

# The Development of a Property Management System for MrPropertyMan

Siti Hajar Munirah<sup>1</sup>, Nurezayana Zainal<sup>1\*</sup>

<sup>1</sup> *Fakulti Sains Komputer dan Teknologi Maklumat,  
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

\*Corresponding Author: [nurezayana@uthm.edu.my](mailto:nurezayana@uthm.edu.my)

DOI: <https://doi.org/10.30880/aitcs.2025.06.02.068>

## Article Info

Received: 16 June 2025

Accepted: 20 November 2025

Available online: 30 November 2025

## Keywords

Property, Decision Tree, Booking,  
Web-Based, Management System,  
RAD, Centralized Database.

## Abstract

The increasing reliance on internet technologies for property transactions highlights the need for a centralized and efficient real estate management system. This study aims to develop the MrPropertyMan Property Management System, a web-based platform designed to streamline operations and improve user experience for admin, agents, and customers. The system uses an object-oriented design approach and adopts the Rapid Application Development (RAD) methodology, integrating modules such as property management, financial analysis, booking, and communication tools. Key findings indicate that the Budget Analysis Module, utilizing a decision tree algorithm, enhances property recommendations based on financial criteria. At the same time, secure user authentication and real-time data updates optimize decision-making. The system reduces inefficiencies and improves workflow, enhancing productivity and customer satisfaction. Future work could explore incorporating artificial intelligence for predictive analytics and expanding functionalities to support larger-scale operations.

## 1. Introduction

The increasing reliance on internet technology has significantly changed how people search for and transact in the real estate market. Today, potential buyers and renters mostly begin their property search online, making web-based platforms crucial for property management companies to remain competitive. As stated by Liu & Xu [1], this shift in behavior needs the implementation of robust and user-friendly digital solutions to fulfill the demands of modern customers. Recognizing this trend, MrPropertyMan aims to develop a centralized property management system to address current inefficiencies and enhance the overall user experience.

Currently, MrPropertyMan uses multiple platforms such as WhatsApp, Telegram, and Instagram for various operations, including client communication, property promotions, and administrative tasks. However, the reliance on separate tools results in disorganized workflows, inefficient operations, and unsatisfactory customer engagement. The absence of a centralized system makes it challenging for admin to manage data effectively, agents to track property listings accurately, and customers to access information seamlessly. These limitations highlight the urgent need for an integrated platform to streamline processes, improve efficiency, and elevate the overall management experience.

This study aims to design and develop the MrPropertyMan Property Management System and test its functionality through user acceptance testing. The system targets three key user groups which are admin, agents, and customers with customized modules to meet user needs. Features such as the Budget Analysis

Module, which utilizes a decision tree algorithm, aim to provide property recommendations based on customers' financial capacity, enhancing their decision-making process.

The following section discusses the related works and explores similar systems. Section 3 details the methodology employed for the system's design and development. Section 4 discusses the system's implementation and results, while Section 5 concludes with key findings, limitations, and recommendations for future work.

## 2. Literature Review

A property management system is a centralized platform to streamline property-related business operations. Currently, MrPropertyMan relies on manual processes, which are inefficient and prone to errors. To enhance the proposed system's development, three web-based systems which are Biddo, PropertyGuru, and Met Property were analyzed to gather valuable insights and best practices.

### 2.1 Study of MrPropertyMan Manual Process

MrPropertyMan, a small real estate company in Tanjung Lumpur, Kuantan, relies on manual processes for daily operations, including client interactions, property matching, loan eligibility checks, and transaction management. Fig. 1 shows the AS-IS model for the existing business process for MrPropertyMan company.

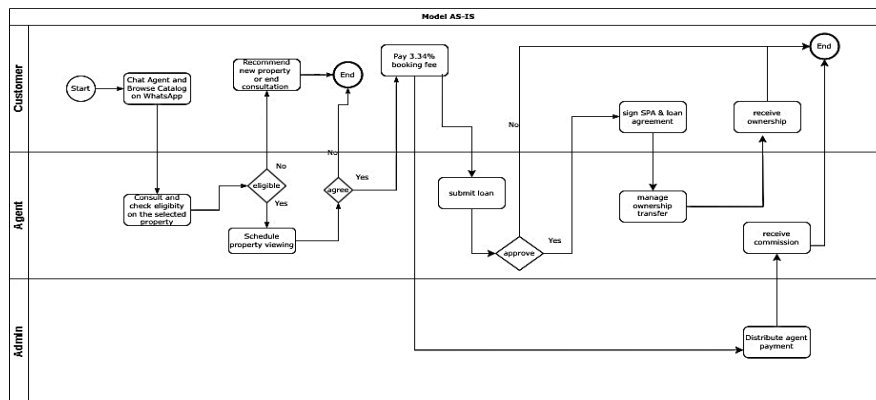


Fig. 1 AS-IS model for the existing business process

Under the current conventional approach, customers typically contact property agents through WhatsApp, where catalogs are shared manually. This process often involves prolonged back-and-forth communication to agree on property deals. Meanwhile, agents are required to manually document client information, manage bookings, and handle paperwork, all of which are time-consuming, prone to human error, and inefficient. These challenges emphasize the need for a centralized, automated system to streamline daily operations, reduce errors, and enhance service delivery.

### 2.2 Biddo: Property Rental Bidding System

Biddo simplifies finding, bidding on, and securing rental properties. Users can register and search properties by keywords, type, location, and place bids. They can digitally sign tenancy agreements, download PDFs, and make secure payments if successful. The platform streamlines the experience by eliminating manual paperwork and in-person processes.

### 2.3 PropertyGuru

PropertyGuru is a comprehensive platform for property search, covering Malaysia's major cities and rural areas. Users can filter listings based on criteria like location and price, access detailed property information, and utilize tools like a mortgage calculator. The platform also connects users with agents, offers property guides, and provides market insights to support informed decision-making.

### 2.4 Met Property

Met Property focuses on property transactions in Selangor and Kuala Lumpur. It offers tools for buying, selling, and renting properties, with features like search filters and agents communication. The platform includes a News, Guides, and Insights section to inform users of the real estate market, investment tips, and legal considerations. The systems studied were Biddo, PropertyGuru and Met Property.

### 2.5 Comparison with Existing System

The comparison between the three systems and the proposed system is presented in Table 1. This section highlights the key features of each system, providing valuable insights for evaluation.

