DESIGNING AND IMPLEMENTING OF MULTI-PLATFORM REAL ESTATE MANAGEMENT SYSTEM

ARIVAN SHAMAL M. ZYAD

QAIWAN INTERNATIONAL UNIERSITY

QAIWAN INTERNATIONAL UNIERSITY 9:16 (Pind. 1/13)

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : Arivan Shamal Zyad

Date of Birth :

Title : DESIGNING AND IMPLEMENTING OF MULTI-PLATFORM REAL ESTATE MANAGEMENT SYSTEM

Academic Session : 2020-2024

I declare that this thesis is classified as:

☐ **CONFIDENTIAL** (Contains confidential information under the Official Secret Act 1972)*

☐ **RESTRICTED** (Contains restricted information as specified by the organization where research was done)*

☑ **OPEN ACCESS** I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE OF STUDENT

QU192SCSJ041
MATRIX NUMBER

Date: 9 MAY 2024

SIGNATURE OF SUPERVISOR

MR. ABDULSALAM ABDULLA
NAME OF SUPERVISOR

Date: 9 MAY 2024

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

"I hereby declare that we have read this thesis and in my
opinion this thesis is suffcient in term of scope and quality for the
award of the degree of Bachelor of Computer Science (Software Engineering)"

Signature	:	
Name of Supervisor	:	MR. ABDUL SALAM ABDULLA
Date	:	9 MAY 2024

DESIGNING AND IMPLEMENTING OF MULTI-PLATFORM REAL ESTATE MANAGEMENT SYSTEM

ARIVAN SHAMAL M. ZYAD

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software Engineering)

School of Computing
Faculty of Engineering
Qaiwan International University

MAY 2024

DECLARATION

I declare that this thesis entitled *"On-Line Recognition of Developing Control Chart Patterns"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : .....

Name : ARIVAN SHAMAL M. ZYAD

Date : 9 MAY 2024

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor, Mr. Abdul Salam Abdulla, for his guidance, support, and encouragement throughout the course of my final year project, "Designing and Implementing of a Multi-Platform Real Estate Management System". His invaluable input and expertise have been instrumental in the successful completion of this project. I would also like to thank the faculty and staff of the Bachelor of Computer Science/Software Engineering program for their support and guidance. Finally, I would like to acknowledge the contribution of my family and friends who have supported me throughout the project. Their love and encouragement have been a source of inspiration and motivation.

ABSTRACT

As a comprehensive real estate management system, I aim to provide an accessible and user-friendly platform for customers to browse, search, and find properties. With a focus on enhancing the customer experience, I will include a mapping system and virtual tour feature to allow for easy navigation and visualization of properties. Both admin and agents will be able to run the system and communicate with customers to manage the properties.

The goal of this project is to develop a system that provides an easy and efficient way for customers to find their desired property, with added features to improve their experience. This solution will be evaluated based on its ability to meet the functional and non-functional requirements, including performance and scalability, user- friendliness, and security. The evaluation will also assess the system's compliance with relevant regulations and standards and its ease of use and accessibility. The expected outcome is a robust and flexible real estate management system that improves the customer experience and the efficiency of property management.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DIMMGS	-	DESIGNING AND IMPLEMENTING OF MULTI-PLATFORM REAL ESTATE MANAGEMENT SYSTEM
SRC	-	Software requirement Specification

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Real estate is a type of real property, which refers to something that is physically tied to a piece of land. It can be used for residential, commercial, or industrial purposes and often includes any water or mineral resources on the land. Real estate is considered the most valuable asset a person can purchase, as it rises in value over time. Therefore, the value of real estate is a leading predictor of the health of an economy.

Real estate significantly affects millions of jobs in home improvement, development, lending, insurance, and business. In addition to homeownership, rental, and property development rates, the value of the real estate is also represented in the homeownership, rental, and property development rates. As we have learned what real estate is and how important it is to everyone, we have observed that in the Kurdistan region, there is a lack of a proper and well-organized agency to control the real estate market, as well as a platform to assist both the customers and the agencies responsible for purchasing and reselling properties. In the Kurdistan region, we must develop each and every industry; by making the real estate industry more user-friendly and straightforward, the market would benefit greatly.

1.2 Problem Background

As we approach the year 2023, we see that technology is improving and making our daily tasks easier; however, the real estate industry in the Kurdistan region still requires improvement. As a customer, it will be quite difficult to discover a house that meets their requirements, as they will need to search on social media or go to each agency and ask for what they are looking for. As we can see, this will take a great deal of time, and if we consider that almost everything can be purchased online or on a platform, for instance, we can see that we can buy anything on Amazon, we can also see that there are no online platforms for real estate in the Kurdistan region, so building one to manage this type of industry and help the market is the most important thing that can be done to help buyers find properties that meet their needs.

1.3 Project Aim

The goal of this project is to create an app that can be used on both mobile and desktop devices, as well as a system that can accommodate multiple users and manage multiple property types and sales channels. The app will also include tools for advertising, virtual tours, and mapping, as well as other features that will improve marketing and customer service while also doing away with outdated methods that are still in use in the Kurdistan region.

1.4 Project Objectives

The objectives of the project are:

- Collect user requirements on the real estate business and the necessary data to construct a solid system to manage everything and add the necessary features so that it benefits both the agency and its clients.

- Plan the development of a multi-platform application system with capabilities such as virtual tours, map locations, and a user-accessible web platform.
- To design and create a prototype that demonstrates the system's functions and appearance based on the collected criteria.
- To implement and test the application system, and then to test the application to determine if it satisfies the specifications and can fulfill the promised functions.

1.5 Project Scope

The scopes of the project are:

- The scope of online property management and real estate management encompasses all the tasks and duties involved in managing rental properties and real estate assets.
- These activities include managing property listings and availability, processing tenant applications and leases, collecting rent payments, and maintaining and repairing properties.
- Online property management systems aim to simplify and automate these processes, providing property owners and managers with a range of tools and features like online listings, application and lease management, payment processing, and maintenance and repair tracking.
- The objective of this project is to create a multi-platform and user-friendly real estate management system that can assist both real estate professionals and clients in the industry.
- The stakeholders in this field include property owners, property managers, tenants, and any other parties with an interest in the management and maintenance of rental properties and real estate assets.
- Property owners and managers are likely to be the primary users of the system, while tenants and other parties may also interact with the system in various ways.

1.6 Project Importance

This Real Estate application is a multi-platform management system for the real estate industry in the Kurdistan region to make it easier to manage, sell, and buy properties, with many features to assist both the agent who will work on the application and the customers who will browse on the application and all users will have multi user accounts. In addition, customers will be greeted with a user-friendly interface and will enjoy a great browsing experience, as there will be a mapping system to display the location of the properties along with a 360-degree view of the house's exterior and a 3D tour of the home, as well as the nearby utilities. As well, the application will include residential projects. Agents will be responsible for adding and managing each property on the system. Customers will browse and select the desired property to reserve and obtain the location of the agency to continue the buying process.

By having this platform manage everything in an easy and simple way, and be able to provide many services and features that will help in marketing the properties and also help customers to find what will fit them best, also this platform will remove the time-consuming and frustrating process of looking for a house that fits customer needs, and helps the agent to have a centralized database, this project is important because it will help the industry and everyone who works in this industry. With this information, we can see that this platform is required in our region to help the market grow further, as the cities are getting larger, the population is increasing, and more properties and residential projects are being constructed; with a well-structured management system, it can be managed and attract agents and customers to use this system.

1.7 Report Organization

This chapter presents the introduction to the proposed project, which includes the introduction, problem context, project purpose, objective, scope, rationale, and significance. In Chapter 2, a literature review of the existing system of real estate applications and systems is presented. This chapter also includes a study of the current

system in Kurdistan, followed by a discussion of the survey's methodology, results, and analysis. In Chapter 3, the methodology used for the system development process and the software and hardware requirements for the Real Estate Application development are discussed. In Chapter 4, the system design for the application is described. Chapter 5 examines the project's conclusion, which includes a summary of PSM 1 accomplishments and a description of PSM2 planning and implementation.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A literature review on online property management and online real estate management would likely include a summary and analysis of existing research and scholarly articles on the topic. The review would likely focus on the benefits and challenges of using online property management systems and other technologies for managing rental properties and other real estate assets.

The literature review might discuss how online property management systems can help to improve the efficiency and effectiveness of managing rental properties and other real estate assets. It could also discuss the various features and tools that these systems provide, such as online listings, application and lease management, payment processing, and maintenance and repair tracking.

The literature review could also discuss the role of online property management systems and other technologies in the broader context of real estate management, including their impact on the industry and on the experiences of property owners, managers, and tenants. It could also discuss the challenges and limitations of these systems, such as security and privacy concerns, the need for training and support, and the potential for technical issues.

Overall, a literature review on online property management and online real estate management would provide a comprehensive overview of the existing research and knowledge on these topics, highlighting the key findings and insights from previous studies.

Generally speaking, a literature review is a curated analysis of existing research pertinent to the proposed system. In other words, it illustrates the process of gathering, receiving, and analysing the necessary information for the system's development. It also provides explanations and justifications that aid in the resolution of research-related issues.

As a real estate management system, my goal is to provide a comprehensive solution for managing rental properties and real estate assets. I aim to collect user requirements on the real estate business and the necessary data to construct a solid system with added features that will benefit both the agency and its clients.

- Basic Information Requirements: The basic information required for real estate management includes the location, type of residence (houses, apartments, etc.), and the details of the property. With the integration of Google Maps, the locations of the properties can be easily visualized and made accessible to customers. This will provide them with a clear understanding of the property's location and surroundings.

- Virtual Tour Feature: One of the key features of my system is the virtual tour feature, which allows customers to experience the property virtually. This feature enhances the customer experience and provides them with a better understanding of the property.

- User-Friendly Interface: The system features a user-friendly interface that allows customers to easily browse, search, and find properties. It includes a mapping system and virtual tour feature to enhance the customer experience and provide them with a comprehensive understanding of the property.

- Communication and Management: The system will be run by both an admin and an agent, who will have the ability to communicate with customers and manage the properties. This feature allows for smooth communication between the agency and its clients, making it easier for them to manage their rental properties and real estate assets.

This project aims to provide a comprehensive solution for the real estate management sector. With the integration of Google Maps and the virtual tour feature, the system offers a user-friendly and efficient way for customers to find their desired property. The system is designed to be run by both an admin and an agent, ensuring smooth communication and management of the properties.

This chapter will cover a number of topics. These comprise a case study, an analysis of the current system in Kurdistan and in general, a review of comparable systems, a comparison with the existing system, the back-end technology, the methodology adopted, and the results of the survey analysis. This chapter will conclude with a quick summary

2.2 Case Study (If any)

Study of domain from general to specific, related studies, a description of the identified problem.

2.2.1 Company Organization Structure

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

2.2.2 Manual Operation

There are several reasons why the current system may be considered bad and in need of replacement. For one, it relies on in-person communication and interactions, which can be limiting for those who live far from the agent's location or who have busy schedules. Additionally, it may not be possible for customers to view every property that meets their criteria, as they must rely on the agent to provide information about available properties.

2.3 Current System Analysis

The current system for buying and selling properties in the Kurdistan region of Iraq is quite outdated and inefficient. Under this system, customers interested in buying or selling a property must visit an agent in person and communicate their specific needs and requirements. If they are able to find a property that meets their needs, they must then set a date to visit the property in person, which can be inconvenient and time-consuming. In addition to this system, there are also several websites and apps that customers can use to search for properties, but these may not be official or authenticated, and may not be supervised or regulated in any way. Some of these websites and apps may also have had negative user experiences, with customers

reporting issues such as incorrect or outdated information, poor user interface or navigation, or difficulty contacting agents or sellers.

This can make it difficult for customers to find their ideal property, as they may not be aware of all the options available to them. The unreliable and potentially negative user experiences of the available websites and apps can also make it difficult for customers to find and purchase the property they are looking for.

Overall, it seems that a more modern and efficient system for buying and selling properties would be beneficial for both customers and agents in the Kurdistan region of Iraq. A system that allows customers to easily search for and view properties

online, and that allows agents to more effectively connect with potential buyers and sellers, would likely be a major improvement over the current system.

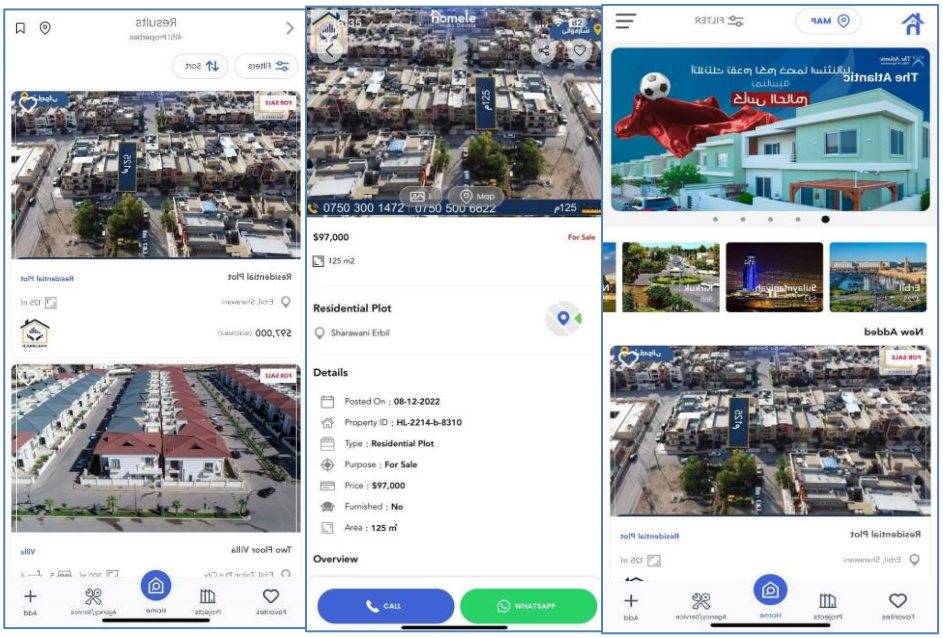


Figure 2.1 Home application Interface

Khanoo is yet another real estate application for selling and purchasing houses. However, this program is unique in that users can list properties for other users, and it also has a feature for finding services such as house cleaning, relocation, and interior design.

The poor UI design and sometimes confusing UI also failure to load data on some properties and lack of modern standard features such as maps and virtual tours caused the majority of users to abandon this application, and it appears to have been abandoned by the developer as many elements of the application are not functioning properly (Khanoo, 2022).

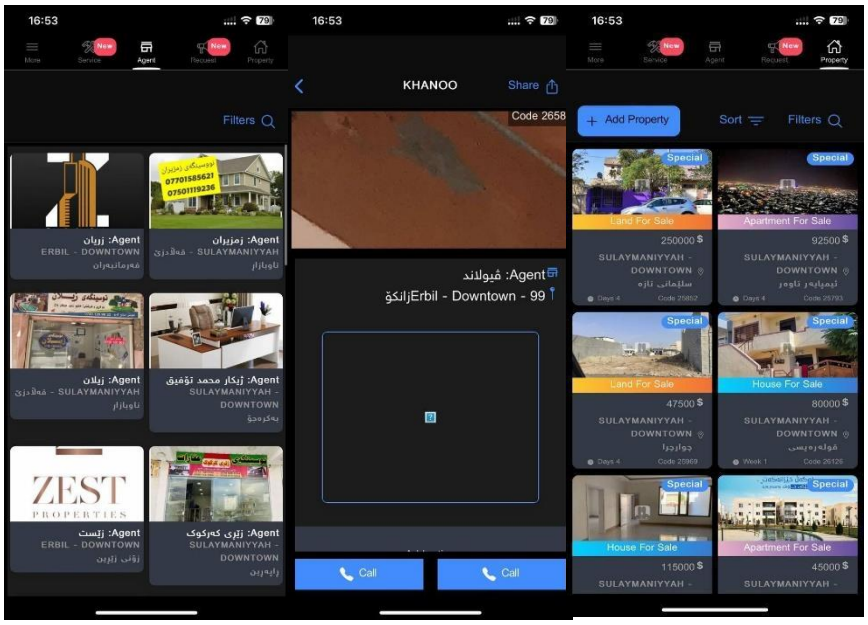


Figure 2.2 Khanoo Application Interface

2.4 Comparison between existing systems

The comparison between the existing systems Homele, Khanoo, and the proposed system.

Table 2.1 Comparison between existing systems

Aspects	Homele	Khanoo	Proposed System (My Real Estate)
Platform Support	IOS, Android, Website	Android, IOS, Web	Windows, Linux, macOS, IOS, Android, Website
User Friendly	Yes	No	Yes
Type of provided Documents	Images, Floor plans	Images	Images, Floor plans, 360 Images
Mapping	Yes	No	Yes
Virtual Tour	No	No	Yes
Multi User Type	No	No	Yes
Searching and Filtering	Yes	Yes	Yes

Modern UI	Yes	No	Yes
Well Presented UI	Yes	No	Yes
User Support	No	No	Yes
Reservation	No	No	Yes
Review on properties added by users	Yes	No	Yes
Focused Area	Irbil	Irbil	All Kurdistan

According to the comparison in Table -, the suggested system will be available on all platforms with a very user-friendly UI and photos, 360 images, and floor plans of every property that the agents will contribute. The application will be divided into two parts: the agent panel and the user panel. The agent panel will be in charge of listing the properties and managing them, as well as reviewing user-added properties. The user panel will be in charge of showcasing the properties. It will also provide a mapping system so that users can search on the map for a described location in different cities, as well as virtual tours for users to use to get a clear vision of the

property. This application will include a fluid and modern UI aspect, as well as the option to reserve for the listed properties if it is permitted on the property. The properties contributed by users will be visited and confirmed by agents to prevent fake property listings, and this project's focus is on the entire Kurdistan region. Finally, this program will provide user support in the form of instruction and assistance if any user-related issues arise.

2.5 Literature Review of Technology Used

Agile Scrum will be the preferred methodology for this project after a thorough investigation of several project management approaches.

Agile methodology is a set of principles and practices for software development that emphasizes collaboration, flexibility, and continuous improvement. It is based on the Agile Manifesto, a set of principles that were developed by a group of software developers in 2001. Agile methodology is often used in agile software development, a type of project management that is designed to be adaptive and responsive to change.

Scrum is a type of agile methodology that is commonly used in software development. It is a framework that provides a set of practices, roles, and artifacts for managing and completing complex projects. In Scrum, teams work in short, iterative cycles called sprints, which typically last one to four weeks. At the end of each sprint, the team reviews the work that was completed and makes adjustments based on feedback and changing requirements.

The main advantage of agile methodology and Scrum is that they enable teams to deliver high-quality software quickly and efficiently, by breaking large projects down into smaller, manageable chunks and by continuously refining and improving the software as it is being developed. Agile methodology and Scrum also emphasize collaboration and transparency, which can improve communication and team cohesion.

However, there are also some disadvantages to agile methodology and Scrum. One potential disadvantage is that they can be difficult to implement and require a high level of discipline and commitment from the team. Another potential disadvantage is that they can be challenging to manage when the scope or requirements (Hayat et al., 2019).

Agile Scrum will be the preferred methodology for this project after a thorough investigation of several project management approaches.

Agile methodology is a set of principles and practices for software development that emphasizes collaboration, flexibility, and continuous improvement. It is based on the Agile Manifesto, a set of principles that were developed by a group of software developers in 2001. Agile methodology is often used in agile software development, a type of project management that is designed to be adaptive and responsive to change.

Scrum is a type of agile methodology that is commonly used in software development. It is a framework that provides a set of practices, roles, and artifacts for managing and completing complex projects. In Scrum, teams work in short, iterative cycles called sprints, which typically last one to four weeks. At the end of each sprint, the team reviews the work that was completed and makes adjustments based on feedback and changing requirements.

The main advantage of agile methodology and Scrum is that they enable teams to deliver high-quality software quickly and efficiently, by breaking large projects down into smaller, manageable chunks and by continuously refining and improving the software as it is being developed. Agile methodology and Scrum also emphasize collaboration and transparency, which can improve communication and team cohesion.

However, there are also some disadvantages to agile methodology and Scrum. One potential disadvantage is that they can be difficult to implement and require a high level of discipline and commitment from the team. Another potential disadvantage is that they can be challenging to manage when the scope or requirements (Hayat et al., 2019).

This section will talk about the technologies and tools that will be used in this project:

Dart & Flutter Dart is a programming language developed by Google. It is an object-oriented language that is designed to be easy to learn and use, with a focus on productivity and flexibility. Dart is used for building web, server, and mobile applications (Flutter b, 2020).

Flutter is a mobile app development framework created by Google. It is built on the Dart programming language and uses the same language for both the front-end and back-end of an app. Flutter allows developers to create natively-compiled, cross-platform apps for Android and iOS using a single codebase. It includes a rich set of pre-designed widgets and tools that make it easy to build beautiful, high-performance apps quickly and efficiently (Darji, 2021).

In summary, Dart is a programming language used for building a variety of applications, while Flutter is a framework for building mobile apps using Dart. Together, they provide a powerful toolkit for creating cross-platform mobile apps.

Firebase is a Google platform that offers cloud-based services for building and managing web and mobile apps. It simplifies development tasks and provides features like firebase authentication, firestore, and firebase storage.

Firebase Authentication is a service that makes it easy to add user login and registration to your applications. It supports various authentication methods, including email/password and social logins, simplifying the authentication process for developers.

Firestore is a NoSQL document database provided by Firebase. It organizes data into collections and documents and offers real-time data synchronization, powerful querying, and automatic scalability. It simplifies data storage and retrieval for web and mobile applications.

Firebase Storage is a cloud-based storage service that allows developers to securely store and serve user-generated files such as images and videos. It provides an easy API for file management and integrates well with other Firebase services.

VS Code Visual Studio Code is a popular source code editor developed by Microsoft. It is a free and open-source tool that is available for Windows, macOS, and Linux. Visual Studio Code is designed to be a lightweight, yet powerful code editor that can be easily customized to fit the needs of individual developers.

One of the key features of Visual Studio Code is its support for a wide variety of programming languages and technologies. It includes built-in support for debugging, source control, and a range of other tools, making it a versatile and powerful tool for developers. Additionally, Visual Studio Code has a large and active

community of users and developers who have created a wide range of extensions and plugins that can be used to further extend the capabilities of the editor.

In summary, Visual Studio Code is a popular code editor that is widely used by developers for a variety of programming tasks. It is known for its flexibility, extensibility, and support for a wide range of languages and technologies (Heller, 2022).

Google Maps API: is a tool provided by Google that enables developers to embed Google Maps into their own websites and mobile applications. The API provides a range of features, including the ability to display maps, search for places, and obtain directions between locations. The API uses a REST interface, which allows developers to make HTTP requests to retrieve information from the API (Google 2021).

When it comes to using the Google Maps API with Flutter, there are a few different options available. Flutter is an open-source mobile app development framework that is used to build high-performance, high-fidelity, and visually appealing applications for iOS and Android. To use the Google Maps API with Flutter,

the optimal choice is ‘google_maps_flutter’ package, which is a Flutter plugin that provides the necessary bindings to the Google Maps API.

The ‘google_maps_flutter’ package makes it easy to display maps within Flutter applications and provides a number of features, including the ability to add markers, draw shapes, and customize the appearance of the map. In addition, the package also provides access to the Google Places API, which enables developers to search for places, such as restaurants or businesses, and retrieve information about them.

