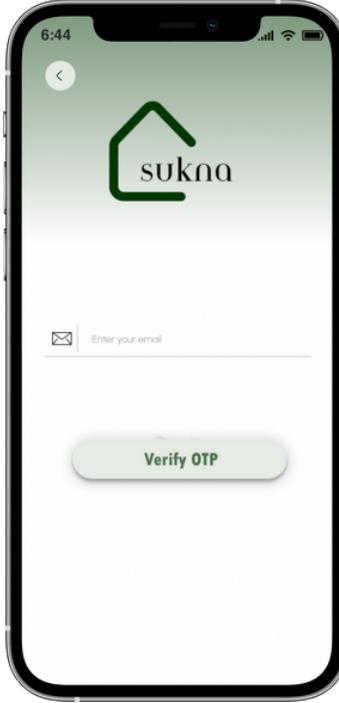
Human Interface Design

6.2. Detail Design of User Interface

Authentication
after sign in



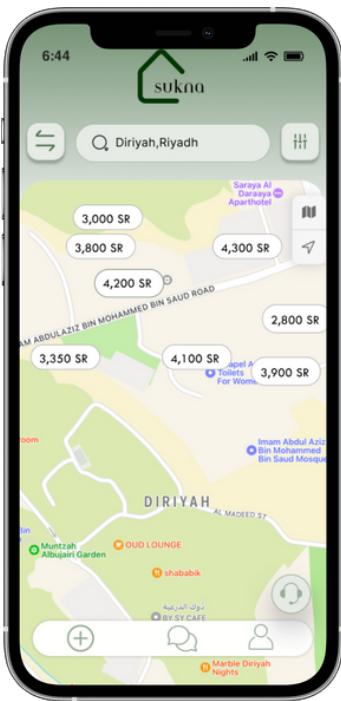
reset password
by entering email



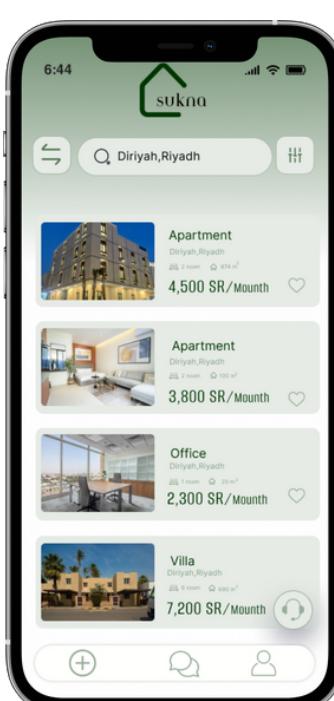
check email



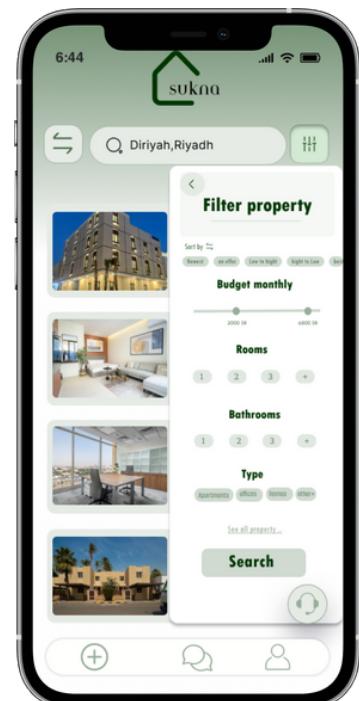
main page



List properties



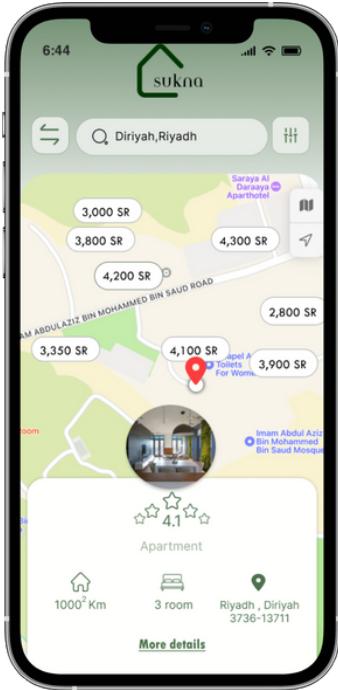
Filter properties



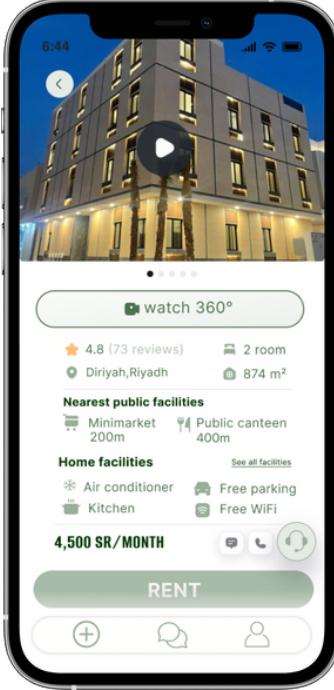
Human Interface Design

6.2. Detail Design of User Interface

select property



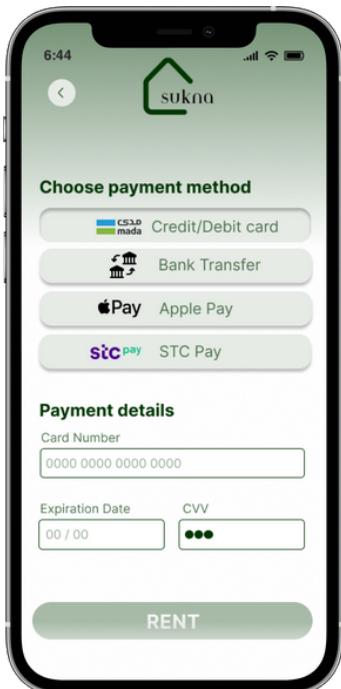
Property details



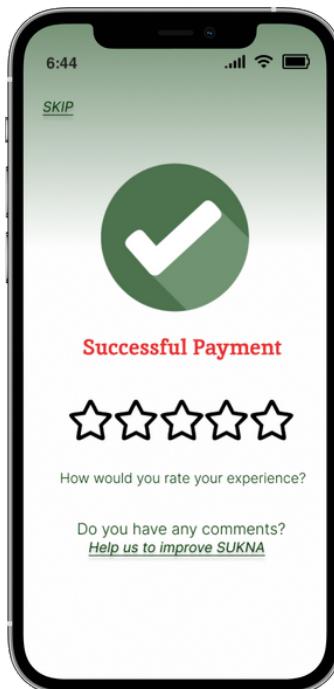