**Table 1** System Comparison

Features/System	MrPropertyMan	PropertyGuru	Met Property	Biddo
Register and Log in Module	√	√	√	√
Profile Module	√	√	√	√
Manage Agent Module	√	√	√	√
Manage Property Module	√	√	√	√
Booking Module	√	X	X	X
Budget Analysis Module	√	X	√	X
Chat Module	√	X	X	X
Generate Report Module	√	√	√	√

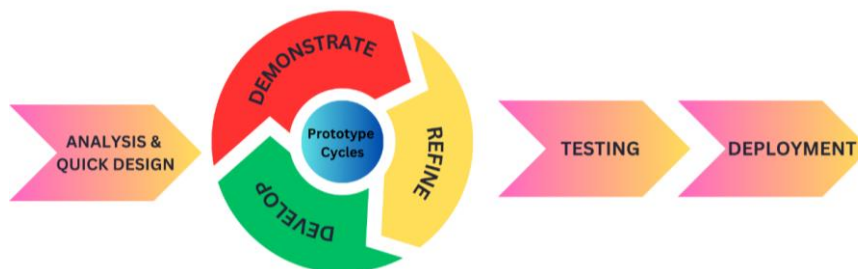
From Table 1, Nine features have been selected for comparison with the developed system, including "Register and Log in", "Booking", "Profile", "Manage Agent", "Manage Property", "Budget Analysis", "Chat", "Generate Report", and others sourced from the four different systems. According to the table, all four existing systems which are PropertyGuru, Met Property, and Biddo, lack a "Booking" feature, while MrPropertyMan includes this functionality. In the proposed system, customers are required to register and log in before making appointments. This requirement ensures that only authorized users can access the booking system, preventing unauthorized access.

Additionally, only MrPropertyMan includes the "Budget Analysis" module, which helps clients analyze their budget for property-related transactions. The "Chat" feature is available only in MrPropertyMan, enabling direct communication between agents and clients. Furthermore, all four systems include a "Generate Report" feature, which allows users to generate reports for better management and decision-making. Unlike the existing systems, the proposed system will offer more comprehensive functionalities, such as managing agents, properties, and customer profiles seamlessly.

### 3. Methodology

Rapid Application Development (RAD), introduced by James Martin in the 1990s, focuses on flexibility, fast development, and cost-effectiveness [2]. It reduces planning time and emphasizes core processes like development, testing, and collaboration, overcoming the limitations of traditional models like Waterfall. RAD's iterative cycles improve software quality and reduce development costs. The MrPropertyMan Property Management System adopts RAD due to its ability to incorporate user feedback, promote early testing, and ensure adaptability. The selection of a methodology significantly impacts on project outcomes, influencing elements like cost, timeline, and overall quality [3]. Therefore, the prototype model is the chosen software process model for developing the proposed system. RAD consists of four phases: Analysis and Quick Design, User Design, Testing, and Deployment, each focusing on gathering requirements, prototyping, validating functionality, and launching the system. Fig. 2 shows the Rapid Application Diagram.

Meanwhile, Table 1 outlines four phases, task related and their outputs. In Analysis and Quick Design, the project is proposed, requirements are identified, and a timeline is developed. The User Design phase involves creating the system prototype, database, and collecting feedback. Testing focuses on user acceptance testing and refining the system. Finally, in Deployment, the system is tested, configured, and deployed, with outputs including the final implementation and user evaluation.



**Fig. 2** Rapid Application Diagram (RAD)

**Table 2** *Project Phase Overview*

Phase	Tasks	Outputs
Analysis and Quick Design	<ul style="list-style-type: none"> <li>Proposed the project</li> <li>Identify requirements</li> <li>Identify problems</li> <li>Identify goals, objectives and scope</li> <li>Develop and specify timeline</li> </ul>	<ul style="list-style-type: none"> <li>Project proposal</li> <li>List of user requirement</li> <li>Analysis of user needs</li> <li>Gantt Chart (refer appendix A)</li> </ul>
User Design	<ul style="list-style-type: none"> <li>Design system prototype</li> <li>Create database structure</li> <li>Get feedback</li> <li>Update design</li> <li>Develop project</li> </ul>	<ul style="list-style-type: none"> <li>System prototype</li> <li>Database framework</li> <li>Relational schema and Data Dictionary</li> <li>Codebase and source file</li> <li>Stakeholder feedback</li> <li>Implemented system</li> </ul>
Testing	<ul style="list-style-type: none"> <li>User acceptance testing</li> <li>Refine based on feedback</li> </ul>	<ul style="list-style-type: none"> <li>Test case</li> <li>Test report</li> <li>Updated test case</li> </ul>
Deployment	<ul style="list-style-type: none"> <li>Test final product</li> <li>System configuration</li> </ul>	<ul style="list-style-type: none"> <li>Final implementation</li> <li>User evaluation</li> </ul>

Table 2 shows project phase overview. The project began with the Analysis & Quick Design phase, where two meetings with stakeholders and additional online research were conducted to gather requirements. This resulted in a project proposal, a list of user requirements, an analysis of user needs, and a Gantt chart for planning. In the User Design phase, the system prototype was developed, the database structure was created, and a relational schema and data dictionary were established. Feedback from stakeholders led to refinements in the design. The project then proceeded through multiple prototype cycles of development, demonstration, and refinement.

The system was demonstrated to stakeholders three times, with each demonstration followed by necessary adjustments based on feedback. In the Testing phase, test cases were prepared, and user acceptance testing was conducted for three roles: admin, agents, and customer. The system was demonstrated to Mr. Aiman as the admin and Mrs. Sara as the agent at Mr Property Man company located in Kuantan. Both were presented with a form to fill out, evaluating whether the system passed or failed based on their interactions. For the customer role, a Google Form was created for users to submit their responses. This form evaluated the overall system, assessing whether it met their needs and was user-friendly. Finally, in the Deployment phase, the final product was tested, system configurations were set, and the final implementation was completed. A user evaluation was conducted to ensure the system's readiness for deployment.

## 4. Analysis and Design

System analysis and design (SAD) is a multi-stage process that includes planning, requirement analysis, development, design, and testing. System analysis aims to understand user demands and the system, while system design aims to address those needs. During this phase, system development is broken down into smaller sub-activities that work together to achieve the overall goal of system development.

### 4.1 System Requirements Analysis

Requirement analysis is a technique that determines user expectations from the proposed system. During system development, system requirements are developed to represent a few types of requirements such as user requirements, system requirements, functional requirements, and non-functional requirements [4]. All functional and non-functional requirements are documented in Table 3 and Table 4.