In summary, the Google Maps API provides a powerful tool for developers to embed maps into their applications and websites. When used in combination with Flutter, the ‘google_maps_flutter’ package provides a comprehensive solution for building high-quality, interactive maps within mobile apps (Google Maps Flutter

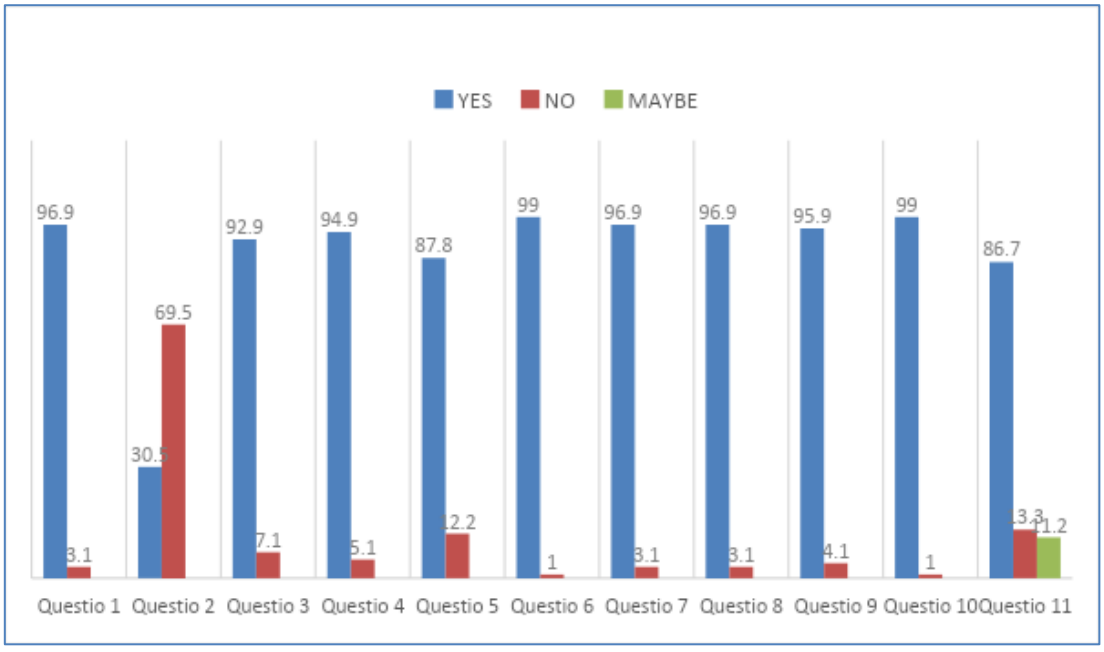


Figure 2.3 Result of Survey Questions

2.6 Chapter Summary

In this chapter, the inter-organization case study was stated. The current system analysis and the comparison between existing systems are also clearly discussed. Next, the technology and tools used to develop the system. Lastly, this chapter ends with the result and analysis of the survey are also well-identified.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

This chapter describes the chosen methodology implemented in developing My Real Estate Application. It is essential in choosing a suitable methodology based on the project as it will define how the project development process will be delivered starting with the first phase up until a working system is produced.

Software development methodology refers to the approach or process that a team follows when developing a software product. There are several different software development methodologies that have been developed over the years, and each has its own unique set of principles, practices, and tools.

One of the main reasons why software development methodology is important is that it helps to ensure that the development process is organized and efficient. By following a defined methodology, teams can plan, track, and execute their work in a systematic and repeatable way. This can help to reduce the risk of errors and delays and increase the chances of delivering a high-quality product on time. In addition to improving efficiency and quality, software development methodology can also help to improve communication and collaboration within a team. By following a consistent process and using common tools and terminology, team members can work together more effectively and avoid misunderstandings or miscommunications.

Overall, software development methodology is an essential aspect of software development because it helps to ensure that the process is well-organized, efficient,

and effective. By following a well-defined methodology, teams can produce higher- quality products more efficiently and with fewer delays or errors.

3.2 Methodology Choice and Justification

Agile Scrum is chosen as the most suitable development methodology for My Real Estate Application, one of the main benefits of Scrum is its focus on delivering working software quickly and efficiently. By breaking down work into small, incremental units (called "sprints") and delivering new functionality at the end of each sprint, Scrum allows teams to deliver high-quality products quickly and efficiently. In addition to its focus on speed and efficiency, Scrum also emphasizes collaboration and transparency. The Scrum framework includes regular meetings, such as daily stand-ups, sprint planning sessions, and retrospectives, which help team members stay informed and aligned on their work. Overall scrum is a popular choice for agile software development teams because it helps them to deliver high-quality products quickly and efficiently, while also fostering collaboration and transparency within the team. Some of the benefits of using Agile Scrum as a software development methodology:

- Faster delivery: By breaking down work into small, incremental units (called "sprints") and delivering new functionality at the end of each sprint, Agile Scrum allows teams to deliver high-quality products quickly and efficiently.

- Improved collaboration: The Scrum framework emphasizes collaboration and transparency, with regular meetings and a focus on teamwork. This can help to improve communication and coordination within the team.

- Enhanced flexibility: Agile Scrum is designed to be flexible and responsive to change, allowing teams to quickly adapt to evolving customer needs and requirements.

- Increased customer satisfaction: By incorporating customer feedback into the development process and delivering working software at the end of each sprint, Agile Scrum can help to increase customer satisfaction and build trust.

- Better visibility: The Scrum framework includes regular review and retrospective meetings, which provide visibility into the progress of the project and help teams identify areas for improvement.

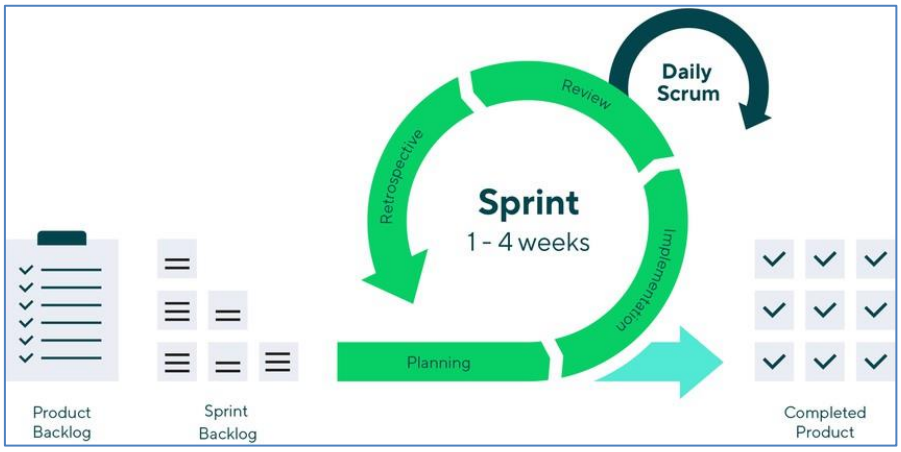


Figure 3.1 Agile scrum

3.3 Phases of the Chosen Methodology

Agile Scrum is chosen as the most suitable process model to develop My Real Estate Application. This is because agile scrum is a popular software development methodology that is based on the principles of agile software development. Agile development emphasizes flexibility, collaboration, and continuous delivery, and Scrum is a framework for implementing these principles in a structured and disciplined way. The Software Development Life Cycle (SDLC) is the process that a team follows when developing a software product, from initial planning to final deployment. Agile Scrum is a software development methodology that follows a specific SDLC, known as the Scrum development cycle. The Scrum development cycle consists of the following steps:

1. Planning: The team defines the goals and scope of the project, creates a product roadmap, and identifies the user stories that will be included in the first sprint.

2. Development: The team works on the user stories identified in the planning phase, using short iterations (called "sprints") to deliver new functionality.

3. Review: At the end of each sprint, the team reviews the work that has been completed and solicits feedback from stakeholders.

4. Retrospective: After each sprint, the team holds a retrospective meeting to review their process and identify areas for improvement.

5. Release: When the team has completed the necessary user stories and the product is ready for release, it is deployed to production.

This procedure is performed for each sprint until the completion of the project. Overall, the Scrum development cycle is intended to be adaptable and responsive to change, enabling teams to rapidly deliver high-quality products and respond to changing client requirements.

3.4 Technology Used Description

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks.

3.5 System Requirement Analysis

When planning a software project, it is essential to consider hardware and software requirements, as they might affect the performance and usefulness of the application or system. Ensuring that the hardware and software requirements are correctly recognized and met can contribute to the application's or system's efficient operation.

3.6 Chapter Summary

This chapter provides an in-depth analysis of the scrum approach used to construct My Real Estate Application. Scrum is a methodology for managing and completing complex projects, especially software development projects, by utilizing iterative, incremental work cycles and frequent reviews and revisions. In this chapter, specifications for hardware and software are also provided.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

Requirement analysis and design is a critical step in the software development life cycle (SDLC) and plays a key role in the success of any software engineering project. It involves gathering, analyzing, and specifying the requirements for a system, and designing a solution that meets those requirements. The requirement analysis and design process can help ensure that the resulting system is well-suited to the needs of the stakeholders and is feasible to implement within the available resources and constraints. A thorough and well-planned requirement analysis and design process can help ensure that the project is completed on time, within budget, and meets the quality standards of the stakeholders.

4.2 Requirement Analysis

This section aims to provide a comprehensive overview of the functional and non-functional requirements, as-is and to-be business processes, use case diagram, sequence diagram, and activity diagram that are necessary for the successful implementation of the system. The functional requirements outline the specific actions and tasks that the system should be able to perform, while the non-functional requirements describe the overall qualities and characteristics that the system should possess. The as-is business processes describe the current state of the system, while the to-be business processes outline the desired state after the implementation of the new system. The use case diagram, sequence diagram, and activity diagram provide visual representations of the interactions between the system and its users, as well as the flow of activities and processes within the system. All of these elements are

essential for ensuring that the system meets the needs of the stakeholders and is able to be implemented in a feasible and cost-effective manner.

Functional requirements describe the specific actions and tasks that a system should be able to perform. They outline the functionality that the system needs to have in order to meet the needs of the stakeholders and users.

- Login and registration: The system should allow users to create an account and log in to the system. This might include features such as email verification and password recovery.

- Search and browse: The system should allow users to search and browse through listings of houses and apartments for sale, using criteria such as location, price, or the number of bedrooms.

- View details: The system should allow users to view detailed information about individual listings, including photos, descriptions, and virtual tours.

- Map view: The system should allow users to view listings on a map, allowing them to see the location of each property.

- Virtual tours: The system should allow users to view virtual tours of properties, allowing them to get a sense of the layout and features of each property.

- Residential projects: The system should allow users to browse and view information about residential projects, including details such as location, amenities, and pricing.

- Upload and add data: The system should allow agents to upload and add data to the system, including information about new listings and updates to existing listings.

- Manage users: The system should allow the admin to manage users, including the ability to add, delete, or edit user accounts.

- Manage properties: The system should allow the admin to manage listings, including the ability to add, delete, or edit listings.

Non-functional requirements describe the overall qualities and characteristics that a system should possess. They outline the non-functional aspects of the system that are necessary for it to be successful.

- Performance: The system should be able to handle a high volume of users and listings without experiencing delays or performance issues. This might involve requirements such as fast page load times and the ability to handle large numbers of search queries.

- Scalability: The system should be able to scale up or down as needed, in order to handle changes in user demand or the number of listings.

- Security: The system should be secure, protecting sensitive data such as user login credentials and financial information. This might involve requirements such as encrypted data storage and secure communication channels.

- Usability: The system should be easy to use, with a user-friendly interface and intuitive navigation. This might involve requirements such as clear labeling, consistent layout, and responsive design.

- Accessibility: The system should be accessible to users with disabilities, in accordance with relevant laws and guidelines. This might involve requirements such as support for screen readers and high-contrast modes.

- Compatibility: The system should be compatible with a range of devices and browsers, allowing users to access it from a variety of platforms.

- Maintainability: The system should be easy to maintain and update over time, allowing for changes or enhancements to be made as needed.

A use case is a description of how a system interacts with its actors to achieve a specific goal or set of goals. In the context of a real estate application system. The use case diagram provided represents the interactions between these actors and the system. The lines connecting the actors to the system represent the interactions, and the oval shapes represent the use cases.

The Figure 4.1 is a use case diagram that represents the different user roles and their respective functions within the system. There are three main actors in the system: user, agent, and admin. Each of these actors has unique use cases associated with their role.

4.3 Project Design

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

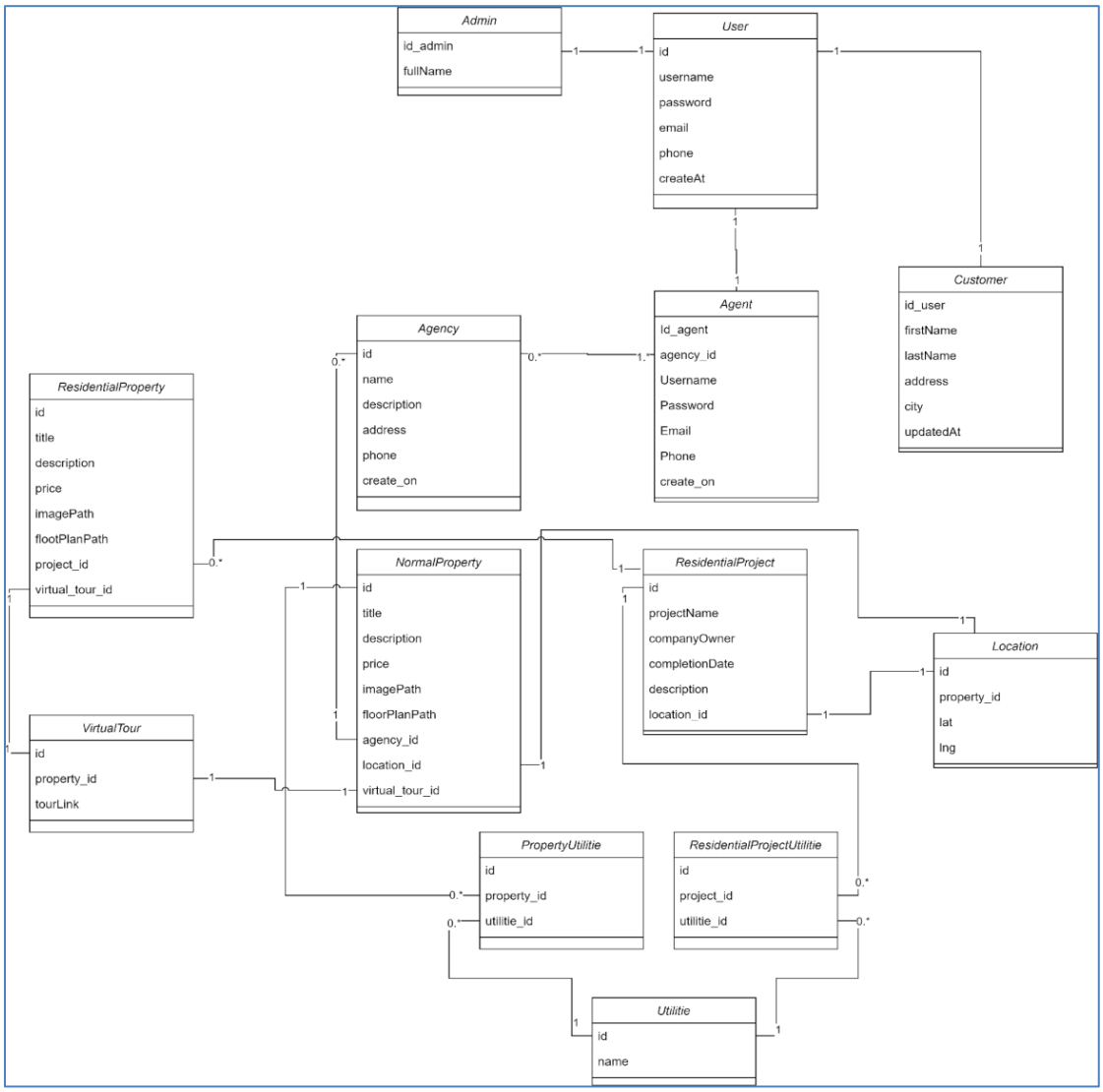


Figure 4.1 class diagram

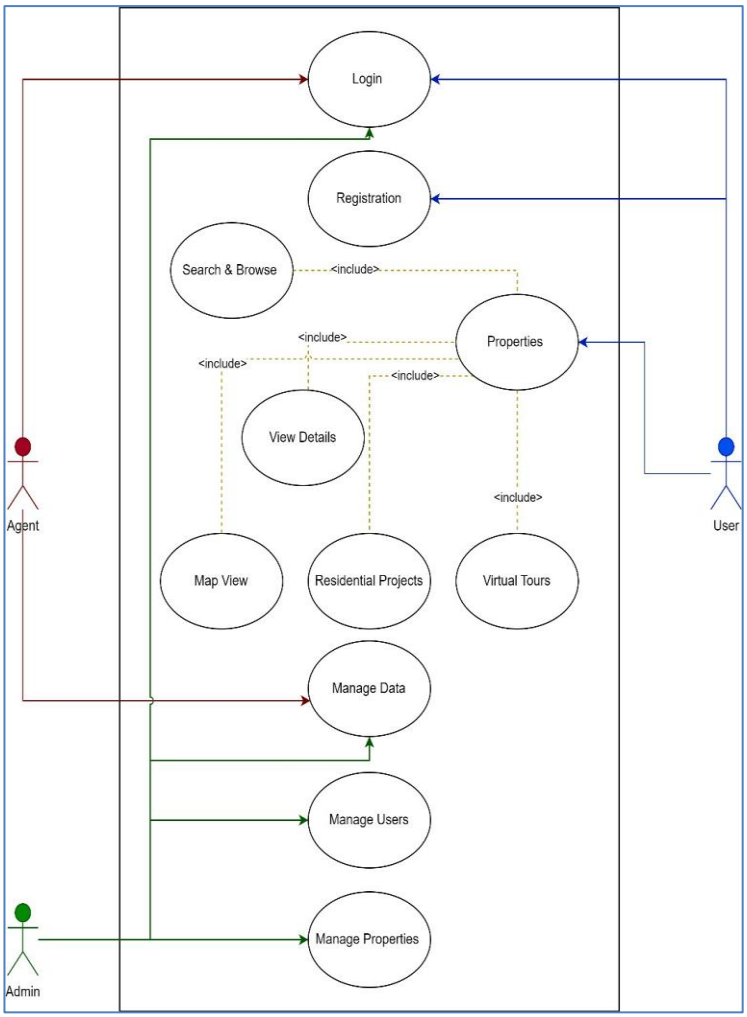


Figure 4.2 Use case diagram

4.4 Database Design

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document.