Select payment method



Enter details of payment



Successful payment



Apartment details



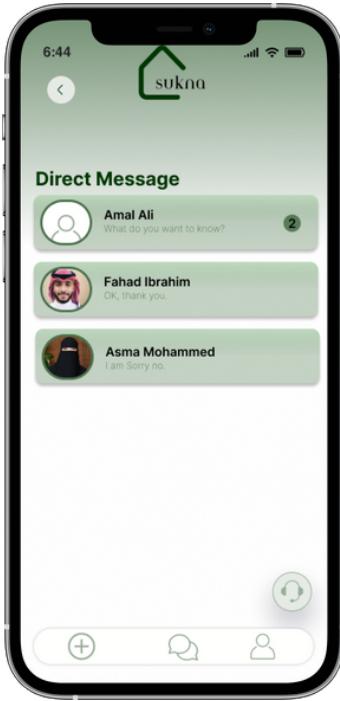
Human Interface Design

6.2. Detail Design of User Interface

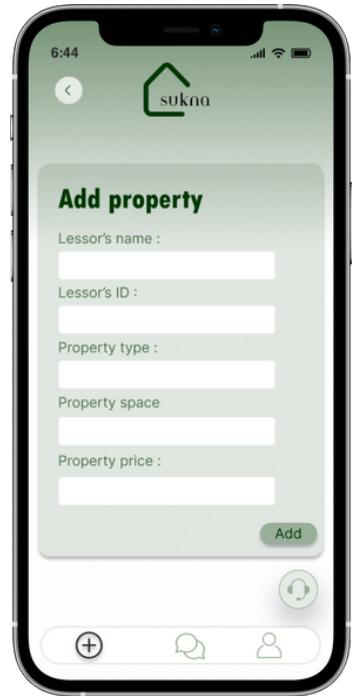
Contract



Chat page



Add property page



Account page



work section

student Names

section work

Reef

Introduction

Leen

System overview

sadeem & rana

**System Architecture and,
Components Design**

Leen & Lena &Raghad

Data structure

**Latifah &sadeem&rana
&Lena**

Data design

**Raghad & rana &reef &
latifah**

Human Interface Design