**Table 3** *Functional requirements*

No	Module	Description
1	Register and login account Module	<ul style="list-style-type: none"> <li>• Allow customer to create account.</li> <li>• Admin and agent have their own username and password</li> <li>• Allow all users to login into account using username and password</li> </ul>
2	Profile Module	<ul style="list-style-type: none"> <li>• Allow agents to manage profile, such as personal data, listed properties.</li> <li>• Allow customer to manage their profile such as personal data, saved properties, and inquiries.</li> </ul>
3	Manage Agent Module	<ul style="list-style-type: none"> <li>• Allow admin to edit agent information such as create agents, update status, register ID and password, salary and bonus.</li> </ul>
4	Manage Property Module	<ul style="list-style-type: none"> <li>• Allow agent to create listing property, edit status property, type property, location and status.</li> </ul>
5	Booking Module	<ul style="list-style-type: none"> <li>• Allow agent to update real time status and manage booking.</li> <li>• Allow customer to book the date with agents.</li> <li>• Allow agents and customer to check the past booking.</li> </ul>
6	Budget Analysis Module	<ul style="list-style-type: none"> <li>• Allow to calculate all the property calculation such as booking fee, legal fee, stamping fee, downpayment fee and others.</li> <li>• Provide recommendation of property by using decision tree algorithm through calculation and other related assessments, suggesting suitable properties that align with the customer's budget and financial capability.</li> <li>• Allow to save the calculation result.</li> <li>• Allow to load past calculation.</li> </ul>
7	Chat Module	<ul style="list-style-type: none"> <li>• Allow agents to reply and interacts with customer inquiries.</li> <li>• Allow customer to ask question at the chat provided.</li> </ul>
8	Report Module	<ul style="list-style-type: none"> <li>• Provide statistical report for admin and agent.</li> <li>• Allow admin to view property reports, agent performance and others.</li> <li>• Allow agent to view sales and booking reports, property listing and others.</li> </ul>

**Table 4** *Non-functional requirements*

No	Requirement	Description
1	Performance	The app should provide smooth operation with no noticeable delays in core functionalities.
2	Operational	The loading time required for an application is no more than 1 minute.
3	Security	The system should be user friendly
4	Usability	The system interface should be user-friendly for both customers and admin.

### 4.2 Use Case Diagram

A use case diagram is a visual representation of the interactions between a system and its external actors. This diagram illustrates the software's functional requirements and simplifies the system's functionality [5]. Fig. 3 shows the use case diagram.



Fig. 3 Use case diagram

The use case diagram for the MrPropertyMan Property Management System illustrates the interactions between three user roles which are Customer, Admin, and Agent, the system's core functions. Key features include registering and logging in, managing profiles, agents, properties, bookings, chats, and generating reports, as well as performing budget analysis. Customers have access to all functions, Admins manage all except budget analysis, while Agents can use most features except agent management.

### 4.3 System Architecture

A system architecture represents the conceptual model of a system together with models derived from it that represent (1) different viewpoints defined as views on top of the conceptual model, (2) facets or concerns of the system in dependence on the scope and abstraction level of various stakeholders, (3) restrictions for the deployment of the system and description of the quality warranties of the system, and (4) embeddings into other systems [6]. Fig. 4 demonstrates the system architecture of the proposed system.

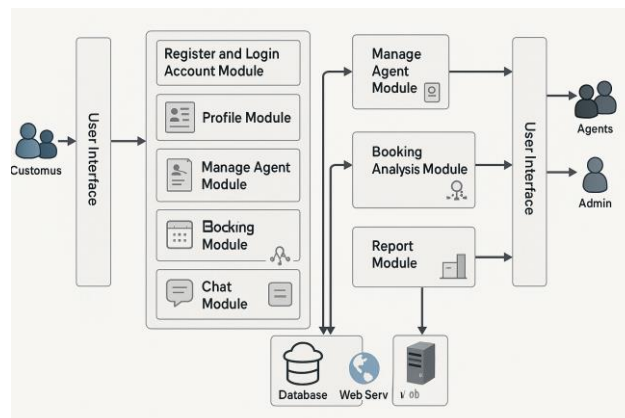


Fig. 4 System Architecture of the Proposed system

In figure 4, the system architecture for the MrPropertyMan Property Management System is a web-based platform that integrates core modules such as registration, profile management, property listing, booking, chat, budget analysis, and reporting for three main user roles which are customers, agents, and administrator. Each module connects through a centralized user interface to a web server and database, streamlining operations, enhancing communication, and improving overall property management efficiency.

#### 4.4 Class Diagram

Class diagrams are used when developing an object-oriented system model to show the classes in a system and the associations between these classes [7].

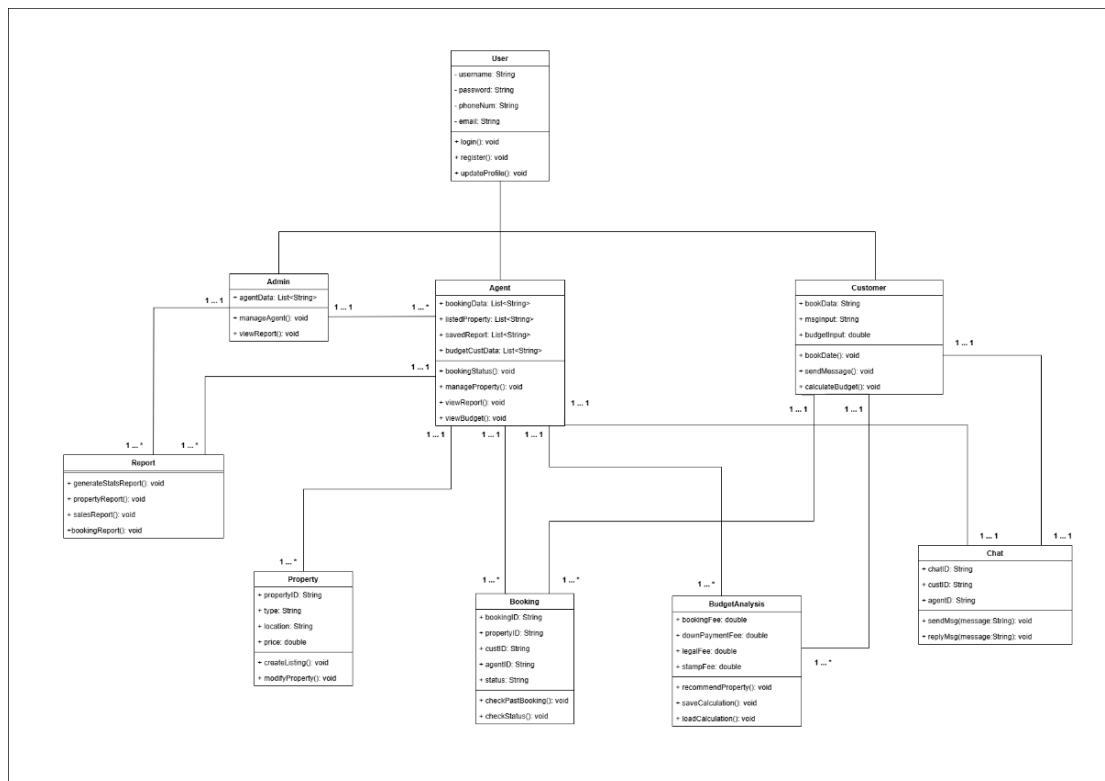
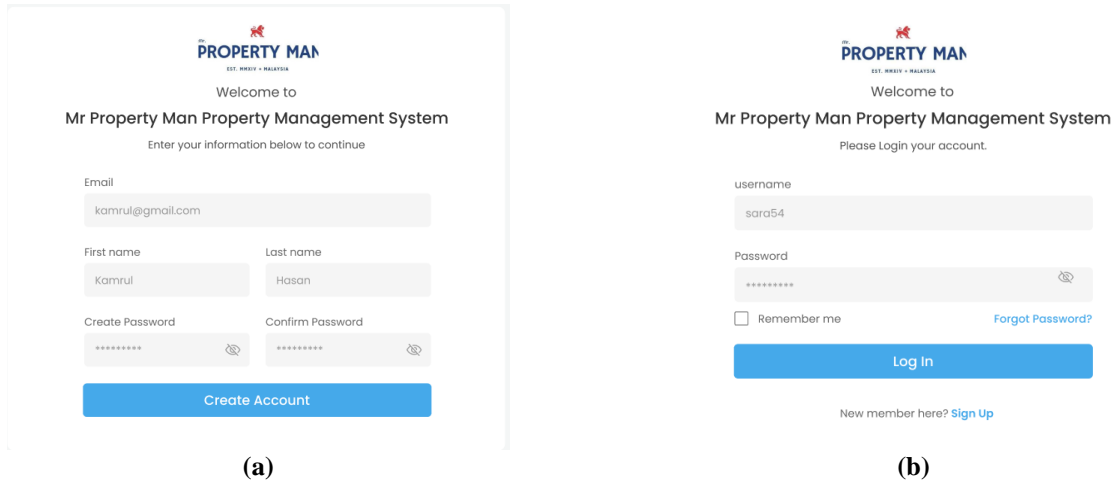