4.5 Interface Design

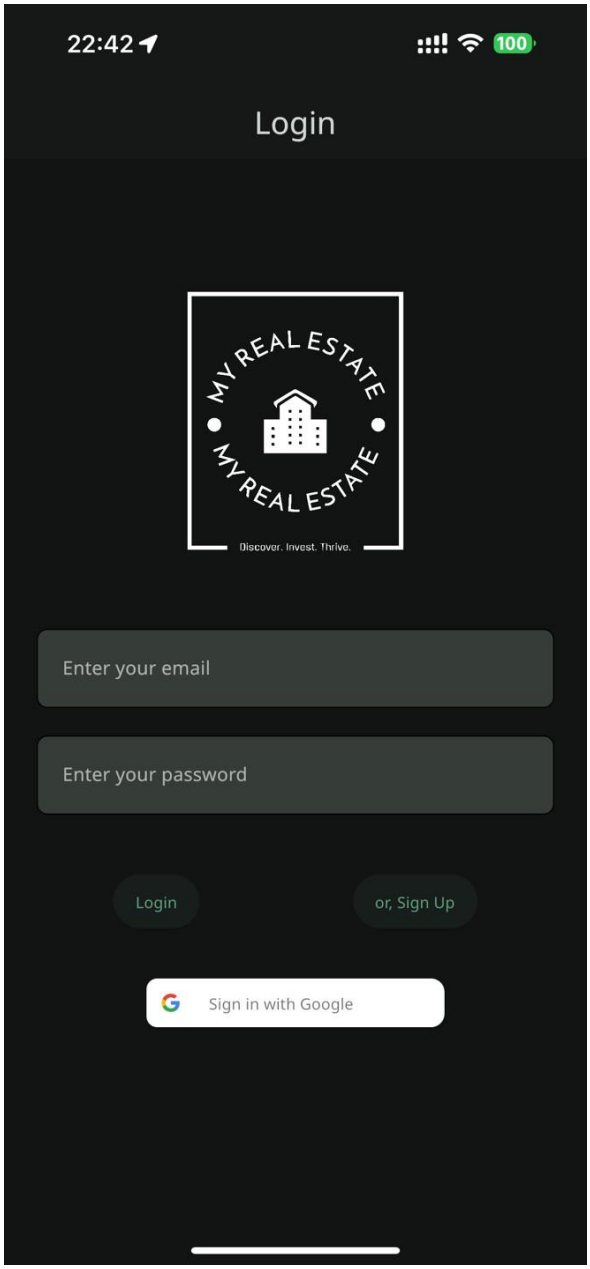


Figure 4-5 Login Interface

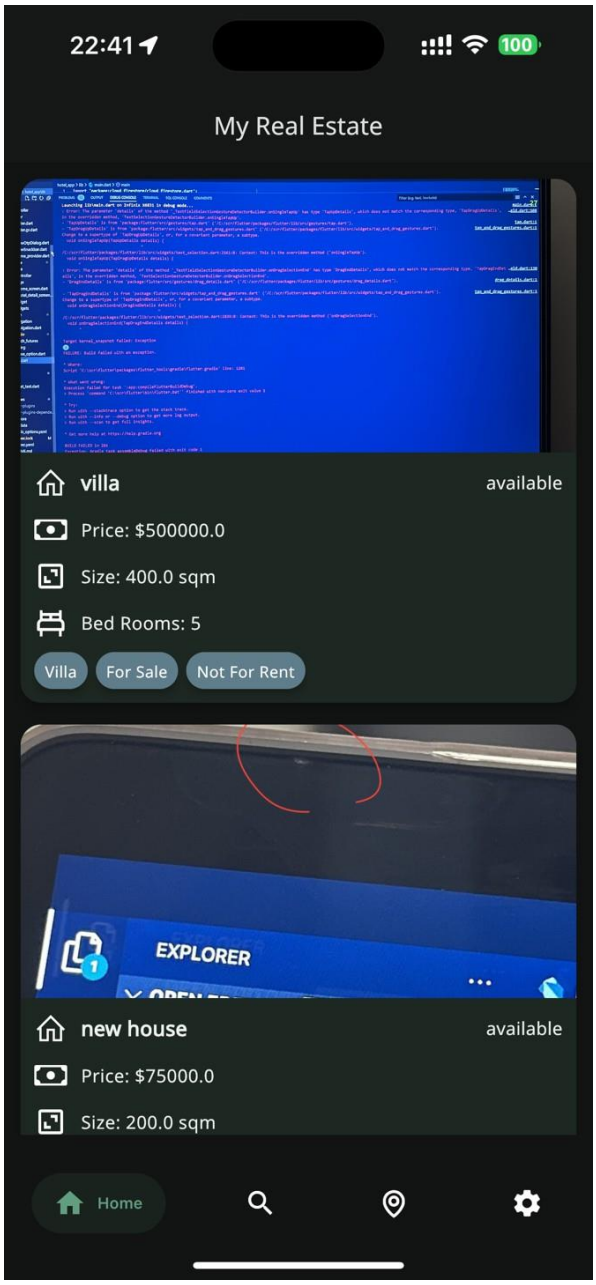


Figure 4-6 Home Page Interface

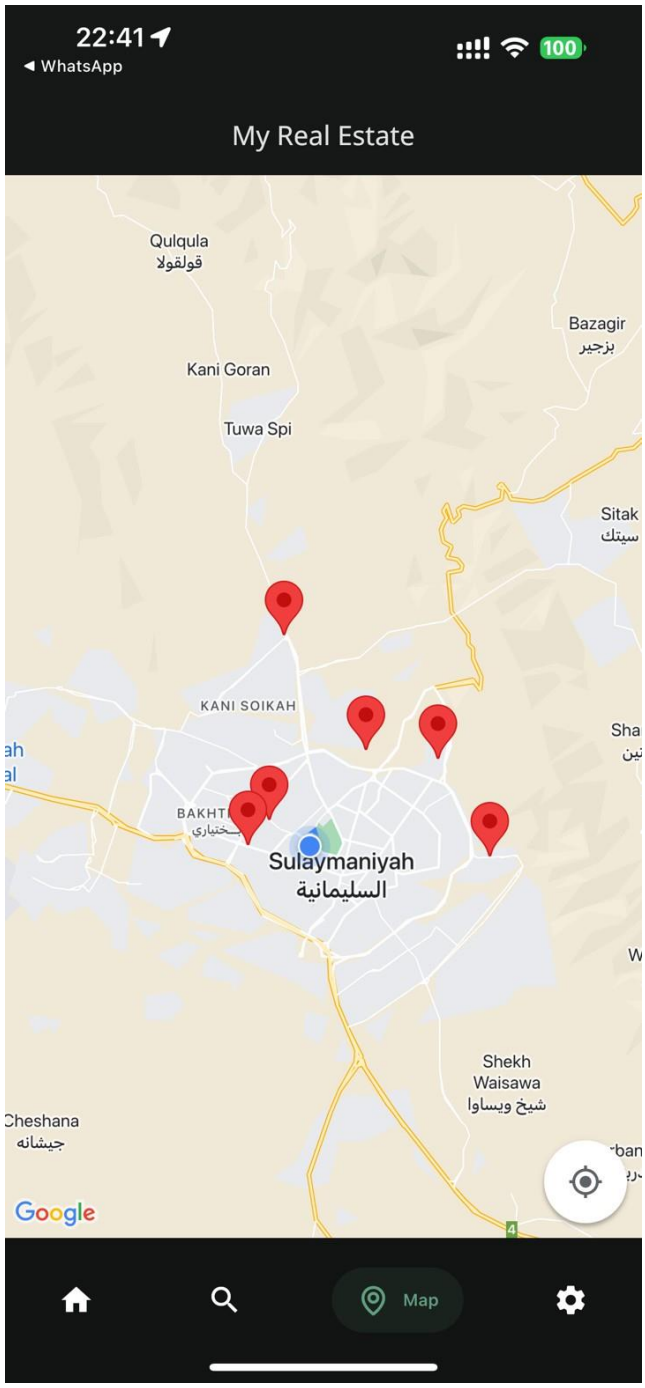


Figure 4-7 Map view Interface

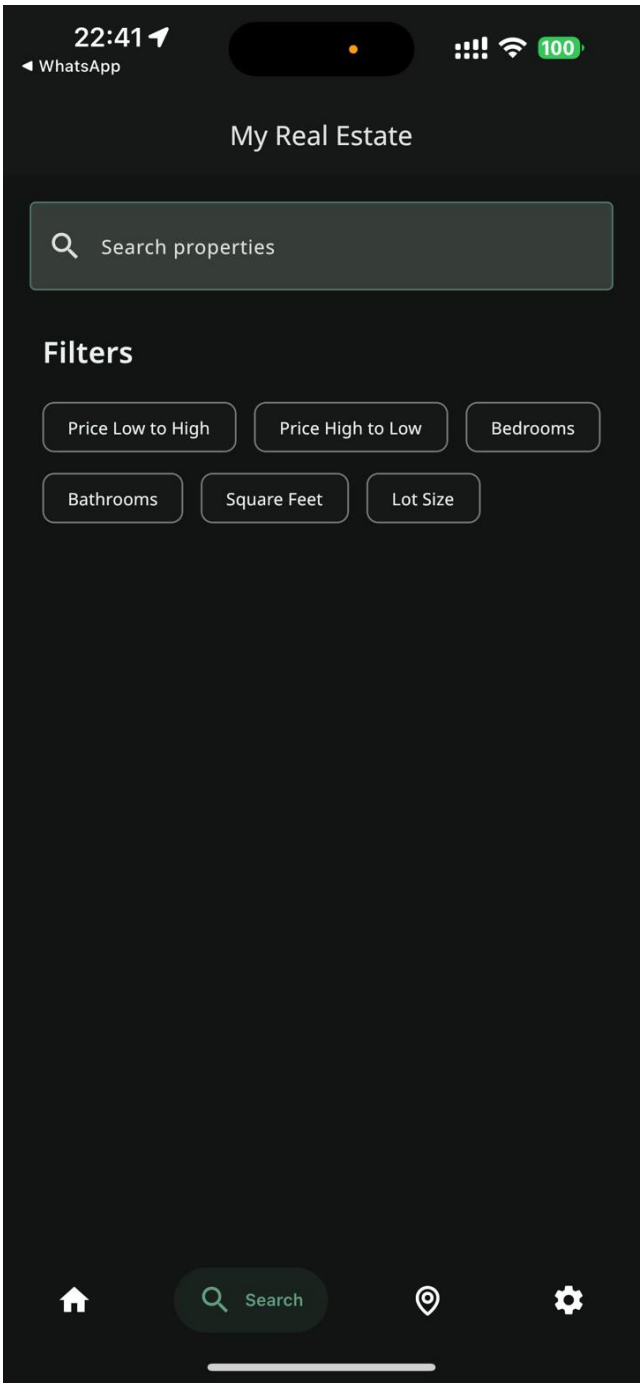


Figure 4-8 Search Interface

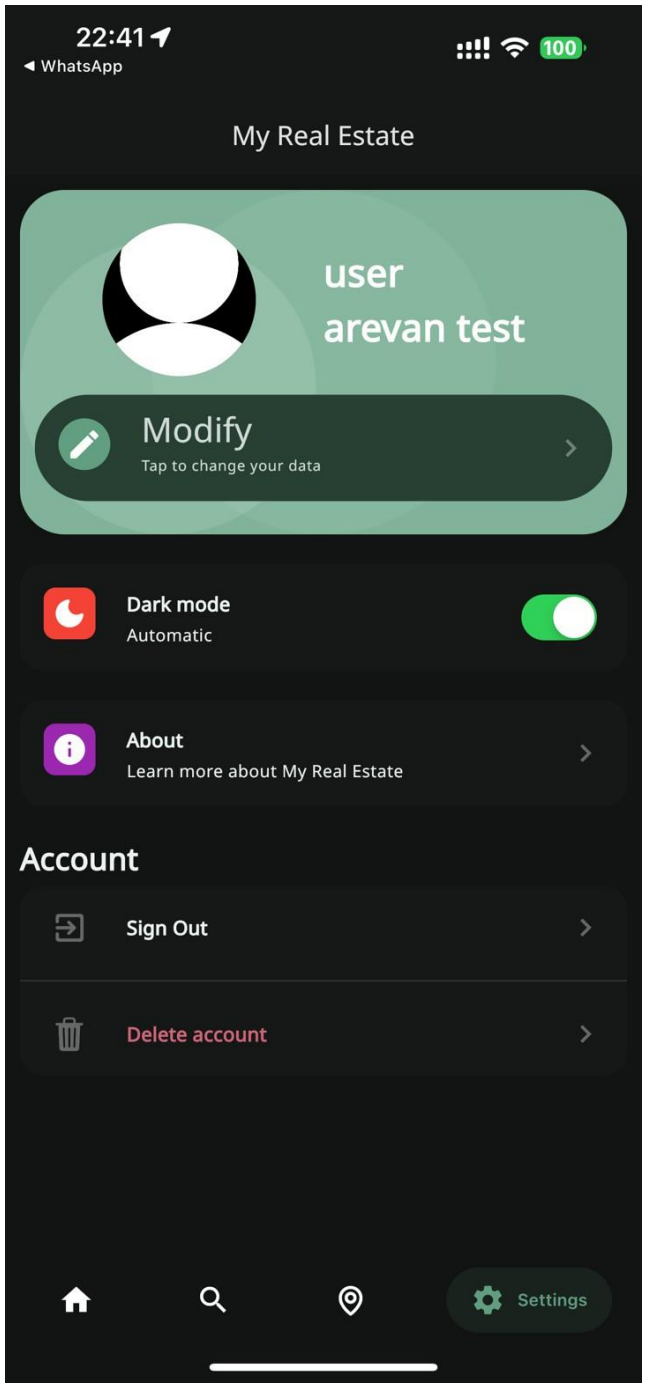


Figure 4-9 Settings

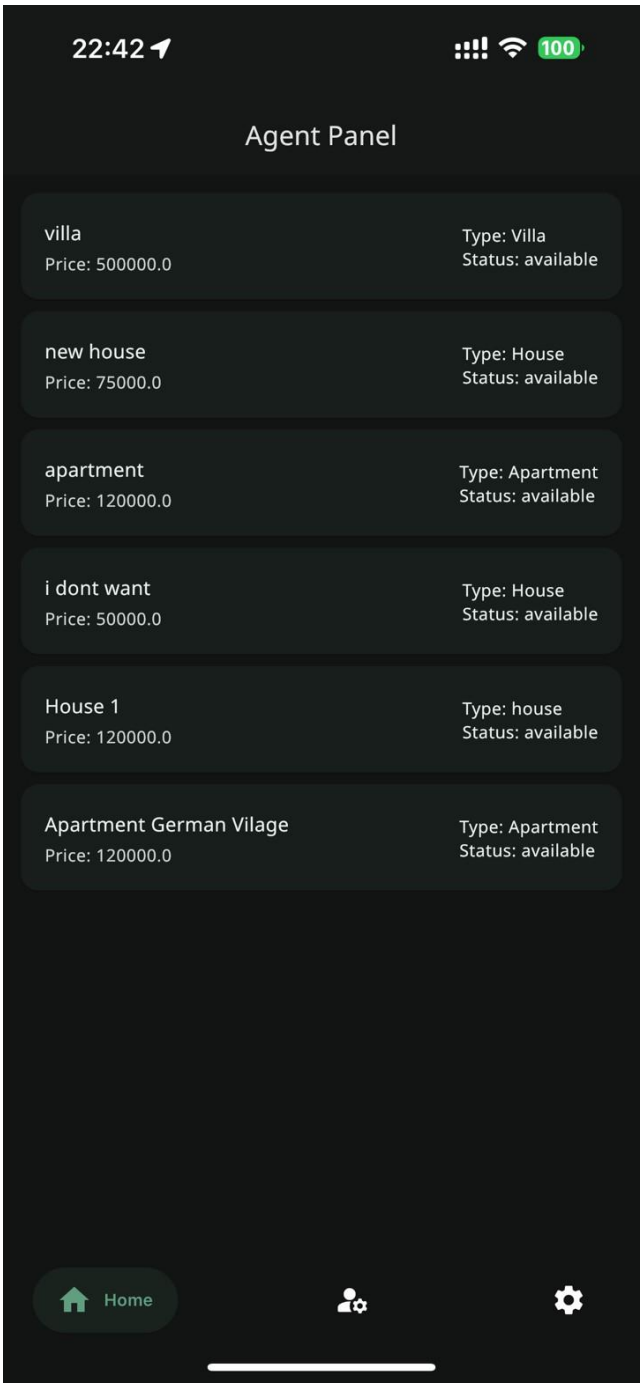


Figure 4-10 Agent Home Interface

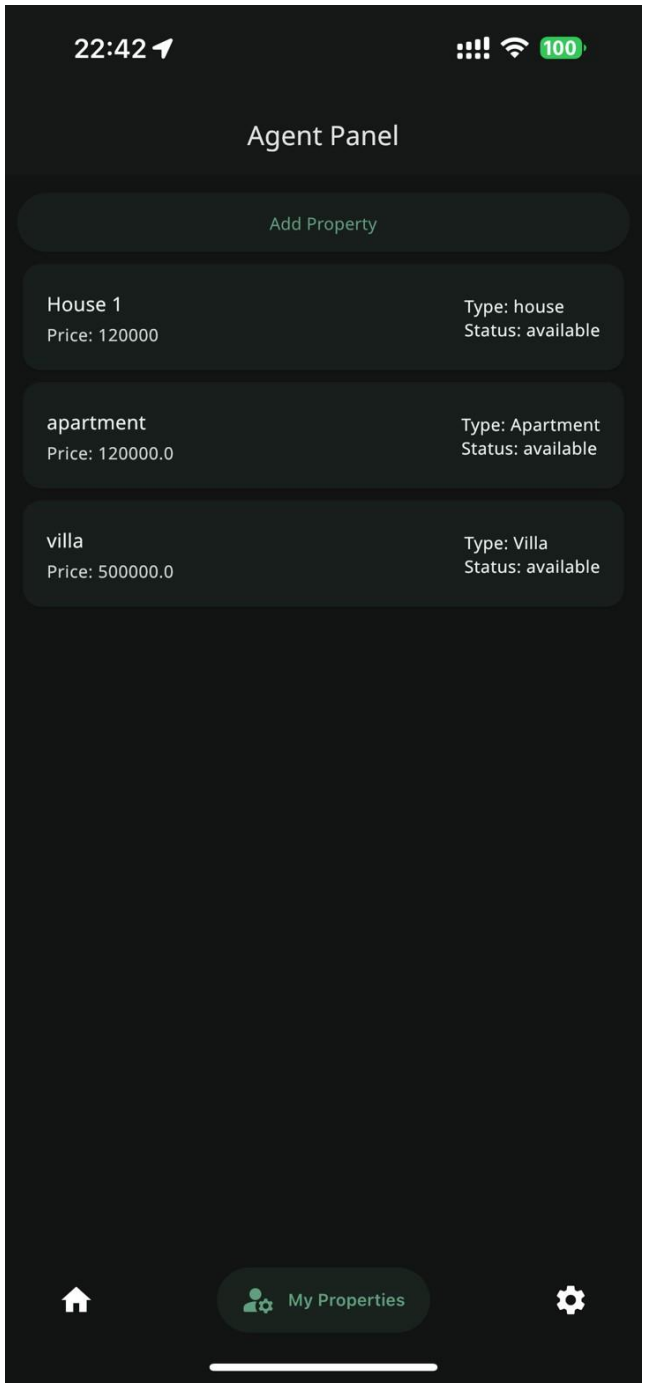


Figure 4-11 Agent Properties Interface

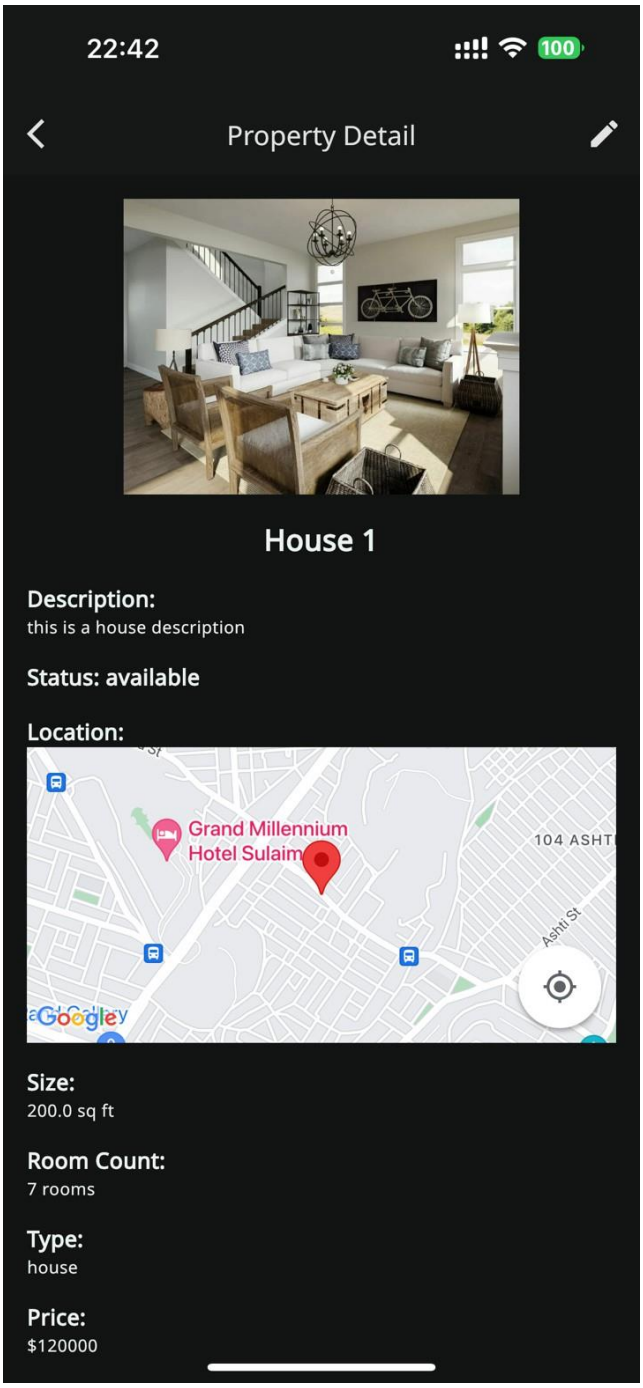


Figure 4-12 Agent View property Interface

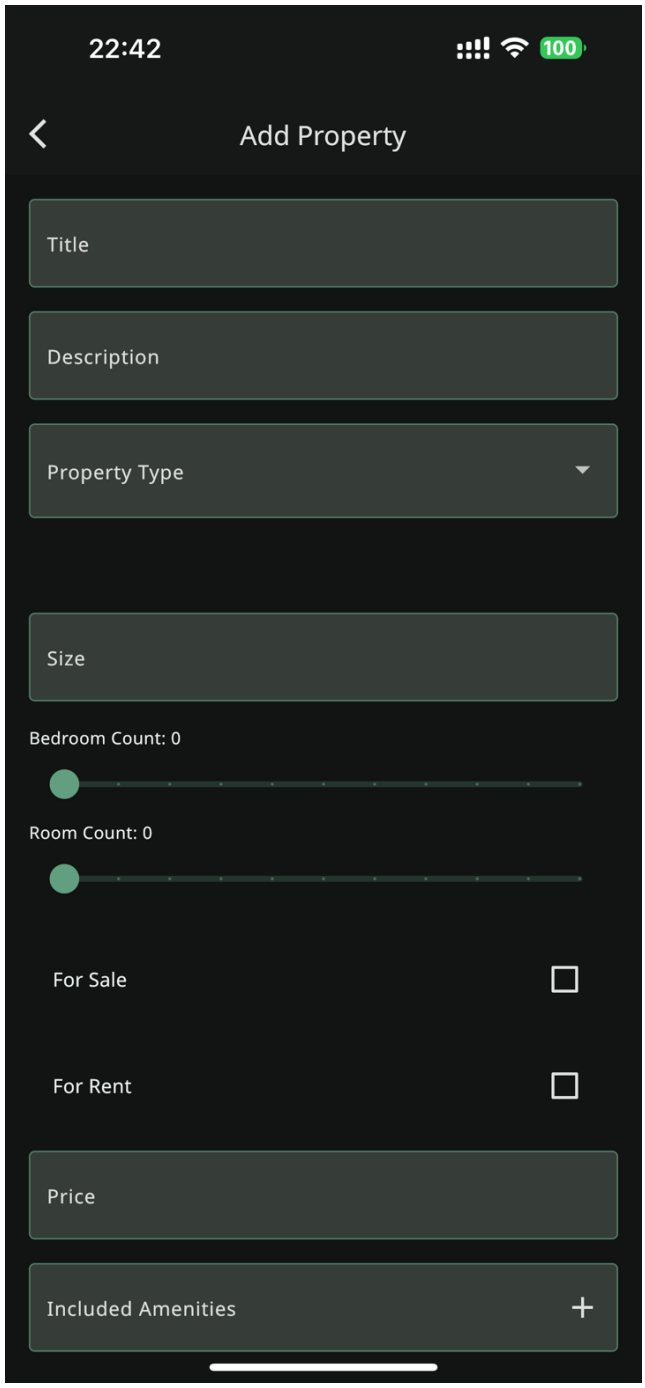


Figure 4-13 Agent Add Interface

Figure 4-14 Agent Select Location Interface

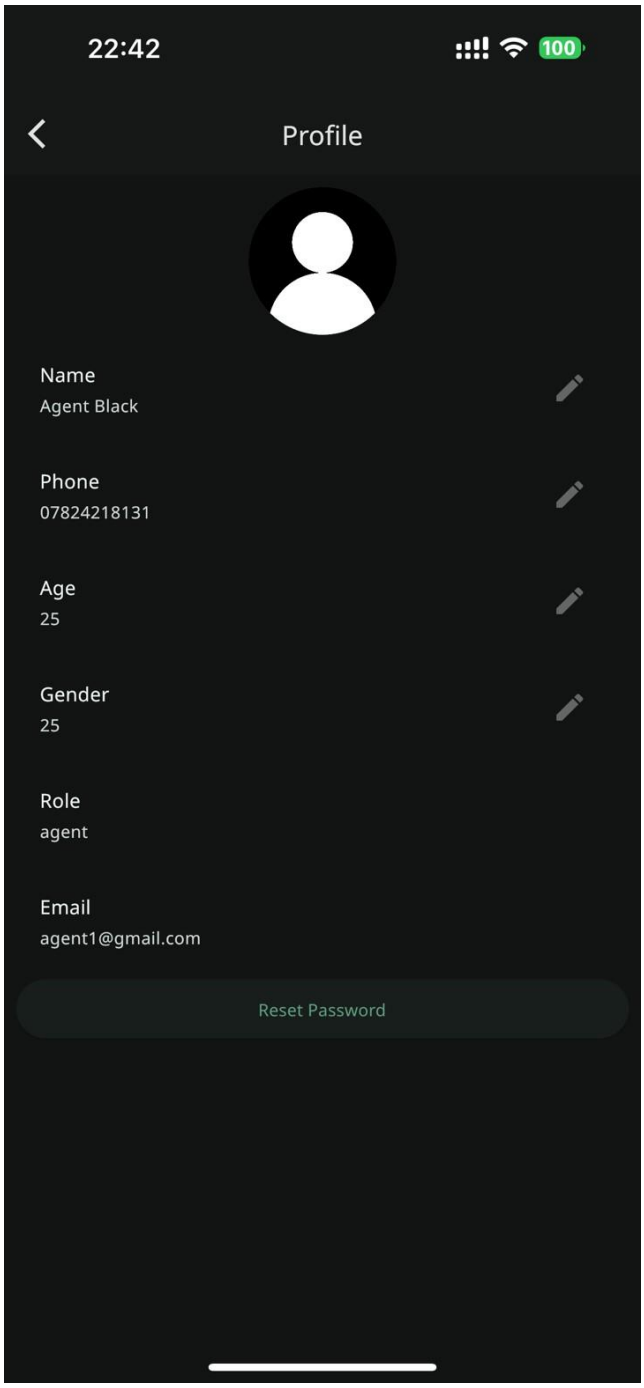


Figure 4-15 Edit Profile All Users Interface

4.6 Chapter Summary

The chapter on requirements analysis design discussed the process of identifying and analysing functional and non-functional requirements for a system. Use case diagrams were used to visually represent the functional requirements, and system architecture and interfaces were considered to ensure the system was well-structured, maintainable, and user-friendly. The importance of interface design was also highlighted as it is crucial to make the system accessible and meet the needs of stakeholders.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

During the implementation phase, the coding of system main functions is carried out, where the core functionalities of the real estate management application are developed and integrated. The interfaces of the system main functions are also designed to provide a user-friendly and intuitive experience for users. Black box testing is conducted to evaluate the application's functionality without examining its internal code, ensuring that it meets the desired specifications and requirements. Additionally, system flow is carefully considered to ensure a seamless navigation and interaction between different components of the application. This chapter aims to address these essential elements of implementation and testing, contributing to the overall quality and effectiveness of the real estate management application.

5.2 Coding of System Main Functions

The following code segment below shows how agents can add properties to the application. The application will need the agent to add images of the property (one for the home page screen view, and others of all around the property), the location of the property, nearby places, description, property type, price. And below is code sample of related part:

```
      class             _AgentAddPropertyScreenState             extends
State<AgentAddPropertyScreen> {
        final _formKey = GlobalKey<FormState>();
        List<Place> places = [];
        final amenityController = TextEditingController();
        List<String> includedAmenities = [];
        bool isLoading = false;

        final List<String> types = cons
```

```
    'bank',
    'gas_station',
    'book_store',
    'cafe',
    'clothing_store',
    'convenience_store',
    'department_store',
    'drugstore',
    'electronics_store',
    'hospital',
    'jewelry_store',
    'movie_theater',
    'park',
    'pharmacy',
    'primary_school',
    'restaurant',
    'secondary_school',
    'shopping_mall',
    'supermarket',
    'tourist_attraction',
    'university'
];

bool isApartment = false;
bool isVilla = false;
bool isHouse = false;
String title = '';
String description = '';

GeoPoint selectedLocation = const GeoPoint(0, 0);
List<Asset> selectedImages = [];
List<String> imageUrls = [];
String propertyType = '';
String? floor;
String? apartmentName;
String? residentialProject;
bool forSale = false;
bool forRent = false;
double price = 0.0;
double size = 0.0;
int roomCount = 0;
int bedroomCount = 0;

Future<List<Place>> fetchUniqueNearbyPlaces(
    {required LatLng location, required List<String> types})
```

```
          const String baseUrl =

'https://maps.googleapis.com/maps/api/place/nearbysearch/json';
          const               String               apiKey                =
'AIzaSyAjaGWBMTQuFityWuHRB5lokimKetelTEE';
          const String radius = '500'; // Radius in meters

          List<Place> placesList = [];


          for (String type in types) {
            final response = await http.get(Uri.parse(

'$baseUrl?location=${location.latitude},${location.longitude}&radius=$ra
dius&type=$type&key=$apiKey'));
            log('Response status: ${response.statusCode} searching...');
            if (response.statusCode == 200) {
              Map<String,        dynamic>        placesResponse        =
json.decode(response.body);
              List<dynamic> places = placesResponse['results'];


              if (places.isNotEmpty) {
                placesList.add(
                    Place.fromJson(places[0])); // Add the first place
of this type


              }
            } else {
              throw Exception('Failed to load nearby places');
            }
          }
          return placesList;
        }


        Future<void> _selectLocation() async {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => PlacePicker(
                apiKey: 'AIzaSyAjaGWBMTQuFityWuHRB5lokimKetelTEE',
                onPlacePicked: (result) {
                  setState(() {
```

Figure 5.1 Code sample of the api

5.3 Interfaces of System Main Functions

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

5.4 Testing

Testing is vital for a real estate management application as it ensures reliability, accuracy, performance, security, and user-friendliness. It validates data integrity, verifies key features, assesses performance, identifies security vulnerabilities, and enhances usability.

5.4.1 Black box Testing

Black box testing is an important aspect of testing a real estate management application. It involves evaluating the application's functionality without having knowledge of its internal code or structure. Testers focus on the inputs and outputs of the system to validate its behavior and ensure that it meets the specified requirements. By treating the application as a "black box," testers can simulate various user scenarios, test different functionalities, and identify any discrepancies or defects in the application's behavior. This type of testing helps uncover issues related to usability, functionality, and compatibility, ensuring that the application functions as intended for end users.

5.4.1.1 System Flow

The system flow of the real estate management application starts on the login page, this is where the user can login into the application with their account credentials that they have created, but if the user does not have an account already, they can easily click on the register button under the login button and fill out the account creation details. Their name, email, gender, phone number, age, and password. After they have created the account, they can now login in and be taken to the home page of the application. For the user, the user can only view the properties for sale, that includes the cost, images and location.

5.4.1.2 Input Output Verification

Tabel 5.1 Input verification test

Test Case ID	TC001	TC002	TC003
Name	Arivan	Evan	Eva
Email	arivangmail	Evan@gmail.com	Eva@gmail.com
Password	Arivan1@	evan@	Evaaa24@#
Age	22	23	15
Phone Number	07701234567	0770123456	
Expected Result	**Actual Result**		
Error message input data is filled out	✓	✓	✗
Error message name must be 4 letters long	✓	✓	✗
Error message	✗	✓	✓

email is available			
Error message password has 1 capital, 1 non-capital, a number, and special character	✓	✗	✓
Error message user is ≥ 18	✓	✓	✗
Error message phone number must be 11 numbers	✓	✗	✗

5.4.1.3 Error Messages

Through testing, developers can address defects, optimize system resources, protect user data, and improve the overall quality of the application. It instills confidence in users and stakeholders, minimizing risks and contributing to the application's success in the real estate industry

5.4.2 White box Testing

. If the user is interested in a property they can click on the property and it will show more information about the property, such as a description of the property and more images of every angle of the property, inside and out. If the user wishes to buy the property or just view it with their own eyes, they can call the agency number that is given near the property description and buy it or just ask them if they can have a property tour.

5.4.3 User Testing

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

5.5 Chapter Summary

In conclusion, the implementation and testing phase of a real estate management application is crucial for its success and effectiveness. During implementation, the coding of system main functions and the design of user-friendly interfaces are carried out, ensuring the development of core functionalities and an intuitive user experience. Black box testing plays a vital role in evaluating the application's functionality, focusing on inputs and outputs to validate behavior and meet specified requirements. System flow is carefully considered to enable seamless navigation and interaction between different components. Testing, including black box testing, ensures reliability, accuracy, performance, security, and user-friendliness of the application. It helps address defects, optimize resources, protect user data, and instill confidence in users and stakeholders. The system flow of the application ensures

a smooth user journey from login to property viewing and potential purchase, enhancing the overall user experience.

CHAPTER 6

CONCLUSION

6.1 Introduction

This chapter provides an overview of the Real Estate management system developed for the Kurdistan region. That will mention the achievement of the project objectives and suggestions for future improvements.

6.2 Achievement of Project Objectives

The project objectives for this Real Estate application are to create a multi-platform management system for the real estate industry in the Kurdistan region, which would make it easier to manage, sell, and buy properties. Some of the key achievements of this project include:

- Multi-platform support: The application is designed to be accessible on multiple platforms, such as web and mobile, making it more convenient for users to access and use.

- User-friendly interface: The interface of the application is designed to be intuitive and user-friendly, which makes it easy for customers to navigate and find the properties they're looking for.

- Mapping and visual aids: The application includes a mapping system that displays the location of the properties, as well as a 360-degree view of the house's exterior and a 3D tour of the home, providing a realistic experience for the customer.

- Residential projects: The application also includes residential projects, which allows customers to browse and select properties that are being built or under construction.

- Multi-user accounts: The application allows for the creation of multi-user accounts, which allows agents to add and manage properties on the system, while customers can browse and select properties.

- Additional utilities: The application also provides additional utilities such as nearby utilities' information.

In summary, the project objectives have been achieved by creating a user-friendly and feature-rich multi-platform management system for the real estate industry in the Kurdistan region, which makes it easier for agents to manage and sell properties, and for customers to find and buy properties they're looking for, with a user-friendly interface, mapping, and visual aids and an easy buying process.

6.3 Suggestions for Future Improvement

In order to enhance the functionality and user experience of the real estate management system, there are several areas that could be considered for future improvement. These includes:

- Implement a recommendation system: A recommendation system that suggests properties to customers based on their browsing and search history, this could help customers discover new properties that align with their interests.

- Integration with smart home devices and technologies: Allowing customers to control smart home devices and view the status of these devices would allow them to see the properties' potential capabilities and connect them to their future smart home.

- Virtual Reality (VR) integration: Integrating VR technology into the application would give customers an even more realistic experience of

  browsing properties. It would also make it easier for agents to showcase properties in a more engaging way.

- Advanced filtering and search options: To make it easy for customers to find the property they're looking for, adding advanced filtering and search options would be beneficial. For example, it could include advanced search options such as price range, location, square footage, and a number of bedrooms.

- Social media integration: Integrating the system with social media platforms would help to promote the properties and make it easier for agents to share them with potential customers.

- Add a feature for customer reviews and ratings: Adding a feature that allows customers to leave reviews and ratings would be useful to agents when deciding which properties to promote and could also help other customers make informed decisions.

- Provide a messaging system: Implementing a messaging system would allow customers to communicate with agents and schedule viewings more easily.

- Add a feature for remote contract signing: The system should provide a feature that allows customers to sign contracts and complete the buying process remotely, making it more convenient for all parties involved.

- Provide an analytics Dashboard: An analytics dashboard would allow agents to monitor and evaluate the performance of their properties, track customer interactions, and measure the overall performance of the system.

REFERENCES

Amadi, T. (2014, September 1). Property management system. Academia.edu. https://www.academia.edu/18928139/PROPERTY_MANAGEMENT_SYSTEM

Cloete, C. (2012, January 1). The selection of property management software. ResearchGate. https://www.researchgate.net/publication/268359756_THE_SELECTION_OF_PROPERTY_MANAGEMENT_SOFTWARE

Ombati, T. (2022). Salient features of a robust real estate property management system. Journal of Service Science and Management, 15, 362-378. https://doi.org/10.4236/jssm.2022.153022

IvyPanda. (2020, January 6). Property management systems in the hospitality industry. https://ivypanda.com/essays/property-management-systems-hospitality-industry/

Flutter by Example. (2020, July 12). About Dart. Flutter by Example. https://flutterbyexample.com/lesson/about-dart

Darji, P. (2021, May 19). What is Flutter? The game changer in application development. Semaphore. https://semaphoreci.com/blog/what-is-flutter

Heller, M. (2022, June 8). What is Visual Studio Code? Microsoft's extensible code editor. InfoWorld. https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html

Homele. (2022, December 10). Homele.com: Properties for sale and to rent in Iraq. Homele. https://homele.com/en

Khanoo. (2022, December 10). Khanoo. Khanoo. https://khanoo.com/tabs/estates

Garcia, J. (2022, October 21). A2 Hosting review: 11 crucial things to consider in 2022. Tooltester. https://www.tooltester.com/en/hosting/a2-hosting-reviews/

Hayat, F., Rehman, A., Arif, K., Wahab, K., & Abbas, M. (2019, January 20). The influence of Agile methodology (Scrum) on software project management. IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/8935813

Google. (2021). Google Maps Platform | Google Developers. https://developers.google.com/maps

Google Maps Flutter. (n.d.). Retrieved February 11, 2023, from https://pub.dev/packages/google_maps_flutter

Appendix A Test cases

appendix A Software Requirement Specification

Software Requirements Specification

Project Title

DESIGNING AND IMPLEMENTING OF MULTI- PLATFORM REAL ESTATE MANAGEMENT SYSTEM

Version 1.0

Department and Faculty Software Engineering

Prepared by:

Arivan Shamal M. Zyad

a. Overview

Real estate refers to physical property tied to land, encompassing residential, commercial, or industrial purposes. It holds significant value and serves as a leading indicator of economic health. The real estate industry impacts various sectors, including home improvement, development, lending, insurance, and business. However, in the Kurdistan region, there is a need for an organized agency and platform to facilitate property transactions and assist both customers and agencies. By enhancing the user-friendliness and simplicity of the real estate industry, the market can experience substantial benefits.

b. Target Audience

- Homebuyers and renters: A real estate system targets individuals looking to buy or rent a home, apartment, or other property. These users might be interested in browsing listings, searching for properties based on specific criteria, and viewing detailed information about individual listings.
- Real estate agents: A real estate system targets agents who are responsible for uploading and managing listings on behalf of clients. These users might need the ability to add and update listings, manage user accounts, and access analytics and reports.
- Property developers and landlords: A real estate system targets developers and landlords who are responsible for managing properties and projects. These users might need the ability to add and update listings, manage user accounts, and access analytics and reports.

c. Project Team Members

Arivan Shamal M. Zyad

d. **Version Control History**

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**

1. Introduction

1.1 Purpose

The purpose of this Real Estate application is to provide a multi-platform management system that makes it easier to manage, sell, and buy properties in the Kurdistan region. This system is designed to assist both agents, who will be responsible for adding and managing properties on the system, and customers, who will be able to browse, select, reserve, and obtain information about the location of the agency to continue the buying process.

The system is designed to provide a user-friendly interface for customers to browse properties, with features such as a mapping system to display the location of properties, 360- degree views of the house's exterior, virtual tours of the home, and information about nearby utilities. This will enhance the browsing experience for the customers and make it easier for them to find and view properties that match their needs.

The system is also designed to support multi user accounts, with different levels of access and functionality for agents and customers. Agents will have the ability to add and manage properties on the system, while customers will be able to browse and select properties to reserve and obtain information about the location of the agency to continue the buying process. The system will allow customers and agents to perform their respective tasks more easily and efficiently. Ultimately, the system will help to streamline the process of buying and selling properties in the Kurdistan region, and provide a convenient and efficient way for agents and customers to interact and conduct real estate transactions.

1.2 **Scope**

The scope of this Real Estate application includes several key features and functionalities to support the management, sale, and purchase of properties in the Kurdistan region. The following are the main components of the scope of the system:

- Multi User Accounts: The system will support multi user accounts, with different levels of access and functionality for agents and customers. Agents will be able to add and manage properties on the system, while customers.

- Property Management System: The system will have a property management system that allows agents to add, edit, delete and view properties on the system.

- Search, filter and sort properties: The system will allow customers to search, filter and sort properties based on their preference.

- Mapping System Integration: The system will include a mapping system that will allow customers to view the location of properties, which will be integrated with the property management system.

- 360-degree views and virtual tours: The system will provide customers with 360-degree views of the house's exterior and virtual tours of the home.

- Nearby Utilities Information: The system will provide customers with information about nearby utilities, like schools, hospitals and transportations

1.3 **Definitions, Acronyms and Abbreviation**

- 3D Tour: An interactive digital presentation that allows customers to navigate a property in a three-dimensional environment.

- RDBMS (Relational Database Management System): A system that stores and manages data in the form of related tables.

- LBS (Location-Based Services): Services that are provided based on a user's current location, such as location-based searches, location-based advertising, and location-based navigation.

- UI (User Interface): The portion of a computer system or application that a user interacts with, including the layout, design, and functionality of the system or application.

1.4 **References**

Davis, A. M. (1993). Software Requirements: Analysis and Specification. Upper Saddle River, NJ: Prentice Hall.

Cohn, M. (2004). User Stories Applied: For Agile Software Development. Addison-Wesley Professional.

Kotonya, G., & Sommerville, I. (1998). Requirements Engineering: Processes and Techniques. Chichester, UK: Wiley.

Leffingwell, D., & Widrig, D. (2003). Managing Software Requirements: A Unified Approach. Addison-Wesley Professional.

1.5 **Overview**

The SRS is organized as follows:

The Introduction section of the SRS document provides an overview of the system, including its purpose, scope, definitions, acronyms, and abbreviations, references and an overview of the SRS.

The Overall Description section provides an overview of the system, including information about its product perspective, such as system interfaces, user interfaces, hardware interfaces, software interfaces, communication interfaces, memory, operations, and site adaptation requirements. It also includes information about the product functions, user characteristics, constraints, assumptions and dependencies.

2. Overall Description

There are several factors that can affect the products and their requirements, and it is important to consider these factors when developing and specifying the requirements for a product. Some of the general factors that can impact products and their requirements include:

- Market demand: The demand for a product in the market can influence its requirements, as products that are in high demand may need to be developed more quickly or have more features to meet the needs of customers.

- Competition: The presence of competing products in the market can also affect the requirements of a product. In order to stand out and differentiate itself from competitors, a product may need to have unique features or capabilities.

- Technology: The state of technology can also impact the requirements of a product. For example, if new technologies become available that can improve the performance or functionality of a product, these technologies may need to be incorporated into the product's requirements.

- Legal and regulatory requirements: Products may also be subject to legal and regulatory requirements, such as safety standards or environmental regulations, which can influence their requirements.

- User needs and preferences: The needs and preferences of the users of a product can also affect its requirements. By considering the needs and preferences of the target audience, a product can be designed and developed to better meet their needs.

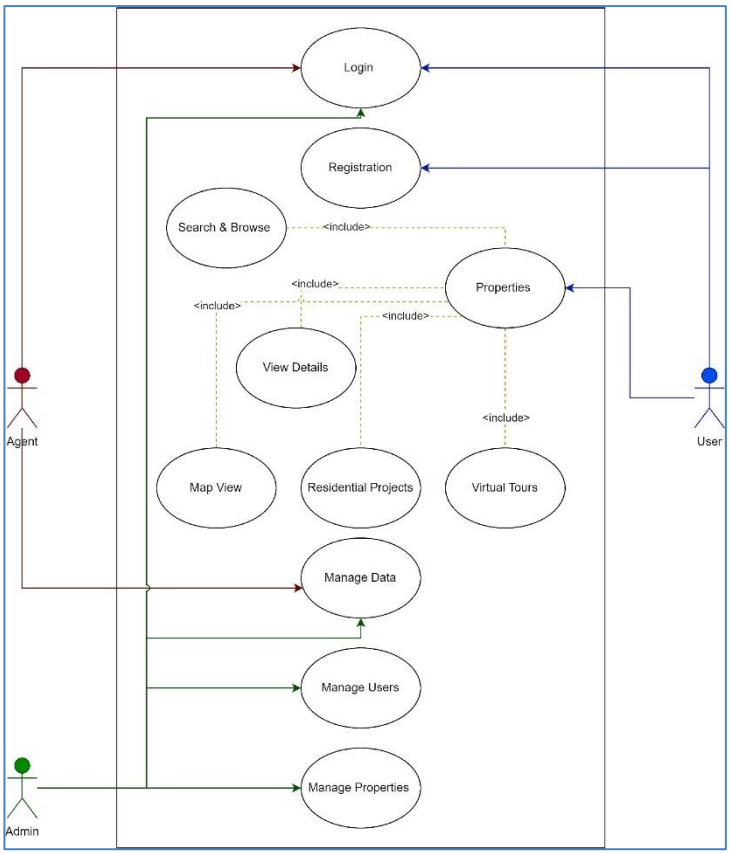


Figure 2.1: Use Case Diagram of My Real Estate

2.1 **Product Perspective**

My Real Estate, the product can be put into perspective with other related products by comparing its features, capabilities, and performance to those of similar products in the market. For example, My Real Estate could be compared to other online real estate platforms in terms of the range of properties it offers, the ease of use of its interface, and the quality of its virtual tours and maps.

2.1.1 System Interfaces

A customer wants to browse properties on the application and is presented with a list of properties that match their search criteria. Behind the scenes, the application's user interface makes a request to the database interface to retrieve the necessary information about the properties from the database. The database interface then performs the necessary queries and returns the relevant information to the user interface, which displays it to the customer. In another scenario, an agent wants to add a new property to the system. The agent uses the user interface to input the property's information and submit it to the application. The user interface then sends the information to the database interface, which performs necessary data validation, such as making sure all required fields are filled, and adding the property information to the database. In addition, when a customer wants to view detailed information about the properties, the user interface sends a request to the mapping and image viewer interface to retrieve the map and 360-degree images of the property. This interface handles the request, by retrieving the necessary map and images data, and displaying it to the customer through the user interface.

As you can see, the system interfaces provide a way for the user interface to interact with the underlying systems and subsystems of the application, such as the database and the mapping and image viewer in an efficient and seamless way, without the customer or the agent being aware of these underlying systems, allowing them to focus on their objectives.

2.1.2 User Interfaces

The system provides a user-friendly and intuitive interface for both customers and agents to use. We understand the importance of having a smooth and seamless user experience when it comes to real estate management.

For customers, the interface allows them to easily browse properties, view detailed information and images of properties, and search for properties that

match their criteria. The application mapping system displays the location of properties and provides a 360-degree view of the property's exterior and virtual tour of the interior, making it easy to get a detailed view of the properties that they are interested in.

For agents, the system user interface provides an easy way to add, manage, and update their properties listings. They can also view customer's requests and communicate with them in a centralized manner, allowing them to manage their properties efficiently.

The user interface is responsive, meaning it adapts to different screen sizes and works well on both desktop and mobile devices, so users can access the system from any device they prefer. The system user interface is designed in a way that makes the system's functionality easy to discover and understand, with clear and consistent layout, and instructions that guide users through their tasks.

Real estate management is a complex process, which is why this system provides a user interface that helps to simplify it for customers and agents. By providing a well-designed user interface, the system helps to make the process of managing, buying and selling properties in the Kurdistan region much easier and more efficient.

2.1.3 Hardware Interfaces

Network interface: The software will need to have a network interface that allows it to communicate with other systems and devices over a network, such as the internet. This might include interfaces with web servers, database servers, or other hardware components that are required for the operation of the system.

- Input/output devices: The software may also need to have interfaces with various input and output devices, such as keyboards, mice, touchscreens, or printers. These interfaces will allow users to enter data into the system and receive output from the system, and they may need to support a variety of devices and platforms.

- Storage devices: The software may also need to have interfaces with storage devices, such as hard drives or solid-state drives, in order to store and retrieve data. These interfaces will need to support fast and efficient data access, and they may need to be able to handle large volumes of data.

- Peripheral devices: The software may also need to have interfaces with various peripheral devices, such as cameras or scanners, in order to capture and process data. These interfaces will need to support the specific capabilities and protocols of the peripheral devices, and they may need to be able to handle large volumes of data.

It is important to carefully consider the hardware interfaces that are required for the real estate system in order to ensure that the software can effectively communicate with the necessary hardware components and devices. This can help to ensure that the system is able to function properly and provide the desired functionality to users.

2.1.4 Software Interfaces

The "My Real Estate Application" system has several interfaces with users and other systems. Here is a list of the system interfaces and the functionality of the software to accomplish the system requirements and the interface description to match the system.