Fig. 5 Class diagram

In figure 5, the system's main inheritance structure centers around the User superclass, which branches into Admin, Agent, and Customer subclasses. The User class includes attributes like username, password, phone number, and email, with methods for login, registration, and profile updates. Admins manage agents and reports, while Agents handle property listings, bookings, and customer interactions. Customers focus on booking and budgeting. Supporting classes include report for listings and financial customer data, property for managing listings, booking linking customers to agents, budget analysis for loan eligibility, and chat facilitating communication between customers and agents. All entities are interconnected to ensure clear relationships.

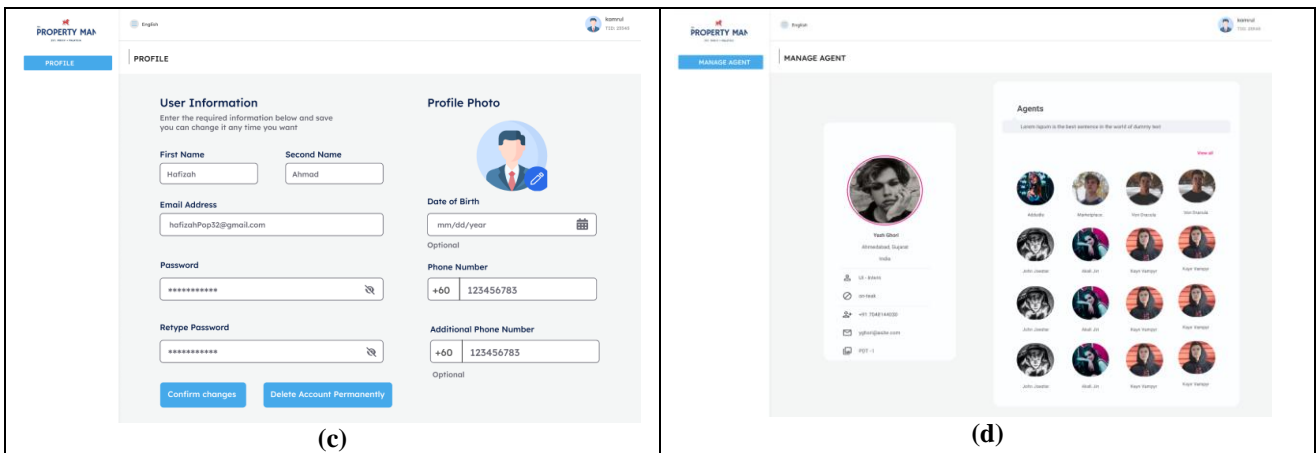
#### 4.5 Interface Design

Interface design is the process of crafting how users interact with a system, focusing on the look, feel, and overall user experience. It involves careful planning of layout, information structure, and navigation flow to ensure intuitive and efficient use. A well-designed interface should be visually clear, user-friendly, and responsive across devices, enabling users to accomplish tasks smoothly and effectively.



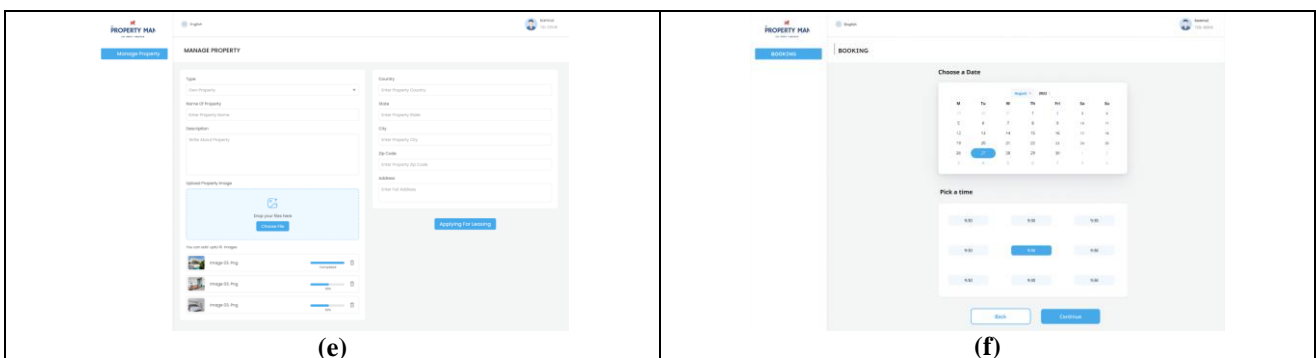
**Fig. 6** Figure (a) register; (b) login

Fig. 6a shows the registration screen, where new users enter their email, name, and password to create an account. Fig. 6b displays the login screen, where existing users can enter their username and password to access their account, with options for password recovery and account creation for new users.