- User interface: This interface allows users to interact with the system through various forms such as website or mobile application, it allows users to login, register, search, browse, and view details of properties, view residential projects, view virtual tours and also allow agents and admin to manage properties and users.

- Google Maps API: This interface allows the system to access Google Maps services to display properties on a map, provide location-based services, and create map views of properties. This interface is necessary for the map view and the location-based services functionalities.

- Database Management system (DBMS): This interface allows the system to store and retrieve data such as properties, users, and their data. It allows the system to perform CRUD operations (Create, Read, Update, and Delete) on the data, it could be relational or non-relational DBMS, and this interface is necessary for data management and user management functionalities.
- Communication Interface: This interface allows the system to communicate with external systems or users through different forms such as email, SMS, notifications. This interface is necessary for sending notifications,

confirmations and other types of communication with users.

2.1.5 Communication Interfaces

The system should be able to securely store and access user data, so a secure connection between the system and the database is required. The system should also be able to access the API in order to retrieve and send data. The system should also be able to connect to the internet in order to access external resources and update itself when necessary. Additionally, the system should also be able to use secure protocols such as SSL to ensure a secure connection and data transfer.

2.2 Product Functions

Table A.1 Functions

1	Login	All three types of users, customers, agents, and administrators, are able to access their respective areas of the system by logging in with their individual credentials.
2	Registration	Customers can create an account to gain access to the system's features and capabilities.
3	Properties - search & browse	Customers have the ability to search and browse through a variety of properties that can be bought or leased. They can narrow down their results by applying filters such as location, cost, and type of property.
4	Properties - view details	Customers are able to see comprehensive details regarding properties that they are interested in.
5	Properties - map view	The system provides an alternative to the traditional presentation of property listings, allowing customers to visualize the location of their desired properties through a map view.
6	Properties - residential projects	Customers can view a list of current residential projects, along with associated information.

7	Properties - virtual tours	The system offers customers the ability to virtually explore properties they are considering, giving them the chance to take a virtual tour.
8	Manage data	Agents are able to organize and keep track of their property listings and customer inquiries, while administrators have the ability to manage system data associated with properties, agents, and customers.
9	Manage users	Administrators are responsible for overseeing user accounts and providing assistance with the maintenance and use of the system. They are in charge of making sure everything runs smoothly.

2.3 User Characteristics

The "My Real Estate" system is intended to serve customers, agents and administrators, who have different needs and characteristics.

Customers are the primary users of the system; they are individuals or families looking to buy or rent properties in the Kurdistan region. They would typically use the system to search for properties that match their criteria, view detailed information about properties, and reserve a property. They would be looking for a user-friendly and intuitive interface that makes it easy to find the properties they are interested in. They would appreciate features like the mapping system and 360-degree views of properties, as well as the virtual tours that provide detailed information about the properties.

Agents, on the other hand, are the professionals who work in the real estate industry and use the system to manage properties and interact with customers. They would use the system to add and update properties listings, view customer's requests, and communicate with them. They would typically want a system that is easy to use and that can help them manage their properties efficiently. They would appreciate features like the centralized database and easy communication

features that allow them to keep track of their customer's request, and follow up with them in a timely manner.

Admins are responsible for managing the system's backend and would have access to various administrative tools, such as the ability to view system logs, manage user accounts, and update system configurations. They would be looking for a system that is easy to manage and maintain, with clear and intuitive administrative tools. They would also need to have access to a robust set of tools to secure the system, such as user authentication and authorization, data encryption, and backups.

2.3 Constraints

- Budget: The project's budget will be a constraint that can limit the options available to the development team. This may include limitations on the tools, technologies, or external services that can be used to build the system, or the number of team members that can be hired to work on the project.

- Timeframe: The project's timeline can also be a constraint, as the development team will need to deliver the system within a specific timeframe. This may limit the number of features that can be included in the initial release of the system, or require the development team to prioritize certain features over others.

- Technical constraints: The system may have technical constraints that limit the options available to the development team. These may include limitations on the hardware or software that can be used to build the system, or limitations on the types of data that can be stored or processed.

- Legal and regulatory constraints: The real estate industry is heavily regulated, so it's important to comply with any relevant laws and regulations. This may include limitations on the types of data that can be collected or stored, or the way in which data is protected and used. Developers will have to make sure that they comply with any relevant laws and regulations to avoid any legal issues.

- Security: Security and data privacy is a critical aspect for the system, developers will have to make sure that the system is designed with security in mind and that the system and user data is protected from potential threats. This may limit the options available for the technology stack or the way data is stored, processed and transmitted.

2.4 Assumption and Dependencies

- Assumptions about user behavior: The development team may make assumptions about how users will interact with the system, such as the types of properties customers will be looking for, or the types of tasks agents will need to perform. These assumptions will need to be validated through user research and testing to ensure that the system meets the needs of the users.

- Assumptions about network connectivity: The system may assume that users will have access to a reliable internet connection, which will be necessary for certain features like mapping and virtual tours. If this is not the case, the development team will need to come up with solutions to handle situations where the network connectivity is poor.

- Dependencies on third-party services: The system may depend on various third-party services, such as mapping and geolocation services, payment gateways, or external data sources. The development team will need to ensure that these services are available, reliable, and that they comply with any relevant regulations or laws.
- Dependencies on internal systems and subsystems: The system may depend on other internal systems or subsystems, such as databases and other services, to function properly. The development team will need to make sure that these systems and subsystems are available, properly configured, and that they comply with any relevant regulations or laws.
- Assumptions about the technology: The development team may make assumptions about the technology stack and its suitability for the system. This may require the team to evaluate the technology options available and make a choice that will be suitable for the system.

By keeping these assumptions and dependencies in mind, the development will be better equipped to design and build a system that meets the needs of the users and ensures that the project is delivered on time and within budget.

3. Specific Requirement

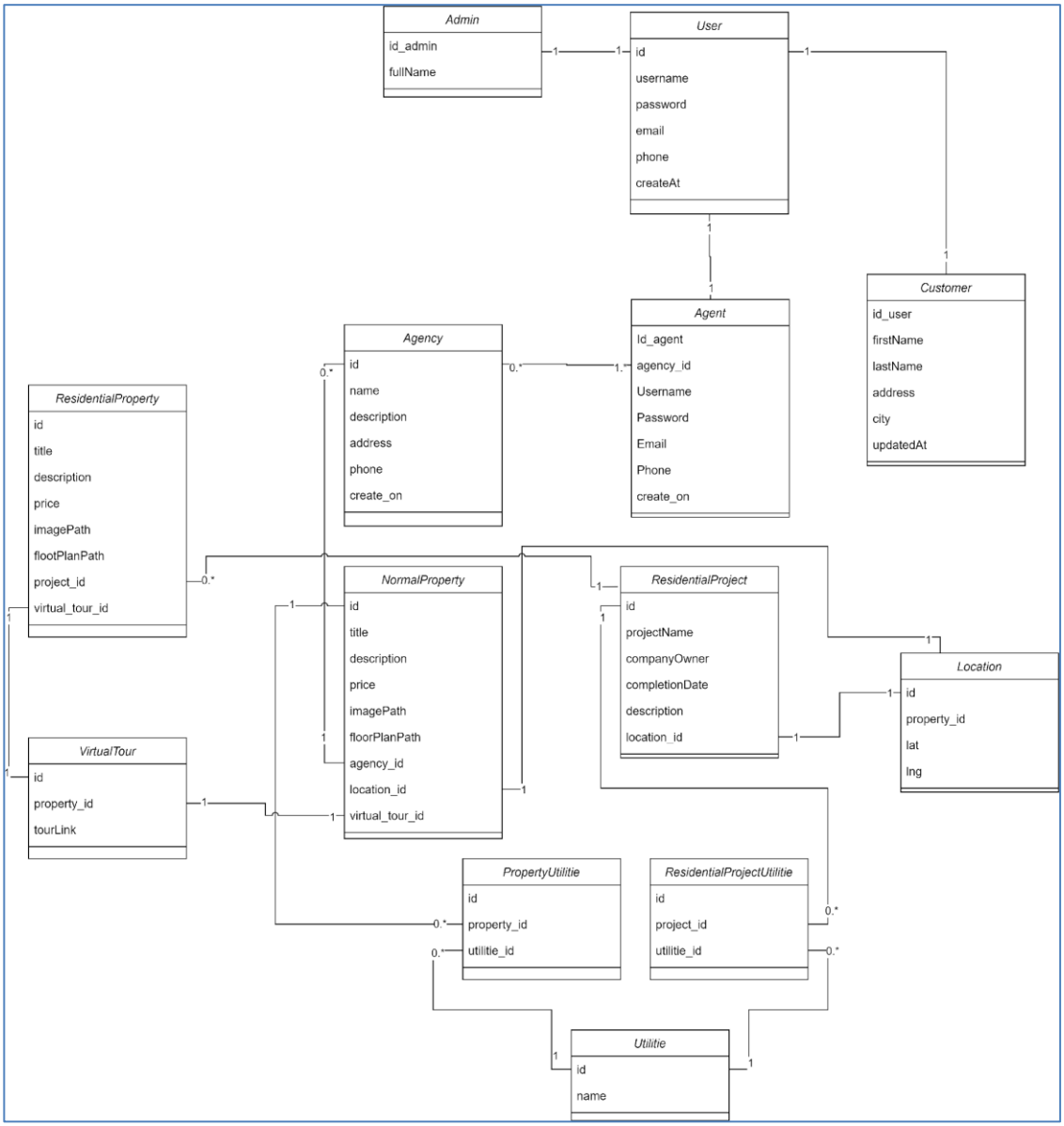


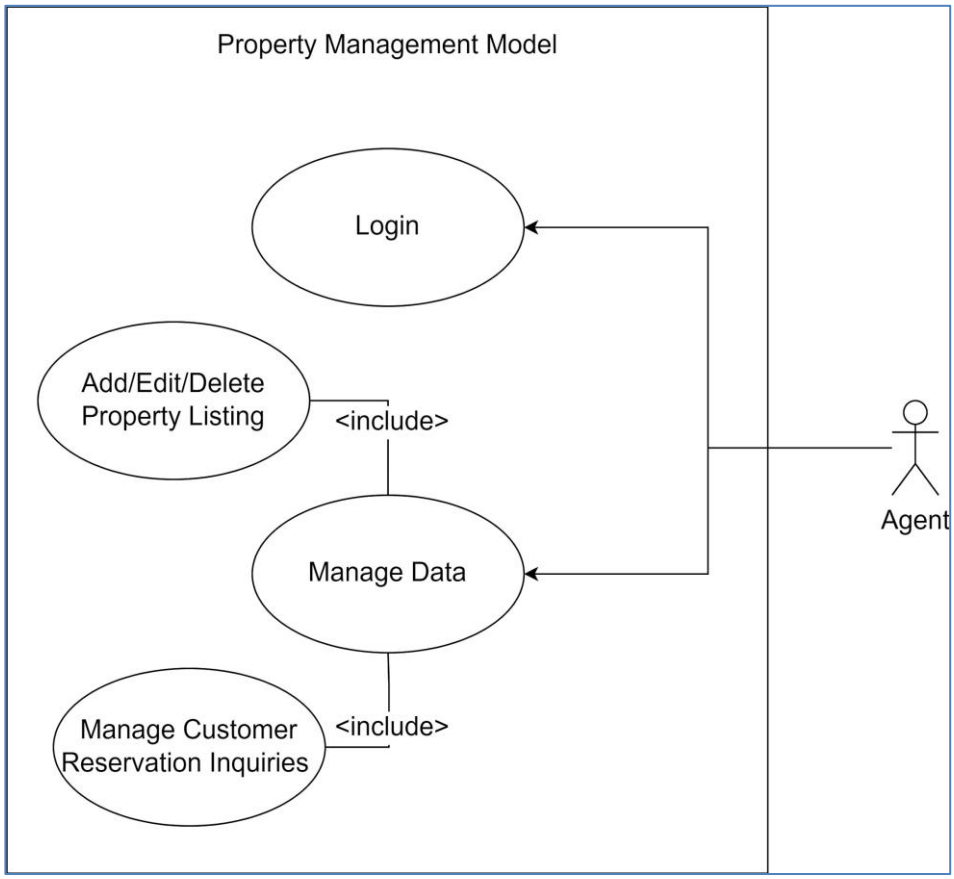
Figure 3.1: Domain Model of My Real Estate

3.1 System Features

3.1.1 Module Property Management Model

This model includes the use cases for adding and updating property listings, managing property reservations, and handling customer inquiries. This model would be used by agents to manage properties and communicate with customers as shown in the useAU case diagram in Figure 2.1.

Figure 3.2: Module Property Management Model



3.1.1.1 UC001: Use Case Login

This you case is responsible to let the customer login

Use Case Name: **UC001 login**

Scenario:	When an Agent opens the application they need will be prompted with a login screen.	
Triggering Event:	Click on application/ Click on login	
Brief Description:	This use case is for logging in the customers	
Actors	Customer/Agent/Admin	
Preconditions:	● Internet connection ● Having an account ● Having the application	
Post conditions:	The user will be logged in and redirected to home screen.	
Flow Event	Actor	System
	● Click login ● Enter login credential ● Logged in	● System sends http request to API ● The API response success ● Redirect to home screen
Exception Conditions:	● Login credentials are wrong if you forgot your account reset password ● New users create an account	

Table 3.1: Use Case Description for <Name of Use Case>

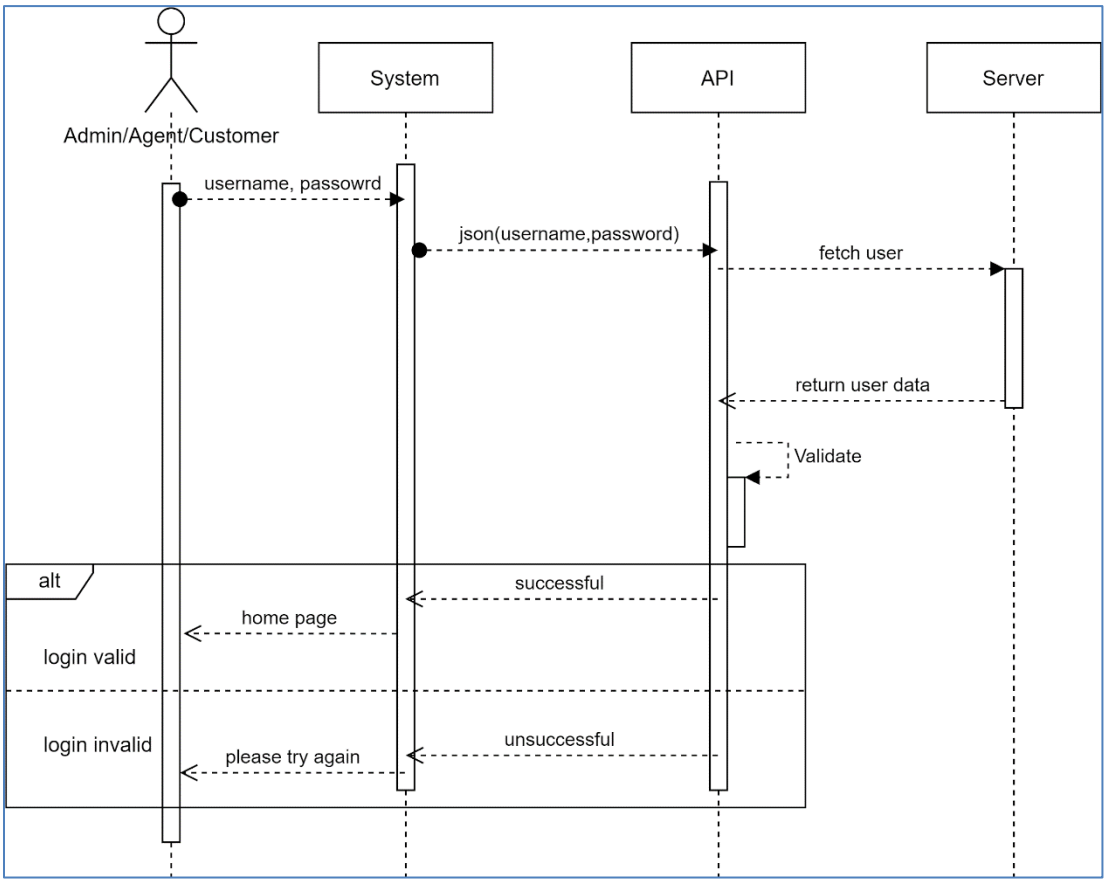


Figure 3.3: System Sequence Diagram of Login to system

3.1.1.2 **UC002 Add Property**

This you case is responsible of adding properties to the system

Table 3.1: Use Case Description for Adding Property

Scenario:	When an agent accesses the application and initiates the process of adding a new property to the system for listing.	
Triggering Event:	Click on Add Property	
Brief Description:	This use case is for adding property to the system	
Actors	Agent/Admin	
Preconditions:	● Internet connection ● Having an account ● Having the application ● Logged in with Agent/Admin account	
Post conditions:	A new property will be added	
Flow Event	Actor	System
	● Click Add property ● Enter property information	● System sends http request to API ● The API validate the information ● The new property will be added in database ● Sends success

		message
Exception Conditions:	● Wrong information, please check the entered information and try again.	

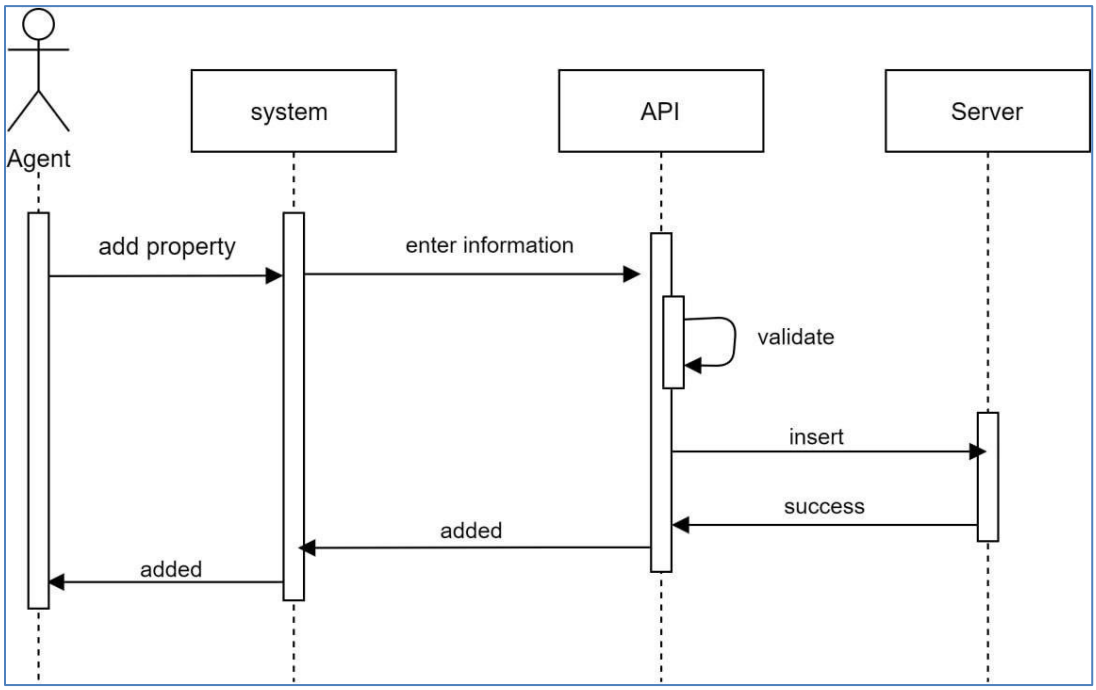


Figure 3.3: System Sequence Diagram of Add prop

3.1.2 Customer Management Module

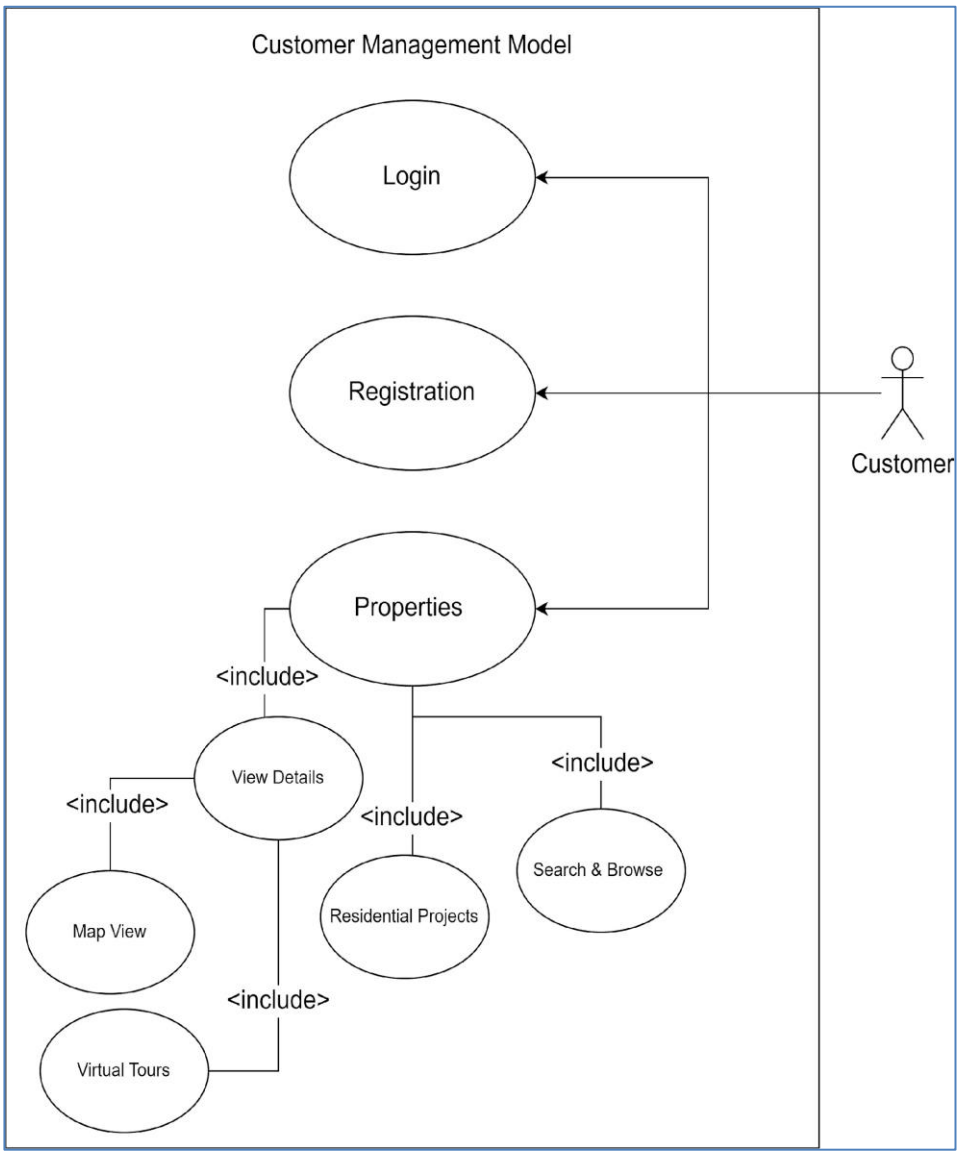


Figure 3.2: Module Property Management Model

3.1.2.1 **UC004**

Table 3.1: Use Case Description for registration

Scenario:	When the a new user download the application and wants to create an account	
Triggering Event:	Click create account	
Brief Description:	This use case is for creating a new account for the customers	
Actors	Customer	
Preconditions:	• Internet connection • Having the application	
Post conditions:	The customer will create a new account	
Flow Event	Actor	System
	• Click Register • Enter required information • Click register	• System will verify the inputs • System sends http request to the API • The API will validate data • Data will be inserted in data • System sends success message
Exception Conditions:	• Wrong information have been entered try again	

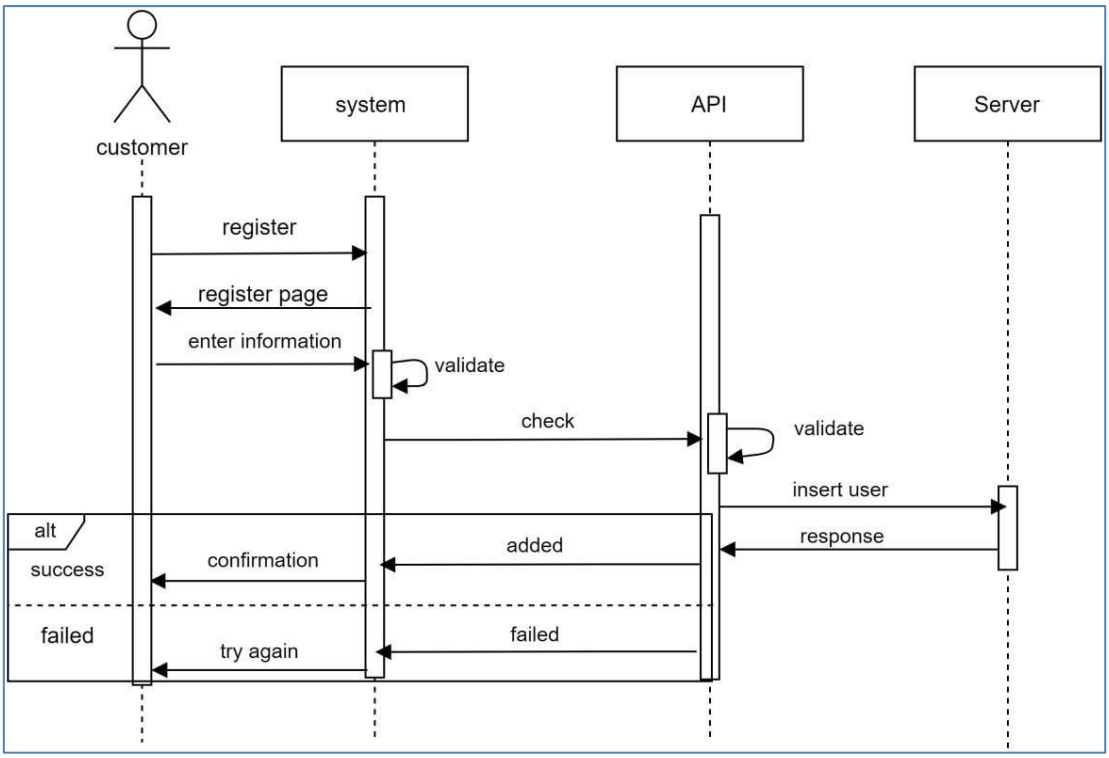


Figure 3.3: System Sequence Diagram of registration

3.1.2.2 **UC005**

Table A.5 Description for Browsing properties

Scenario:	When the customer browses for properties.
Triggering Event:	Click properties
Brief Description:	This use case is for browsing properties on the system
Actors	Customer
	• Internet connection • Having a customer account • Having the application • Logged in
Flow Event	

Exception Conditions:		
Preconditions:		
Post conditions:	The customer will be browse the property	
	Actor	System

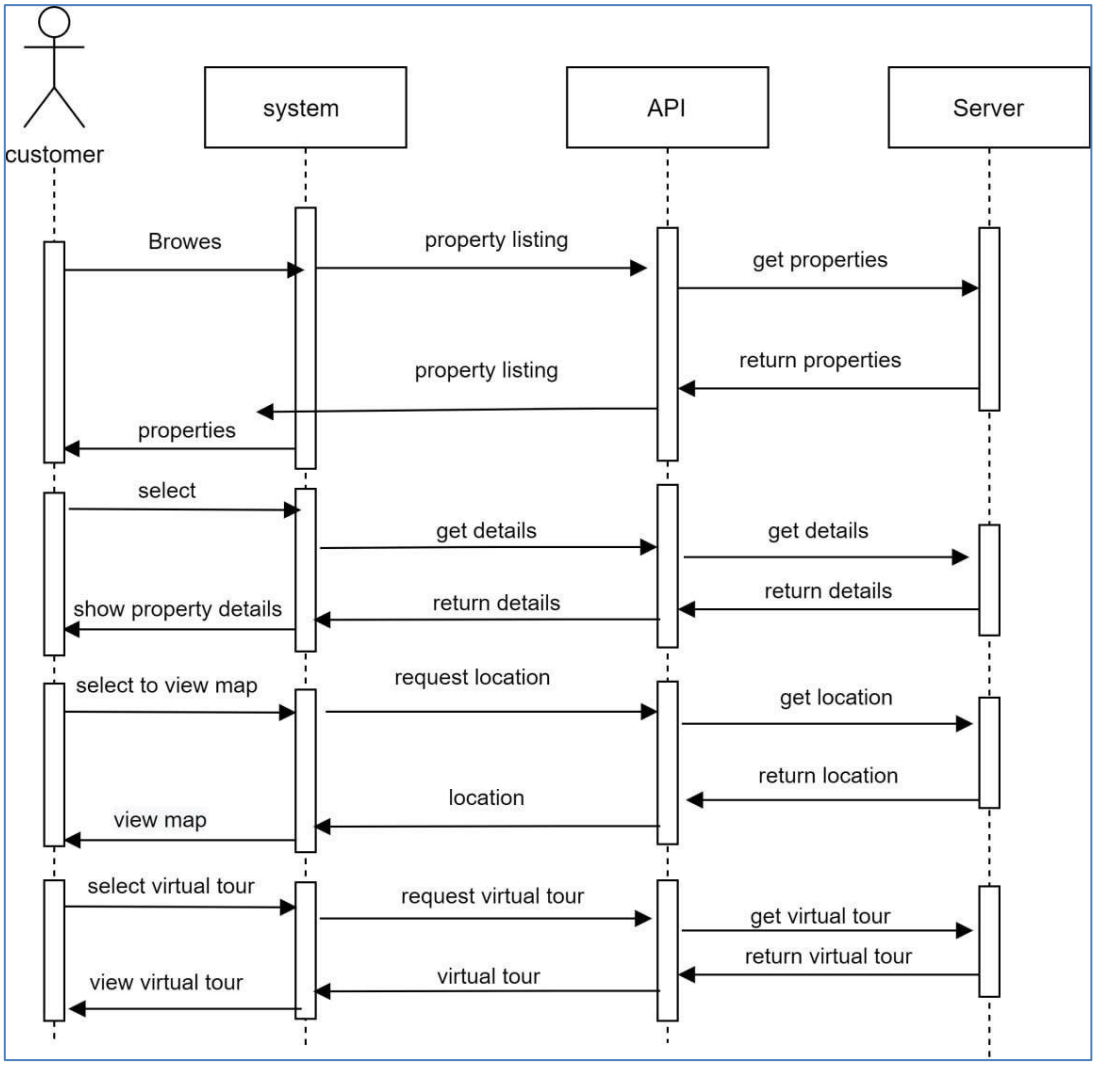


Figure 3.3: System Sequence Diagram of Browsing properties

3.1.2.3 **UC006**

This you case is responsible to let the customer login

Table 3.1: Use Case Description for residential projects

Scenario:	When the customer views customer inquiries about adding a property.
Triggering Event:	Click Residential Projects
Brief Description:	This use case is for viewing residential projects

Actors	Customer	
Preconditions:	• Internet connection • Having a customer account • Having the application • Logged in	
Post conditions:	The customer will view residential projects	
Flow Event	Actor	System
	• Click residential projects	
		• System sends http request to API to get residential projects
	• Actor view the residential projects • Actor open a residential project	
		• System sends http request to the API to get the residential project details
	• Actor view the details and view the properties	
		•
Exception Conditions:	• Non	

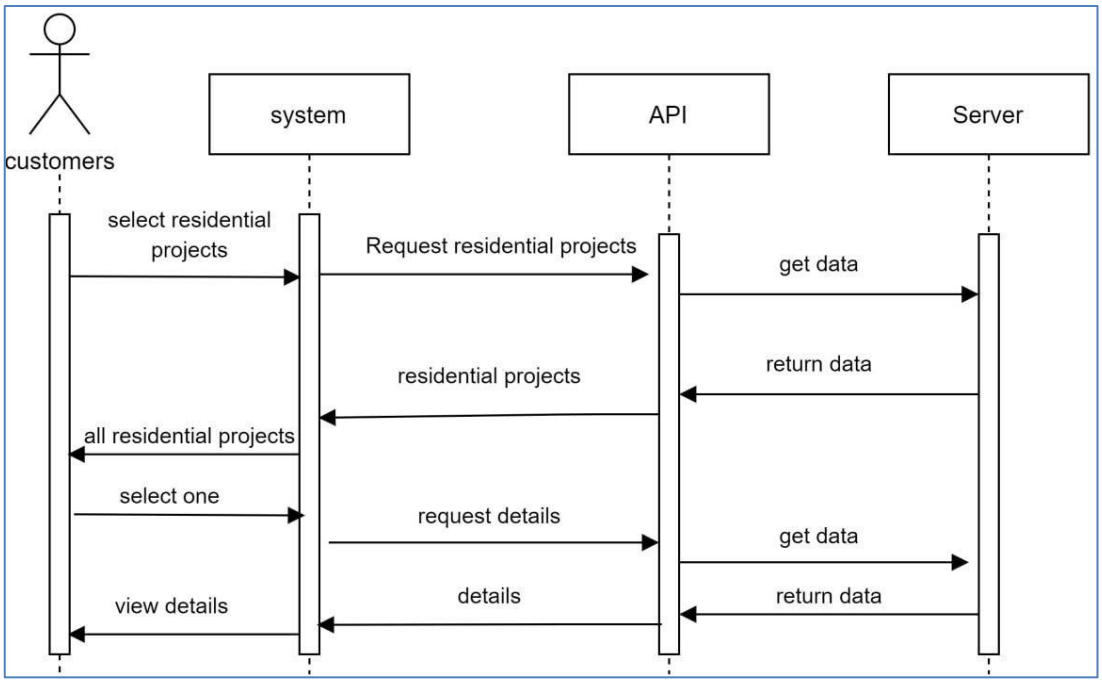


Figure 3.3: System Sequence Diagram of residential project

3.2 **Performance Requirements**

- Response time: The time it takes for the system to respond to a user's request. For example, the time it takes for the system to load a property's details or display a map view.

- Throughput: The number of requests that the system can handle at a given time. For example, the number of concurrent users that can search and browse properties at the same time.

- Scalability: The ability of the system to handle an increasing number of users or requests. For example, the system should be able to handle an increase in the number of registered users or properties without a significant decrease in performance.

- Security: The system should be able to protect data and user's information from unauthorized access.

- Maintainability: The system should be easy to maintain and update, this will help in the long term to keep it running smoothly and add new features.

- Reliability: The system should be reliable and available for the users almost all the time.

3.3 **Design Constraints**

- Technical constraints: Technical limitations of the system such as the use of specific programming languages, frameworks, or libraries. For example, the system might be built using a specific version of a programming language or framework, or it might be required to use a specific database management system.

- Hardware constraints: Hardware limitations of the system, such as memory and storage requirements, or the need for specific hardware components.

- Security constraints: Security requirements for the system such as data encryption or secure communication protocols.

- Legal and regulatory constraints: Compliance with legal and regulatory requirements, for example, data privacy laws or accessibility guidelines.

- Operational constraints: The system should be easy to operate and maintain.

- User experience constraints: The system should be user-friendly and should provide a good user experience.
- Accessibility: The system should be accessible for people with disabilities to use it.

3.4 **Software System Attributes**

- Portability: The system should be able to run on different platforms and devices such as smartphones, tablets, and desktops.

- Reusability: The system should be designed in a way that allows for easy modification and extension for future needs.
- Maintainability: The system should be easy to maintain and update, this will help in the long term to keep it running smoothly and add new features.

- Testability: The system should be designed in a way that makes it easy to test, this will make sure that the system is working as expected.

- Usability: The system should be easy to use and understand, this will make sure that the users are able to complete their tasks and achieve their goals.

- Security: The system should be secure and protect user's data and information from unauthorized access.

- Interoperability: The system should be able to communicate and exchange data with other systems.

- Reliability: The system should be reliable and available for the users almost all the time.

Software Design Documentation

Project Title

DESIGNING AND IMPLEMENTING OF MULTI- PLATFORM REAL ESTATE MANAGEMENT SYSTEM

Version 1.0

Department and Faculty Software Engineering

Prepared by:

Arivan Shamal M. Zyad

a. Overview

Real estate refers to physical property tied to land, encompassing residential, commercial, or industrial purposes. It holds significant value and serves as a leading indicator of economic health. The real estate industry impacts various sectors, including home improvement, development, lending, insurance, and business. However, in the Kurdistan region, there is a need for an organized agency and platform to facilitate property transactions and assist both customers and agencies. By enhancing the user-friendliness and simplicity of the real estate industry, the market can experience substantial benefits.

b. Target Audience

Homebuyers and renters: A real estate system targets individuals looking to buy or rent a home, apartment, or other property. These users might be interested in browsing listings, searching for properties based on specific criteria, and viewing detailed information about individual listings.

Real estate agents: A real estate system targets agents who are responsible for uploading and managing listings on behalf of clients. These users might need the ability to add and update listings, manage user accounts, and access analytics and reports.

Property developers and landlords: A real estate system targets developers and landlords who are responsible for managing properties and projects. These users might need the ability to add and update listings, manage user accounts, and access analytics and reports.

c. Project Team Members

Arivan Shamal M. Zyad

d. Version Control History

Version	Primary Author(s)	Description of Version	Date Completed
1.0	**Arivan Shamal**		**14-01-2023**

1. Introduction

Software design architecture for the "My Real Estate Application" system would provide a high-level overview of the system's overall structure and organization. It would describe how the different components of the system interact with each other and how they are organized to meet the requirements and constraints of the system.

The architecture would include a description of the different layers of the system, such as the user interface, business logic, and data storage layers. It would also describe the key components of the system, such as the mapping system, 360- degree view, and virtual tour features, as well as the residential projects and property management features. The architecture would also describe how the system interfaces with external systems, such as the mapping service, and how data is stored and retrieved.

1.1 Purpose

The purpose of the Software Design Document (SDD) is to provide a detailed description of the software design. The SDD will provide a detailed and precise description of the software architecture, design, and implementation, and how it satisfies the system requirements. Its targeted audience is the development team, the stakeholders, and the quality assurance team, it will be used as a reference for development, testing, and maintenance.

The SDD will be used to communicate the design decisions that were made during the development process, and to provide a clear and complete picture of the system's architecture and design. It will also be used to ensure that the system is designed to meet the requirements and constraints of the system, and that it is easy to maintain and update.

1.2 Scope

The "My Real Estate Application" is a multi-platform management system that is designed to make it easier to manage, sell, and buy properties in the Kurdistan region. The system offers a wide range of features that assist both agents and customers in their real estate transactions.

The scope of the software includes the following features:

- Multi-user accounts: Customers, agents, and admins can register and login to the system.

- Property management: Agents can add and manage properties on the system.

- Property search and browsing: Customers can search and browse properties on the system.

- Property details: Customers can view detailed information about properties on the system, including location, 360-degree view of the exterior and virtual tour of the interior.

- Residential projects: The system includes residential projects that customers can browse and view details.

- Mapping system: The system uses a mapping system to display the location of properties.

- User-friendly interface: Customers will be greeted with a user-friendly interface that makes it easy to navigate the system.

- Data management: Agents and admins can manage data in the system.

- Security: The system provides a secure environment for the user's data. The goals and objectives of this project are to provide a convenient and user- friendly platform for the real estate industry in the Kurdistan region, which will simplify the process of buying and selling properties, and provide easy access to information about properties, residential projects, and nearby utilities.

The benefits of this project include:

- Convenience: The system makes it easy for agents to manage properties and for customers to find and purchase properties, this will save time and effort.

- Detailed Information: The system provides customers with detailed information about properties, including location, 360-degree view of the dexterior, and virtual tour of the interior, this will help customers to make better decisions.

- Data Management: The system allows agents and admins to manage data in the system, this will make the process of managing properties more organized.

- Security: The system provides a secure environment for user's data, this will protect user's information from unauthorized access.

- User Experience: The system improves the user experience by providing a user-friendly interface, this will make it easy for customers to navigate the system and find what they are looking for.
- Accessibility: The system should be accessible for people with

disabilities to use it, this will give everyone a chance to access the service.

- Scalability: The system should be able to handle an increasing number of users or requests, this will make the system more efficient and sustainable in the long term.

- Multi-Platform: The system can be accessed from different platforms, such as smartphones, tablets, and desktops, this will make the system more accessible.

1.3 References

Specify complete list of references using a standardized reference format.

Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

IEEE Computer Society. (2000). IEEE Recommended Practice for Software Design Descriptions (IEEE Std 1016-2009). New York, NY: Institute of Electrical and Electronics Engineers.

Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Upper Saddle River, NJ: Prentice Hall.

1.4 Overview

The software design document encompasses key sections including the System Architectural Design, Detailed Description of Modules (Class Diagram, Sequence Diagrams), Data Design, and User Interface Design. The System Architectural Design provides a high-level overview of the system's structure and interactions. The Detailed Description of Modules includes Class Diagrams and Sequence Diagrams to illustrate the internal organization and interactions between modules. The Data Design section focuses on the system's data structures and relationships, often depicted using Entity-Relationship Diagrams. Lastly, the User Interface Design section describes the layout, visual elements, and interactions of the software's user interface. This document serves as a blueprint for developers, facilitating effective collaboration and implementation of the software system.

2. System Architectural Design

For this system the selected architectural design is the client-server architecture with the Model-View- Controller (MVC) design pattern for the "My Real Estate Application" system. The client-server architecture is a distributed computing architecture in which the client and server are separate entities that communicate with each other over a network. The client, in this case, would be the user's device such as a smartphone, tablet, or desktop computer and the server would be the system's back-end where all the data is stored and managed.

2.1 Architecture Style and Rationale

The MVC design pattern is a software design pattern that separates the application's data model, user interface, and control logic. The Model represents the data and the business logic of the application, the View represents the user interface, and the Controller handles the communication between the Model and the View. This separation of concerns allows for better maintainability and flexibility of the system.

The component and subsystem diagram would show the different components and subsystems of the system and how they are organized and interact with each other. It would include the client-side components such as the user interface and the mapping system, and the server-side components such as the property management system, data storage, and security.

The rationale for selecting this architecture and design pattern is that it allows for a clear separation of concerns and a modular design, which makes the system more maintainable, testable, and scalable. The client-server architecture also allows for a better distribution of resources and a

more secure environment for user's data. The MVC design pattern also allows for a better separation of concerns, which leads to a more maintainable and flexible system.