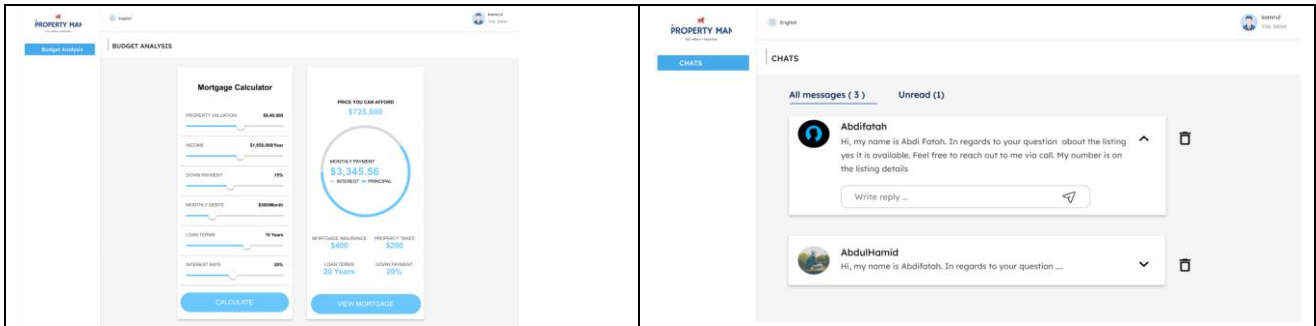
**Fig. 7** Figure (c) profile; (d) manage agent

Fig. 7c displays the Profile interface, where users can update their personal details, change passwords, and manage their account settings. Fig. 7d shows the Manage Agent section, where users can view and manage the profiles of agents within the platform.



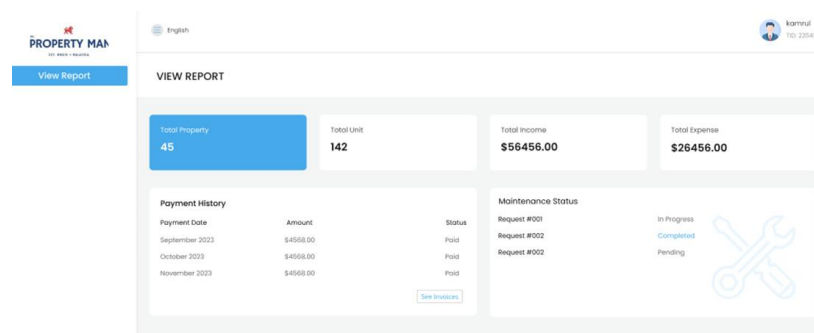
**Fig. 8** Figure (e) manage property; (f) booking

Fig. 8e displays Manage Property interface, allowing users to update property details such. It also has options for uploading new property images and applying for listing. Fig. 8f shows the Booking interface, where users can select a date and time for booking, with a calendar and time slots displayed. The user can then proceed by clicking the Next or Confirm buttons, completing the booking process.



**Fig. 9** Figure (g) manage property; (h) booking

Fig. 9e displays Mortgage Calculator interface, allowing users to input details and interest rate to calculate their mortgage payments. The calculator also displays an estimated price, providing users with insights into their potential financial commitment. Fig. 9f presents the Chats interface, where users can view and manage their messages with property agents. It displays the messages, enabling users to easily communicate with agents regarding property inquiries and updates.



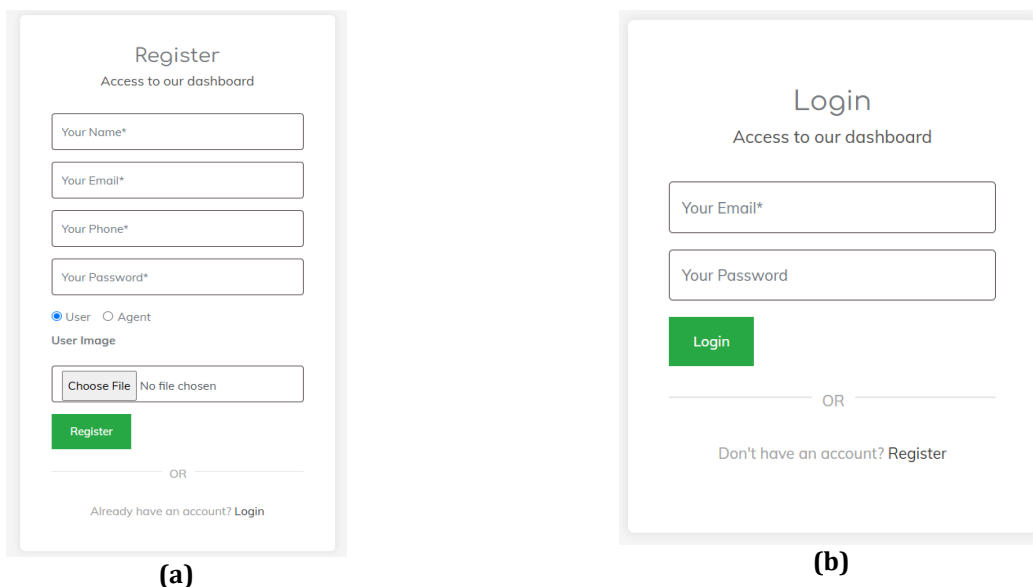
**Fig. 10** Report

Fig. 10 displays View Report interface, displaying reports generation for admin to view and save.

## 5. Implementation and Testing

Implementation and Testing are critical phases in development into real-world environments. During implementation, the system is configured and integrated with modules requirements and followed by testing phases to identify and resolve any issues to ensure system’s reliability and performance.

### 5.1 Implementation



**Fig. 11** Figure (a) register; (b) login

Fig. 11a the registration form for dashboard access, where users provide their name, email, phone, password, and upload a user image, while selecting whether they are a User or Agent, and fig. 11b shows the login form, which requires an email and password for users to access their dashboard or register if they don't have an account.

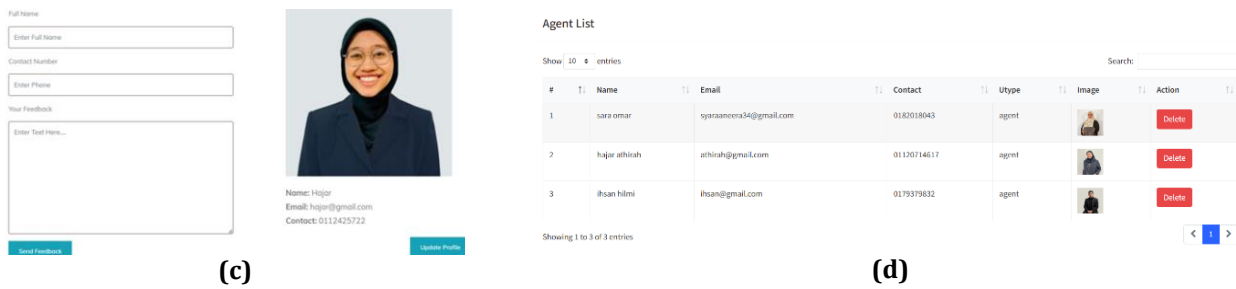


Fig. 12 Figure (c) profile; (d) manage agent

Fig. 12c shows the profile update page, where users can edit their personal information and feedback, and fig.12d shows the agent management page, which lists agents with options to delete or manage their details.