2.2 Use Case Diagram

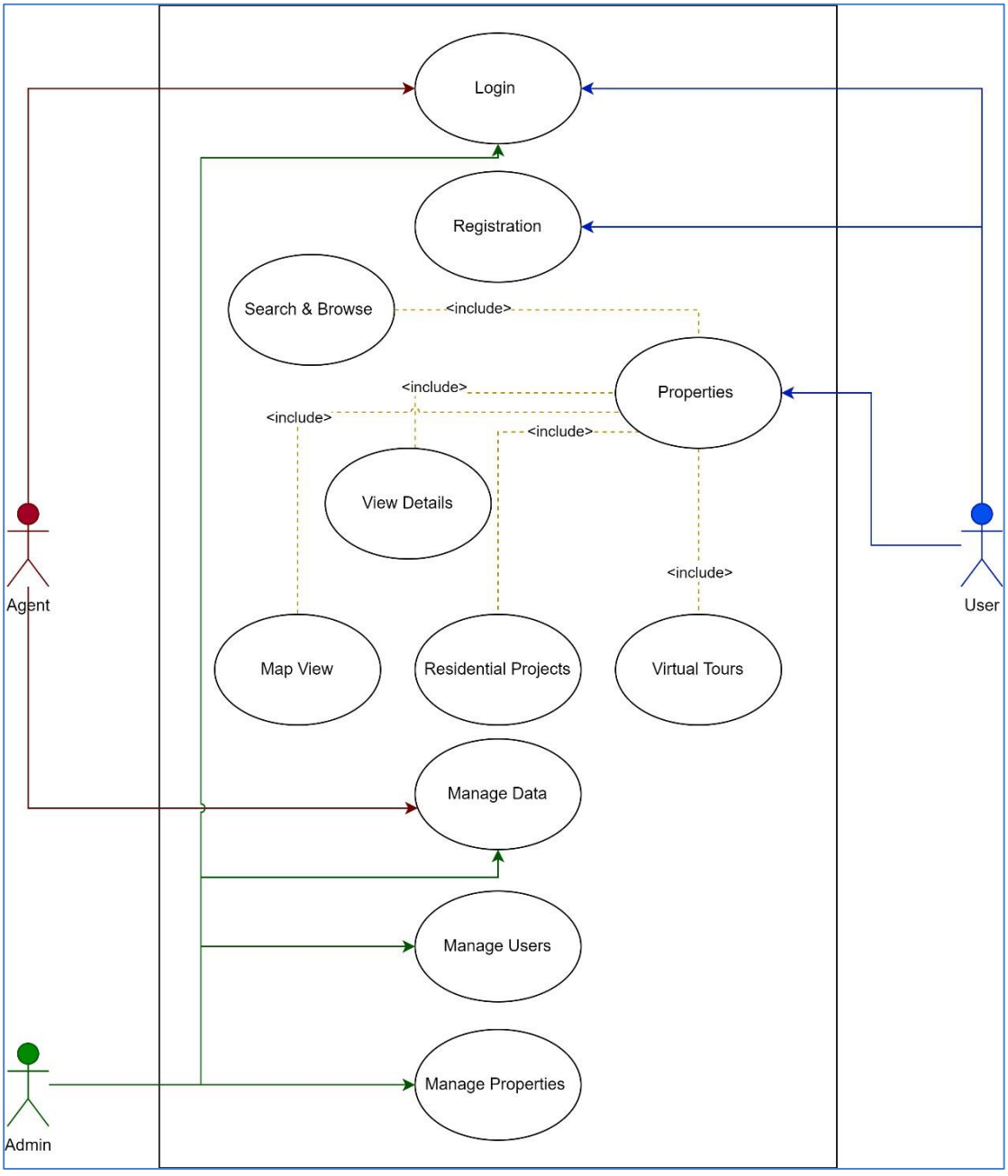


Figure 2.2: Use Case Diagram of My Real Estate

3. Detailed Description of Components

3.1 Detailed Description

3.1.1.1 Class Diagram

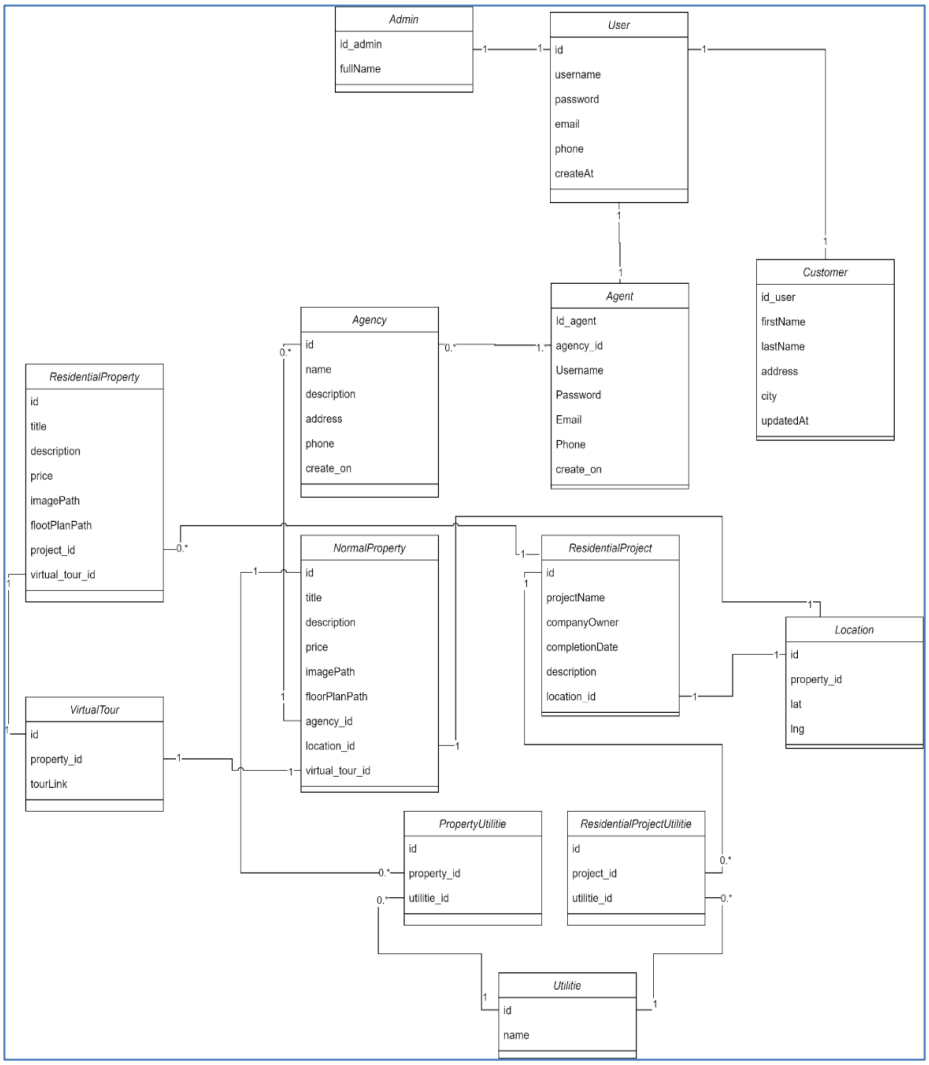


Figure 3.2: Class diagram for <Order Entry Package>

3.1.1.2 Sequence Diagrams

a) SD001: Sequence diagram for login

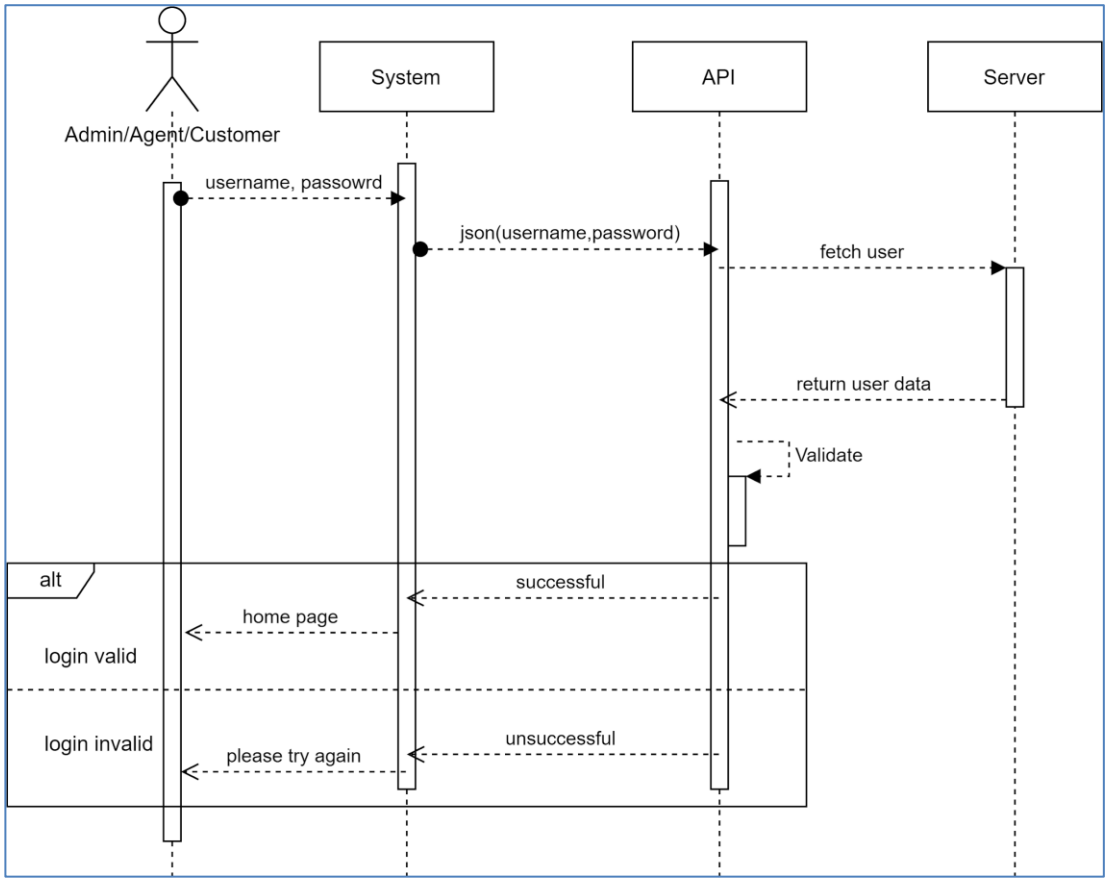


Figure 3.3: Sequence Diagram of <login>

b) SD002: Sequence diagram for Add property

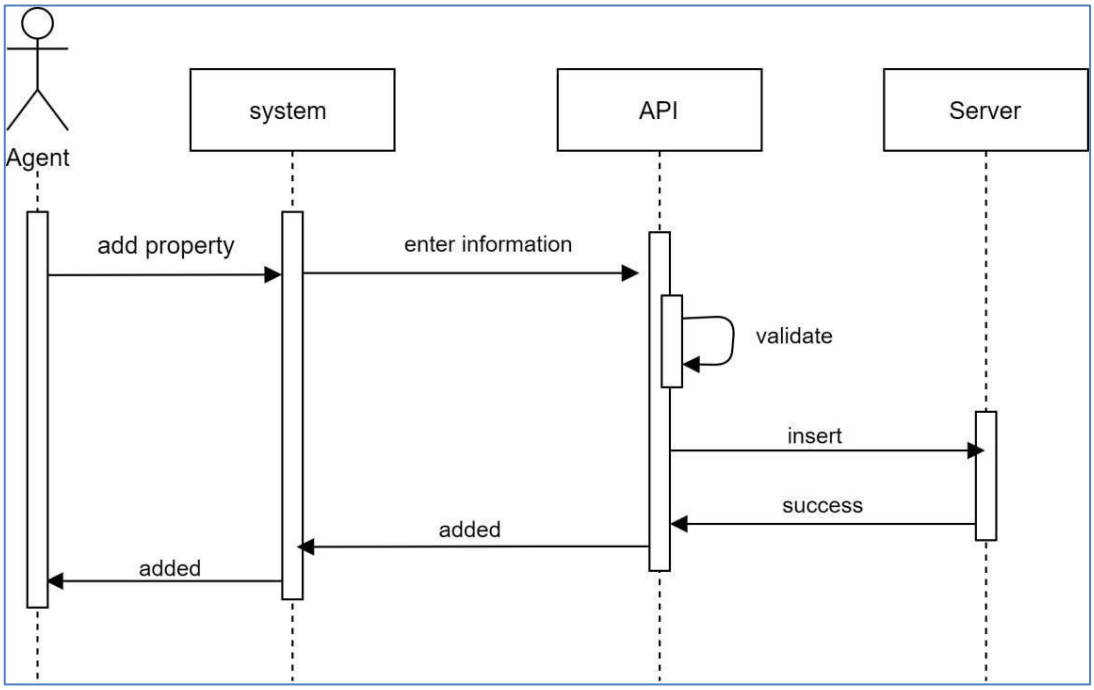


Figure 3.4: Sequence Diagram of <Add property>

c) SD003: Sequence diagram for Customer Inquiries

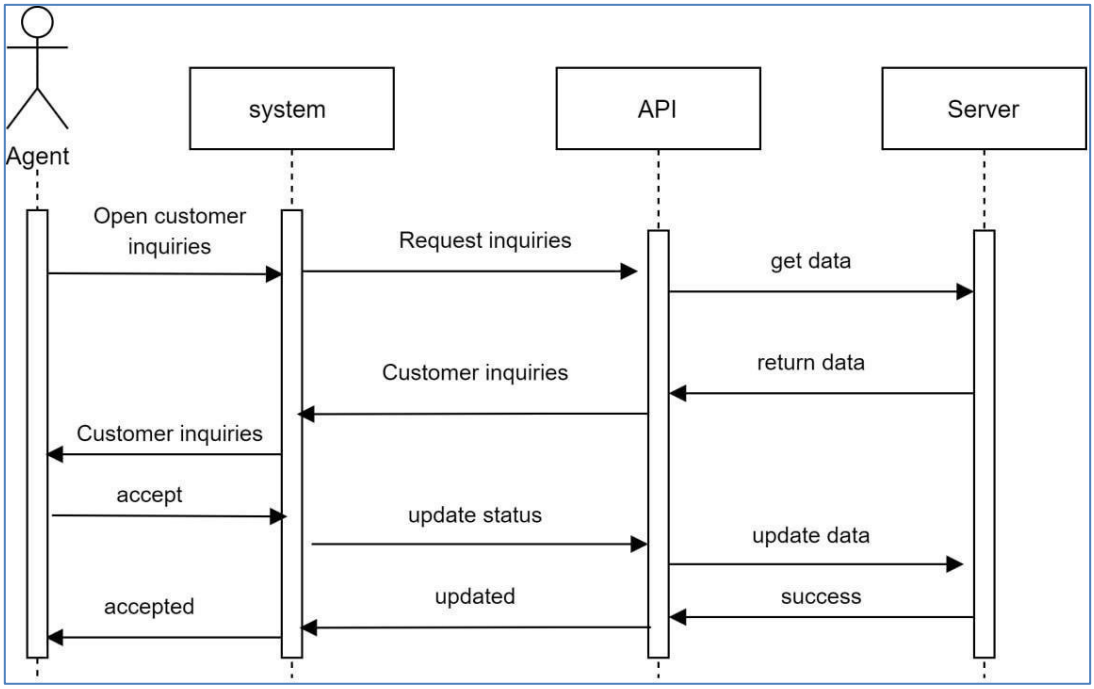


Figure 3.4: Sequence Diagram of <Customer Inquiries>

3.1.2 Subsystem <Customer Management>

3.1.2.1 P002: Package <Customer Management>

3.1.2.2 Class Diagram

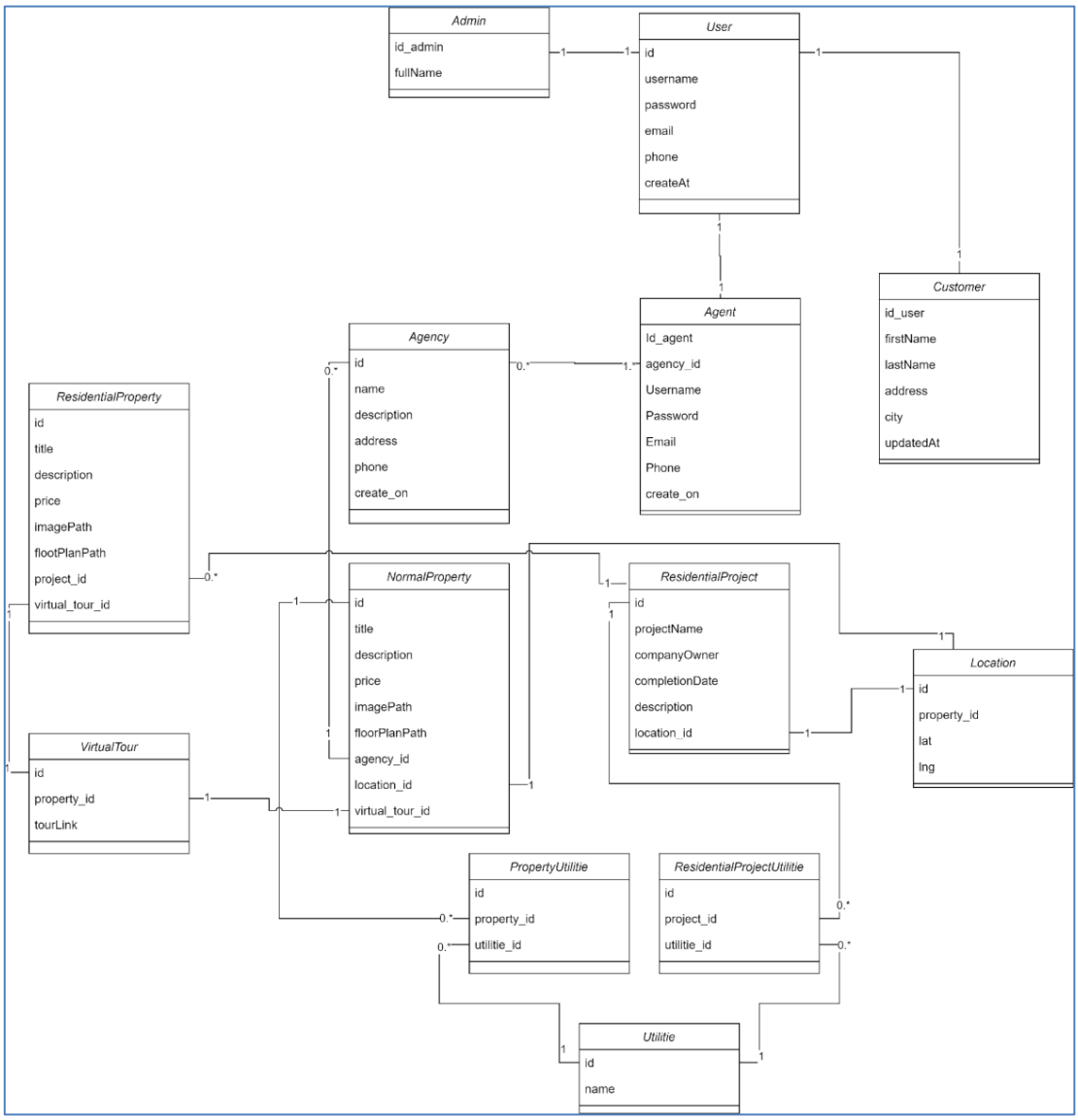


Figure B.4 Class diagram

3.1.2.3 Sequence Diagrams

a) SD001: Sequence diagram for Registration

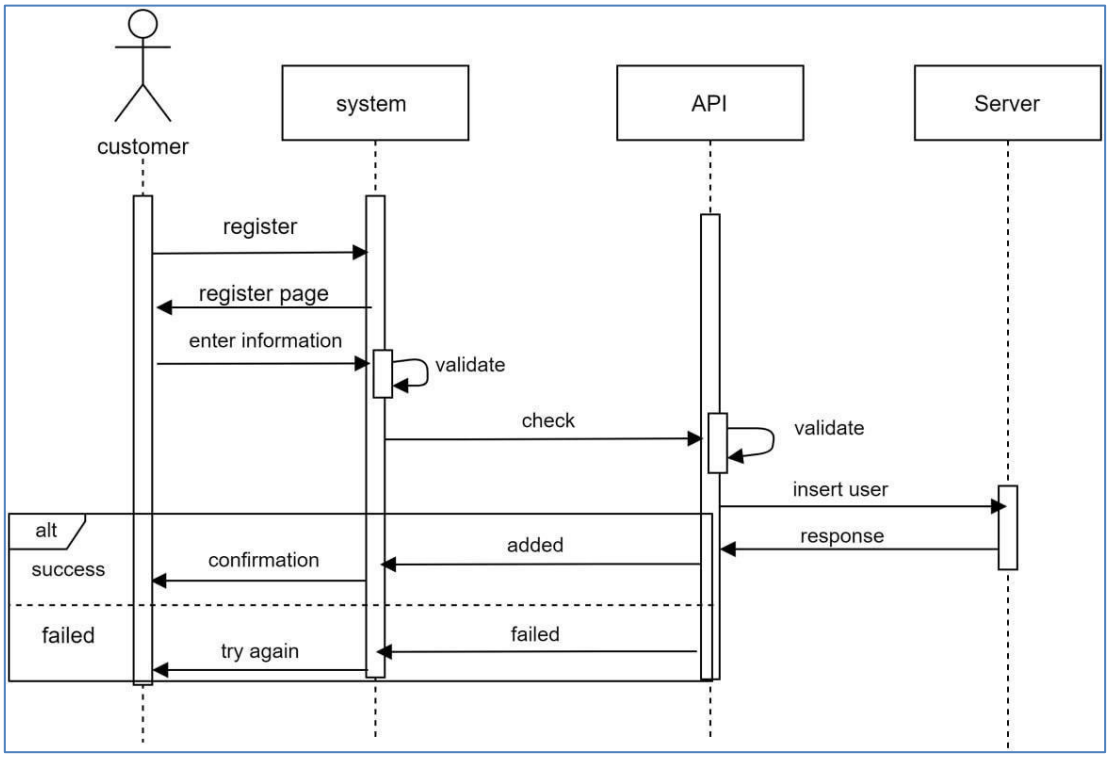


Figure 3.3: Sequence Diagram of <Registration>

b) SD002: Sequence diagram for Browsing properties

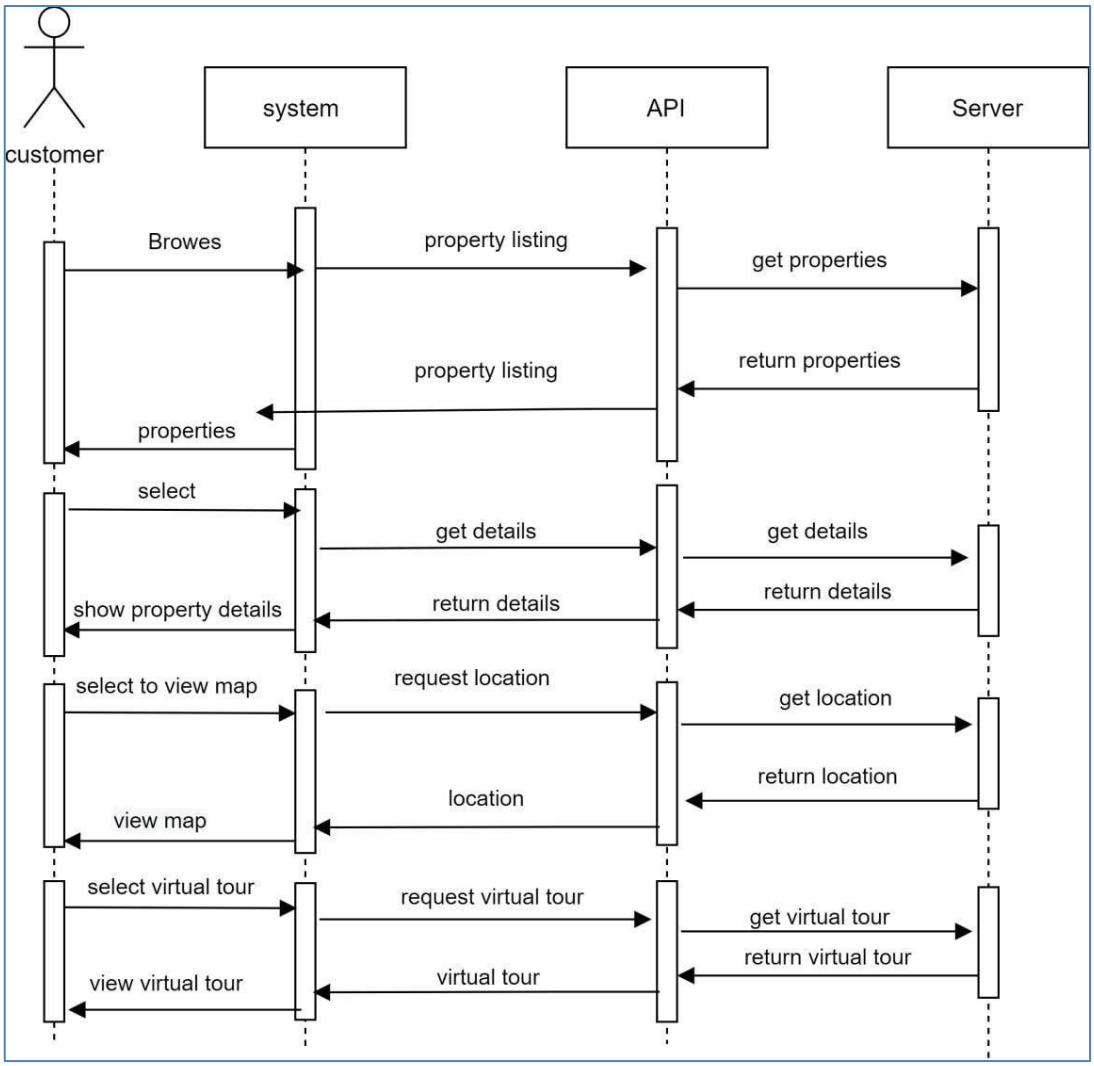


Figure 3.4: Sequence Diagram of <Browsing properties>

c) SD003: Sequence diagram for residential projects

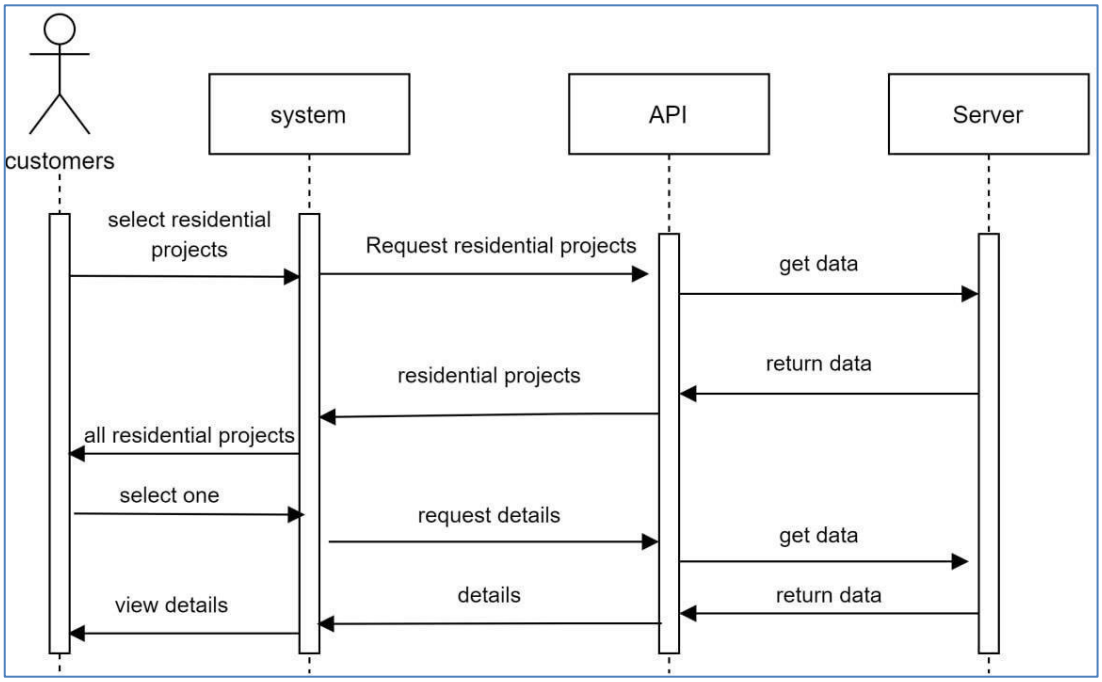


Figure 3.4: Sequence Diagram of <residential projects

3.1.2.4 Sequence Diagrams

a) SD001: Sequence diagram for Manage User

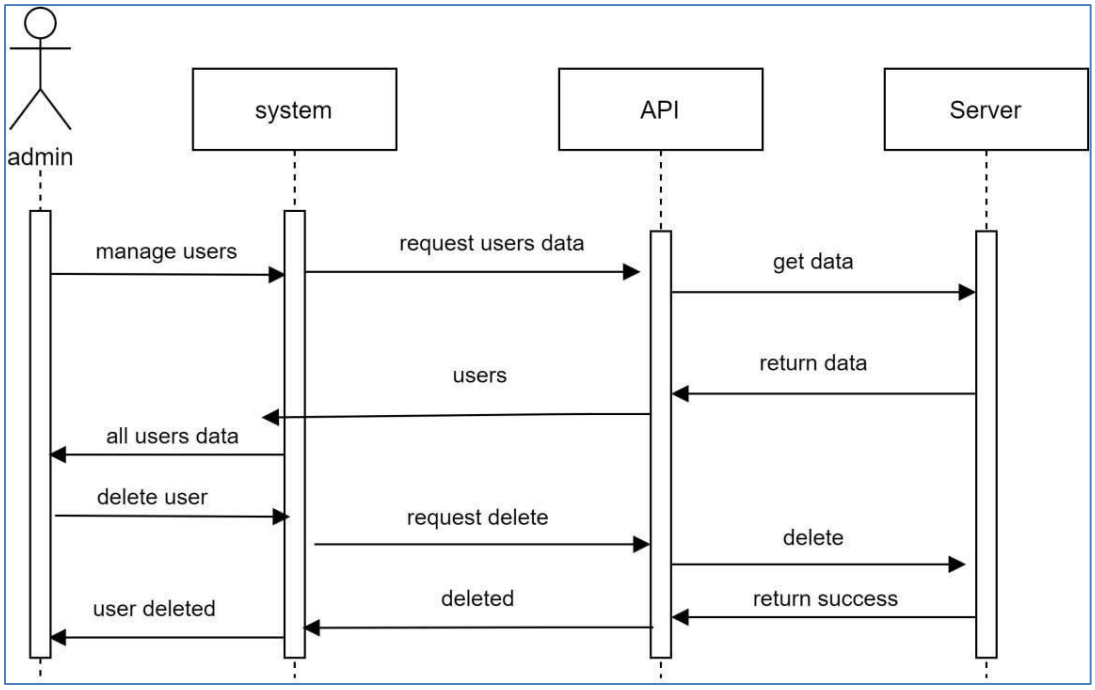


Figure 3.3: Sequence Diagram of <Manage User>

b) SD002: Sequence diagram for Property Management

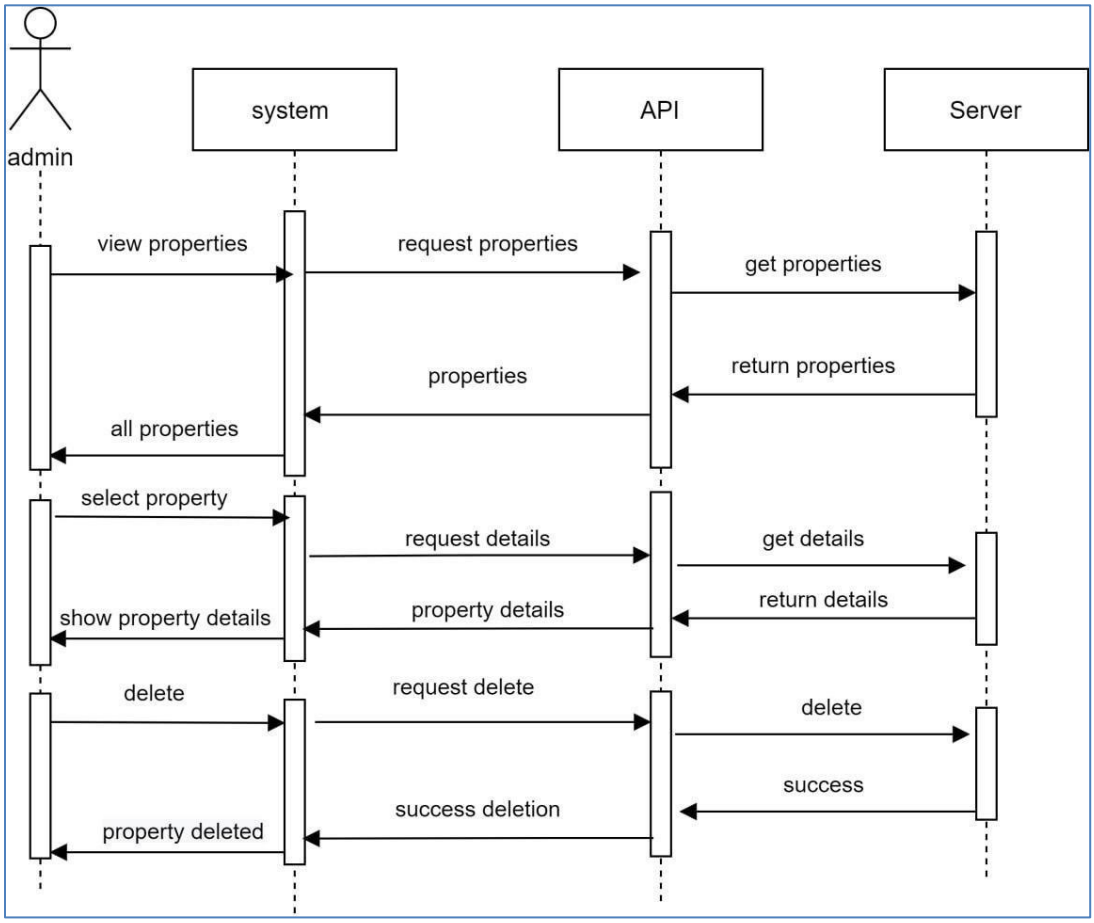


Figure 3.4: Sequence Diagram of <Property Management>

4. Data Design

4.1 Data Description

Entity Name	Description
User	this entity contains perspective of the Agent and the Customer
Customer	This entity contains perspective of only the User, and can view properties
Agent	This entity is linked with the Agency and is the one that discovers the properties
Agency	This entity contains Normal Property which allows the Agency to input properties into the application
Resident Project	This entity contains the whole project of the resident, such as the location, residential project util, and residential property
Location	This entity shows the location of the residential property
Residential Property	This entity contains the full detail of the property, such as the place name, price, images of the property, and description of the place

4.2 Data Dictionary

Class	Attributes	Data type
Admin	id_admin fullName	String String
User	id username password email phone createAt	String String String String String Date
Agent	id_agent agency_id username password email phone create_on	String String String String String String Date

Customer	id_user firstName lastName address city updateAt	String String String String String Date
Agency	id name description address phone create_on	String String String String String Date
Normal Property	id title description price imagePath floorPlanPath agency_id location_id virtual_tour_id	String String String Double String String

		String String String

5. User Interface Design

5.1 Overview of User Interface

As a user in the application, the user can login into the application with their username and password, but if the user does not have an account they can easily click on the register button that will allow the user to fill out all the fields on the register page to create an account on the application. After the user has logged in with their own username and password that they created or had, they can then be taken to the home page where it will allow the user to scroll through properties up for sale. The user can see the property price, location, and a front of the property image. For those who want more information about a particular property, a simple click on the desired listing takes them to a dedicated page.

This page provides additional images showcasing the property's interior and exterior from many angles. Furthermore, users can also learn more about the property by reading the description written down for the property and also a phone number of the agency so they can contact an agent for the property.

5.2 Screen Images

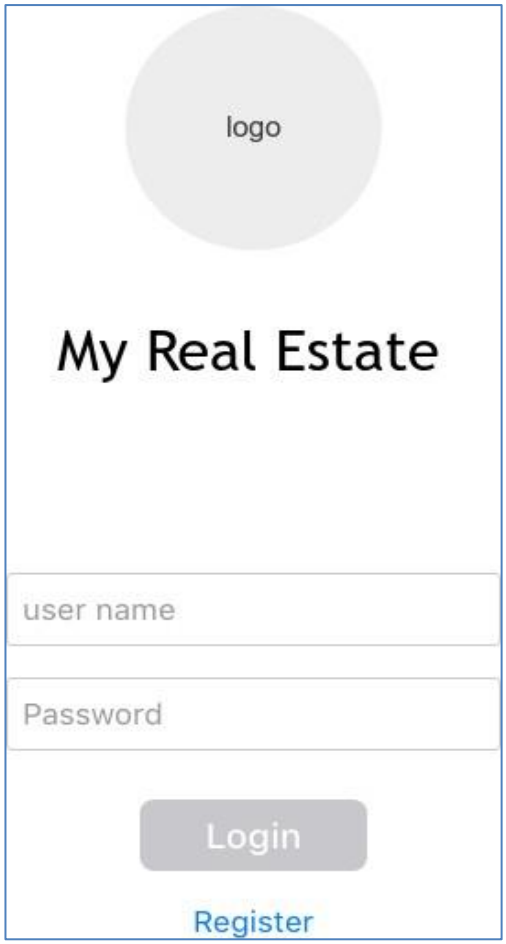


Figure B.1 Login screen

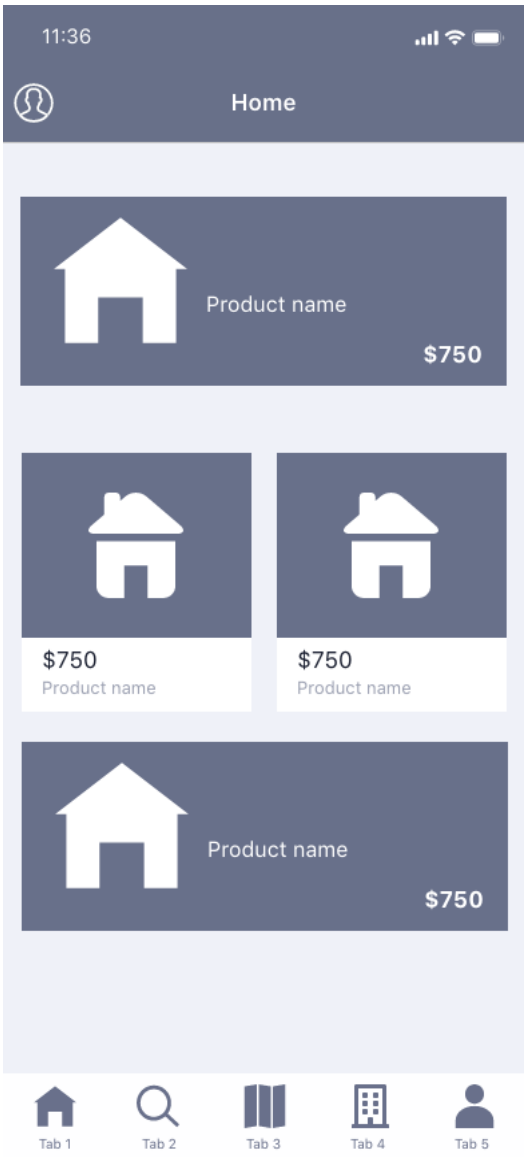


Figure – Home Page

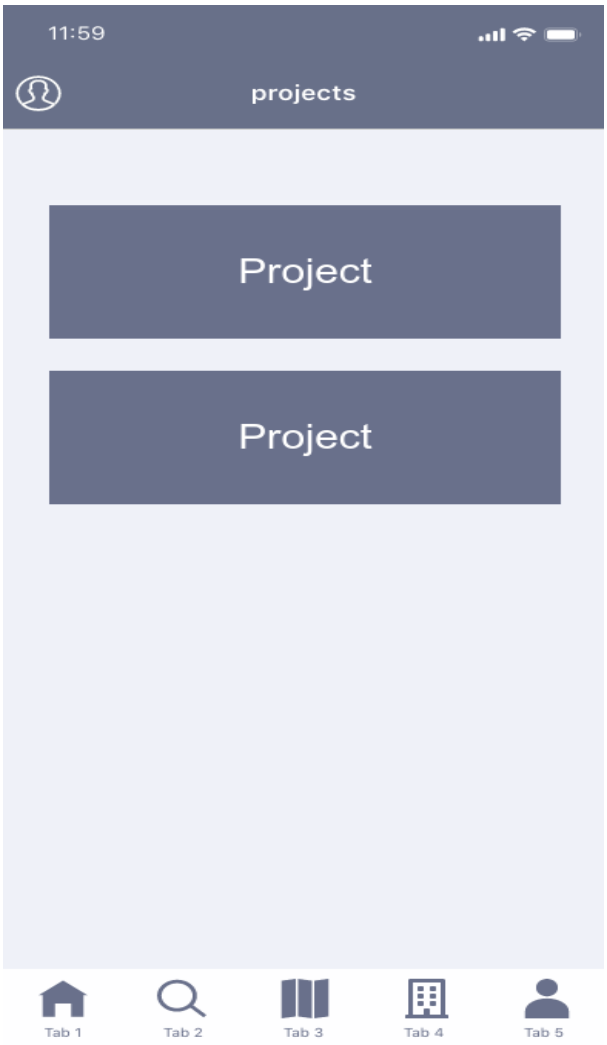


Figure – Residential Projects

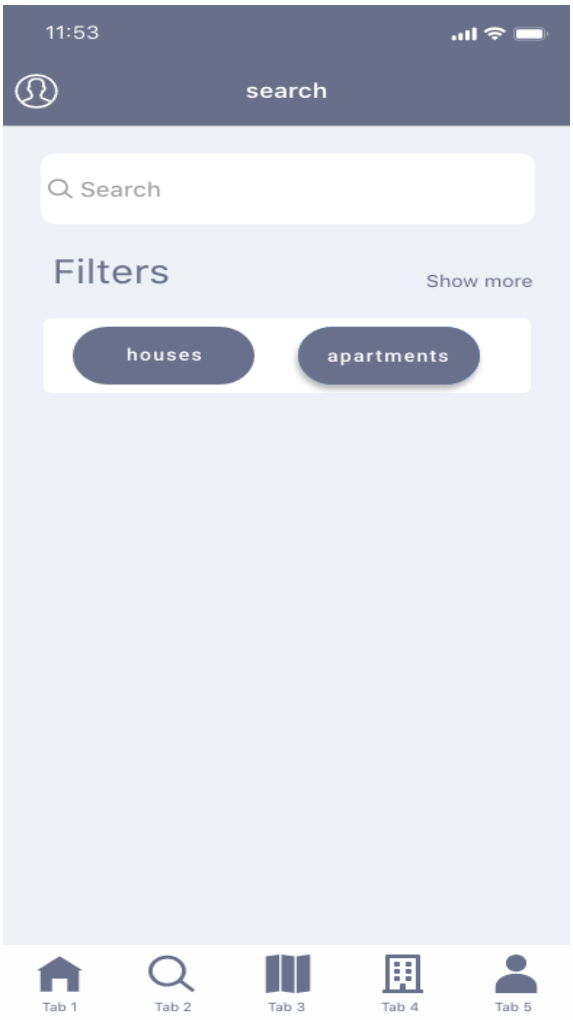


Figure – Search Interface

6. Requirements Matrix

	P001	P002	P003	P004	P005
UC001	X				
UC002	X				
UC002		X			
UC004				X	
UC005			X		
UC006		X			
UC007					X
UC008			X		

Appendix C Software Testing Documentation

Software Testing Document

Project Title

DESIGNING AND IMPLEMENTING OF MULTI-PLATFORM REAL ESTATE MANAGEMENT SYSTEM

Version 1.0

Department and Faculty

Software Engineering

Prepared by:

Arivan Shamal M. Zyad

Revision Page

a. **Overview**

The content is about designing and implementing a multi-platform real estate management system. The current version is 1.0.

b. **Target Audience**

The target audience for the Designing and Implementing of Multi-Platform Real Estate Management System includes real estate professionals and clients interested in the real estate industry.

c. **Project Team Members**

Arivan Shamal M. Zyad.

d. **Version Control History**

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**
<Current Version>	**Arivan Shamal** **M. Zyad**		

1. Introduction

1.1 Purpose

The purpose of this Software Test Documentation (STD) is to provide a comprehensive overview of the testing activities for the designing and implementing of multi-platform real estate management systems. This software engineering project aims to develop a user-friendly platform for customers to search and find properties, with features such as a mapping system and virtual tour for easy navigation and visualization. The system will be utilized by both admin and agents to manage properties and communicate with customers. This STD will provide a description of the test procedures and results to evaluate the system's ability to meet its functional and non-functional requirements, including performance, scalability, user-friendliness, and security.