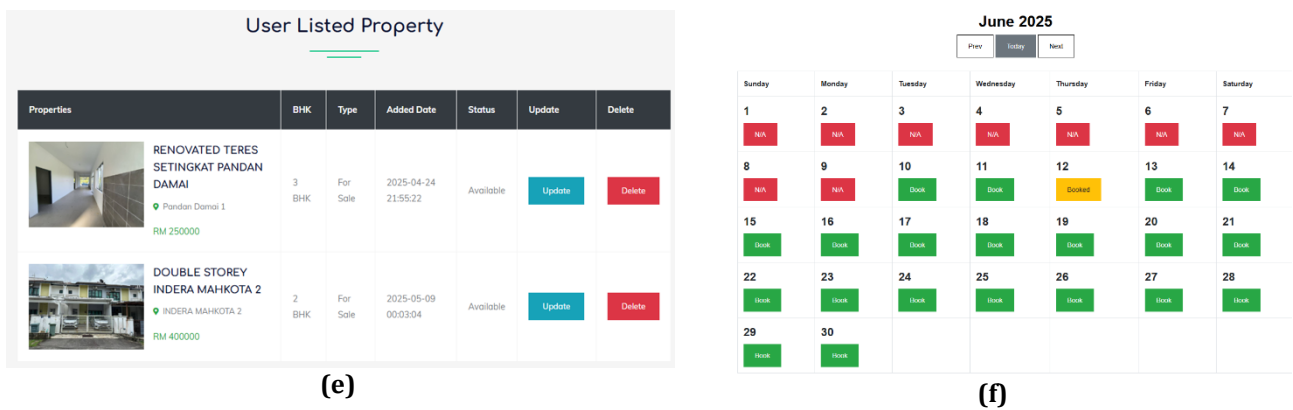


Fig. 13 Figure (e) manage property; (f) booking

Fig. 13e shows the Manage Property page, where users can view, update, or delete their property listings. Fig. 13f displays the Booking page, which features a color-coded calendar to track property availability, with red for unavailable dates, green for available dates, and yellow for partially booked days.

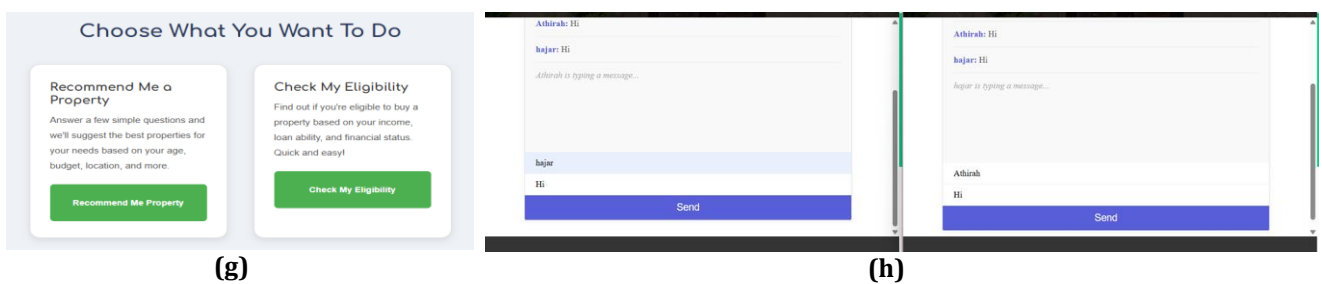
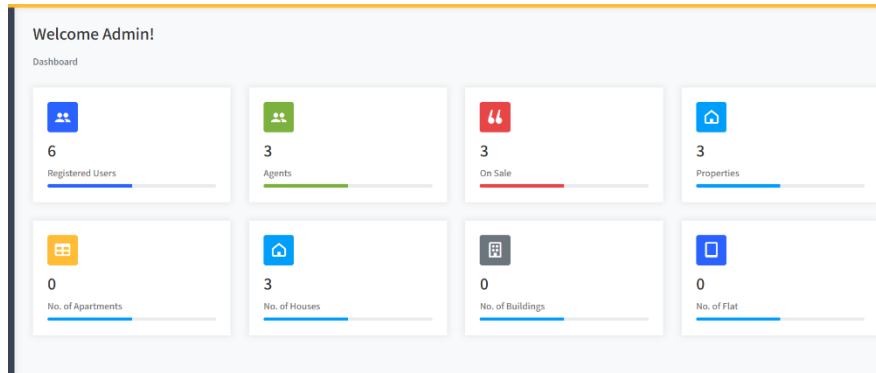


Fig. 14 Figure (g) budget analysis; (h) chat

Fig. 14g displays a budget analysis interface where users can likely manage or review customers' financial plans for property selection. Fig. 14h shows a chat interface with options for both customer and agent, facilitating communication and message exchanges within the system.



**Fig. 15 Report**

Fig. 15 represents a report interface. It summarizes key data or performance metrics for the platform. This includes user registrations and property listings.

## 5.2 Testing

This section outlines the development of the system's functional modules, detailing the purpose and implementation of each component. Program code snippets are included to aid in understanding the logic and structure of the system. These examples serve to illustrate how specific functionalities are achieved, providing insight into the overall system architecture and development process.

### 5.2.1 Test Cases

The test cases are divided into 8 categories, corresponding to each module of the system, ensuring comprehensive coverage and verification of all specified requirements. This systematic approach helps track whether each requirement has been met. The detailed test cases and their results are presented in Tables 8 through 18.

**Table 4 Test Case for all Module**

Software Requirement Specification	Test Case ID	Test Case Description	Test (Pass/Fail)
<b>UC001 Test Case for Register and Login</b>			
SRS_REQ_1001	TEST_1001	The system shall allow users/agents to register with valid data.	Pass
SRS_REQ_1002	TEST_1002	The system shall allow users/agents to login with valid credentials.	Pass
SRS_REQ_1003	TEST_1003	The system shall restrict login with incorrect credentials.	Pass
SRS_REQ_1004	TEST_1004	The system shall allow users to reset forgotten passwords.	Fail
<b>UC002 Test Case for Profile Management</b>			
SRS_REQ_2001	TEST_2001	The system shall allow agents/customers to view and update their profile.	Pass
SRS_REQ_2002	TEST_2002	The system shall restrict invalid profile data updates.	Fail
SRS_REQ_2003	TEST_2003	The system shall allow admin to view but not	Pass

update their profile.