The evaluation will also assess the system's compliance with relevant regulations and standards, ease of use, and accessibility. The intended audience for this STD is the software engineering student and anyone involved in the development and testing of the designing and implementing of multi-platform real estate management systems. The goal of this project is to produce a robust and flexible real estate management system that enhances the customer experience and improves the efficiency of property management.

1.2 Scope

The software product is a comprehensive real estate management system with the goal of providing an accessible and user-friendly platform for customers to browse, search, and find properties. The system will offer a mapping system and virtual tour feature to allow for easy navigation and visualization of properties.

Additionally, the system will have a login portal for both agents and administrators, allowing for seamless communication and property management.

The scope of the project will be focused on developing a solution that meets the functional and non-functional requirements, including performance and scalability, user-friendliness, security, and compliance with relevant regulations and standards. The system will also be designed to be easy to use and accessible to a wide range of users.

The main objective of the project is to improve the customer experience in searching for properties and streamline the property management process for agents and administrators. The benefits of the project include increased efficiency, better customer satisfaction, and a more robust and flexible real estate management system.

This project will be evaluated based on its ability to meet the functional and non- functional requirements and its overall impact on the real estate industry. The result is expected to be a cutting-edge real estate management system that sets a new standard for the industry and provides significant benefits for all stakeholders involved.

1.3 **Definitions, Acronyms and Abbreviation**

Definitions of all terms, acronyms and abbreviation used are to be defined here.

STD - Software Design Documentation

1.4 **References**

Fewster, M., & Graham, D. (1999). Software Test Automation: Effective Use of Test Execution Tools. Boston, MA: Addison-Wesley Professional.

Ligman, G. (2019). Advanced Software Testing: Volume 2: Guide to the ISTQB Advanced Certification as an Advanced Test Manager (3rd ed.). Rocky Nook.

Black, R. (2006). Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional. Raleigh, NC: Pragmatic Bookshelf.

Beizer, B. (1990). Software Testing Techniques. New York, NY: Van Nostrand Reinhold.

1.5 **System Overview**

Software testing documentation typically includes two key components: Test Cases and Test Approach Analysis. Test Cases are detailed instructions outlining specific steps and conditions for testing, ensuring that the software functions as intended and meets requirements. They cover various types of testing and serve as a guide for accurate and consistent testing. Test Approach Analysis involves evaluating the most

effective testing strategy considering project requirements, resources, and risks. It outlines the overall testing strategy, techniques, and key focus areas to optimize test

efforts and minimize software issues. Both Test Cases and Test Approach Analysis contribute to comprehensive testing and ensuring the quality of the software

QAIWAN INTERNATIONAL UNIVERSITY	Form No.:LIB003 Edition: 01 Effective Date: 01/ 06/ 2024 Page (s): 03
FINAL THESIS SUBMISSION FORM	

Section 1 (To be completed by Student)
(Please tick (√) where applicable)

Library
Qaiwan International University

Submission of Final Copies of Thesis

I Arivan shamal M. Zyad (name) hereby submit:
the final copies of my thesis for the degree of [/] Bachelor / [ ] Master / [ ] Doctor of Philosophy

The title of the thesis is:

Designing and Implementing of Multi-Plat
form Real Estate Management system

Hence, I declare that:

- [/] i) My thesis has been reviewed by my Main Supervisor / HoD / Dean / Faculty
- [/] ii)I had submitted the CD of my softcopy of the thesis (the contents are similar with the hardcopy of the thesis) in a single PDF file.
- [/] iii)I had submitted the Degree Scroll Information Form.

Student Name: _____

Department: Software Engineering

Signature: _____

Date: 11/8/2024

ENDORSEMENT BY MAIN SUPERVISOR

Section 2 (To be completed by Main Supervisor)

Supervisor's Name: Mr. Salam

Department: S.E

I have examined the thesis of Mr. /Mrs. /Ms.: Arvin Shamal

I hereby confirm that all corrections and amendments made to the thesis have been rectified by the candidate.

Dr.Shahab

(Main Supervisor's Signature & Stamp)

(Date)

ENDORSEMENT BY Librarian

Section 3 (To be completed by Librarian)

Librarian's Name: Dlkhwaz Othman

☑ Approved ☐ Not Approved

I hereby confirm that all corrections and amendments made to the thesis have been rectified by the candidate.

(Librarian's Signature & Stamp)

11/8/2024

(Date)

STUDENT'S DECLARATION

Section 4 (To be completed by Student)

Student Thesis's Copyright and University's Intellectual Property

"The copyright to a thesis belongs to the student. However, as a condition of being awarded the degree, the student hereby grants to the University, a free, ongoing, non-exclusive right to use the relevant work and/or thesis for the University's teaching, research and promotional purposes as well as free and the non- exclusive right to retain, re-produce, display and distribute a limited number of copies of the thesis, together with the right to require its publication for further research and archival use."

I declare that the contents presented in this thesis are my own which was done at Qaiwan International University unless stated otherwise. The thesis has not been previously submitted for any other degree.

I also declare that my thesis has been reviewed by the Main Supervisor / HoD / Dean / Faculty the comments are as stated in Section 2 (page 2).

Name of Student: Aviran Shamal M. Zyad

Signature of Student: .....

IC. No. / Passport No.: .....

Date: 11/8/2024