**UC003 Test Case for Manage Agent (Admin)**

SRS_REQ_3001	TEST_3001	The system shall allow admin to delete agents.	Pass
SRS_REQ_3002	TEST_3002	The system shall allow admin to search for agents.	Pass
SRS_REQ_3003	TEST_3003	The system shall allow admin to view agent list.	Pass

**UC004 Test Case for Manage Property**

SRS_REQ_4001	TEST_4001	The system shall allow agents to create new property listings.	Pass
SRS_REQ_4002	TEST_4002	The system shall allow agents to update property details.	Pass
SRS_REQ_4003	TEST_4003	The system shall allow agents to delete property listings.	Pass
SRS_REQ_4004	TEST_4004	The system shall display newly submitted listings on property listing page.	Pass
SRS_REQ_4005	TEST_4005	The system shall restrict invalid property data.	Pass

**UC005 Test Case for Manage Booking**

SRS_REQ_5001	TEST_5001	The system shall allow customers to create bookings.	Pass
SRS_REQ_5002	TEST_5002	The system shall allow agents to update booking status.	Pass
SRS_REQ_5003	TEST_5003	The system shall notify dedicated agents about bookings.	Pass
SRS_REQ_5004	TEST_5004	The system shall allow agents to view booking status.	Pass

**UC006 Test Case for Budget Analysis & Eligibility Check**

SRS_REQ_6001	TEST_6001	The system shall allow customers to perform budget analysis.	Pass
SRS_REQ_6002	TEST_6002	The system shall allow customers to perform eligibility checks.	Pass
SRS_REQ_6003	TEST_6003	The system shall recommend properties based on budget analysis.	Pass
SRS_REQ_6004	TEST_6004	The system shall allow users to save analysis results.	Pass

SRS_REQ_6005	TEST_6005	The system shall allow users to retrieve past analysis results.	Pass
<b>UC007 Test Case for Manage Chat</b>			
SRS_REQ_7001	TEST_7001	The system shall allow customers to send messages to agents.	Pass
SRS_REQ_7002	TEST_7002	The system shall allow agents to respond to customer inquiries.	Pass
SRS_REQ_7003	TEST_7003	The system shall save and retrieve past chat history.	Fail
<b>UC008 Test Case for Generate Report</b>			
SRS_REQ_8001	TEST_8001	The system shall allow admin to print reports.	Pass
SRS_REQ_8002	TEST_8002	The system shall allow admin to copy reports.	Pass
SRS_REQ_8003	TEST_8003	The system shall allow admin to search within reports.	Pass

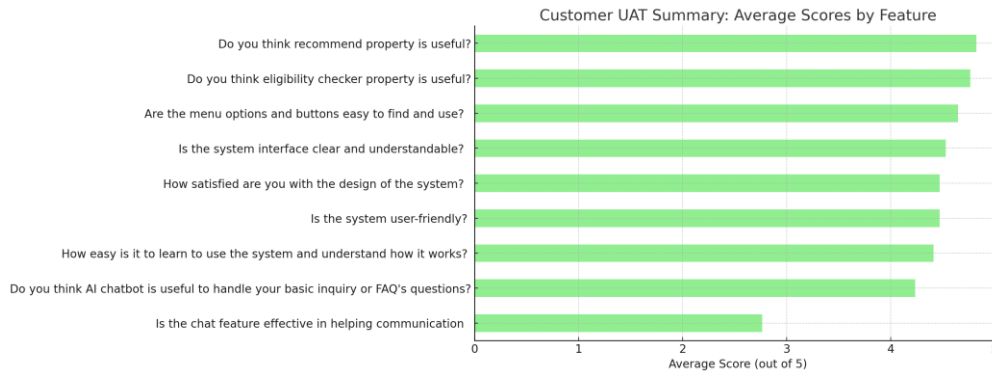
From table 4, the system was tested across eight core modules with 29 test cases executed. A total of 25 tests passed, 4 failed. Key modules such as Manage Agent, Manage Property, Manage Booking, and Generate Report achieved 100% pass rates, showcasing solid core functionality. Failures were observed in the Forgot Password, Profile Validation, and Chat History features. The overall pass rate is 86.2%.

### 5.2.2 User Acceptance Testing

User acceptance testing was conducted for three roles: admin, agents, and customer. The system was demonstrated to Mr. Aiman as the admin and Mrs. Sara as the agent at Mr Property Man company located in Kuantan. Both were presented with a form to fill out, evaluating whether the system passed or failed based on their interactions. For the customer role, a Google Form was created for users to submit their responses. This form evaluated the overall system, assessing whether it met their needs and was user-friendly.

The system has successfully passed all key tests (Refer Appendix B), however, stakeholders has suggested several areas for improvement in their remarks. The agent management process on the admin side needs improvement, as agents can register without verification. Some property data fields should also be reviewed. For budget analysis, data should be sent only to the admin. On the agent side, the property management section should allow adding a WhatsApp link instead of just a phone number. Some property data is underutilized by agents and should be refined. The booking functionality should let agents update their availability. Lastly, stronger integration with WhatsApp is recommended for customer chats.

For customers, a total of 17 respondents, randomly selected from the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM), participated in the user acceptance testing (UAT) for the MrPropertyMan Property Management System (Refer Appendix C). The participants were primarily students who are familiar with digital platforms and represent a segment of the intended user base, such as customers and agents. Feedback was collected through a structured online questionnaire consisting of 14 questions, which included a mix of Yes/No responses and 1–5 scale ratings to assess usability, functionality, and overall satisfaction with the system. Fig. shows the average scores for each feature based on responses from customers in the Google Form questionnaire.



**Fig. 14** Average scores by feature

On average, 94% of users agreed that the system is user-friendly and visually clear, with most users giving high scores (4 and above) for ease of navigation and feature accessibility. Features like the AI chatbot and financial eligibility checker were found useful by approximately 82% of respondents. All users were able to perform basic tasks such as registration, login, profile updates, and booking without significant issues. However, the chat feature received lower ratings, averaging around 3.0 out of 5, indicating potential areas for improvement. Overall, the UAT results show a high level of user satisfaction, suggesting that the system meets most user expectations and functional requirements.

## 6. Conclusion

In conclusion, the MrPropertyMan system successfully meets its objectives by offering a centralized, user-friendly platform for managing property-related tasks. It simplifies the process for administrators, agents, and customers through secure login, real-time bookings, smart budget analysis, and built-in communication tools. Key advantages include faster decision-making, integrated property recommendations, and AI-assisted support. However, some limitations remain, such as the lack of support for legal processes like SPA signing, no saved chat history, and potential security vulnerabilities. Future improvements could include stronger data protection, message storage, and wider coverage beyond Kuantan. With continuous development, MrPropertyMan has strong potential to become a reliable and efficient property management solution in Malaysia.

## Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia and Arif Sufian Enterprise for its support.

## Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

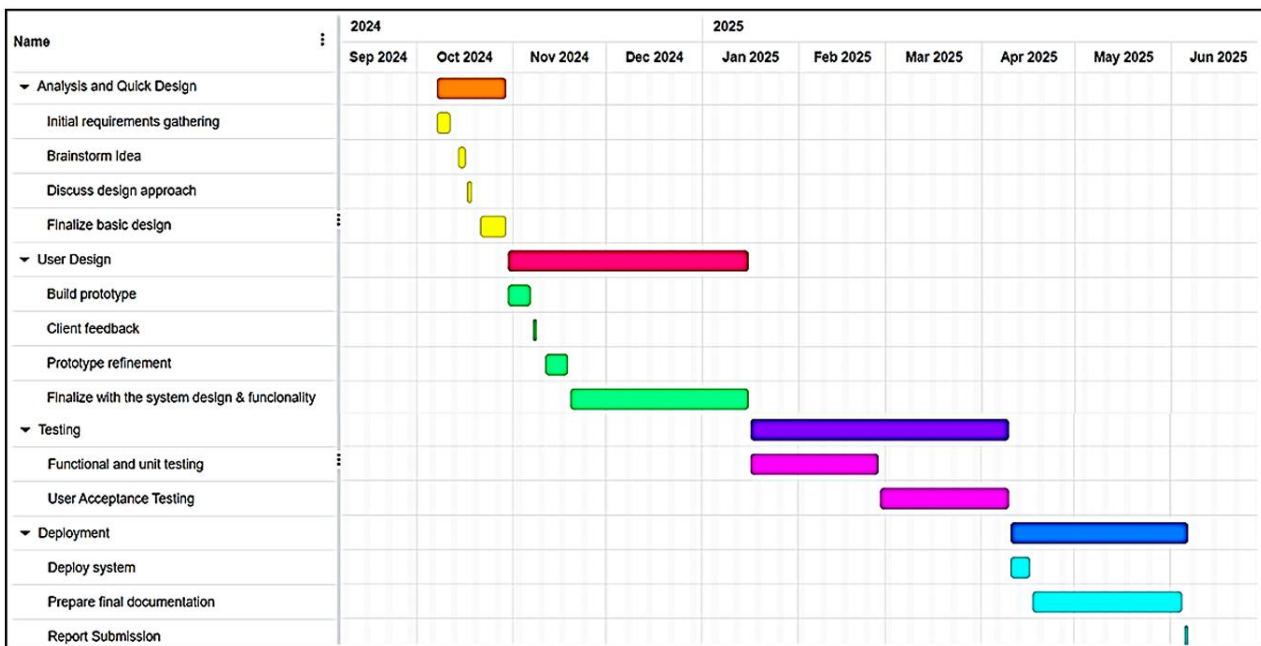
## Author Contribution

The authors confirm contribution to the paper as follows: study conception and design: Siti Hajar Munirah Binti Kamarulzaman, Dr. Nurezayana Zainal; data collection: Siti Hajar Munirah Binti Kamarulzaman; analysis and interpretation of results: Siti Hajar Munirah Binti Kamarulzaman, Dr. Nurezayana Zainal; draft manuscript preparation: Siti Hajar Munirah Binti Kamarulzaman, Dr. Nurezayana Zainal. All authors reviewed the results and approved the final version of the manuscript.

## References

- [1] Liu, D. T., & Xu, X. W. (2001). A review of web-based product data management systems. *Computers in Industry*, 44(3), 251–262. [https://doi.org/10.1016/s0166-3615\(01\)00072-0](https://doi.org/10.1016/s0166-3615(01)00072-0)
- [2] F. Fatima, M. Javed, F. Amjad, and U. G. Khan, “An Approach to Enhance Quality of the Rad Model Using Agents”, *theijst*, vol. 2, no. 13, Dec. 2014.
- [3] B. W. Boehm, “A spiral model of software development and enhancement,” *Computer*, vol. 21, no. 5, pp. 61–72, May 1988, doi: <https://doi.org/10.1109/2.59>
- [4] P. Loucopoulos and V. Karakostas, *System Requirements Engineering*. London, U.K.: McGraw-Hill, 1995.
- [5] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, “A different approach on automated use case diagram semantic assessment,” *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 1, pp. 496–505, 2020, doi: [10.22266/ijies2021.0228.46](https://doi.org/10.22266/ijies2021.0228.46).
- [6] H. Jaakkola and B. Thalheim, "Architecture-Driven Modelling Methodologies," in *Information Modelling and Knowledge Bases*, vol. XXII, A. Heimbürger, Ed. Tampere, Finland: IOS Press, 2011, pp. 97–116.
- [7] I. Sommerville, *Software Engineering*, 9th ed. Boston, MA, USA: Addison-Wesley, 2011.

## Appendix A: Gantt Chart



### Appendix B: User Acceptance Testing (Admin & Agent)

**USER ACCEPTANCE TESTING (UAT)**  
MRPROPERTYMAN PROPERTY MANAGEMENT SYSTEM

Name : MUHAMMAD AIMAN HARITH BIN KAMARUZAMAN  
Date : 8/6/25  
Phone Number : 012 9818 983  
Company : MR PROPERTY MAN

No.	Module/Acceptance Criteria	Test Result		Remark (optional)
		Pass	Fail	
1.	<b>Register and Login</b>	Pass	Fail	
a)	There is a register and login page	/		
b)	Able to register and login	/		
c)	Redirect to homepage after login	/		
d)	Redirect to login after register	/		
e)	Error message shown if wrong credentials entered	/		
f)				
g)				
h)				
i)				
2.	<b>Profile</b>			
a)	Customer and agent can view profile information	/		improve for invalid info
b)	Customer and agent can update profile information	/		
c)	Customer and agent can send feedbacks	/		
d)				
3.	<b>Manage Agent</b>			
a)	Admin can view list of agents	/		improve for registration agent
b)	Admin can delete agents	/		(need unique ID from company to verify)
c)	Admin can search agents	/		
d)				
e)				
4.	<b>Manage Property</b>			
a)	Agents can add property	/		the data can be reviewed again / some don't need
b)	Agents can submit property	/		
c)	Agents can edit property	/		
d)	Agents can delete property	/		
e)	User can search and filter property	/		
f)	User can view property	/		
g)	User can...	/		
h)				

5.	<b>Booking</b>			
a)	Customer can book a property viewing	/		
b)	Customer can edit	/		
c)	Customer can delete	/		
d)	Agent can view booking specifically	/		
e)				
6.	<b>Budget Analysis</b>			
a)	Customer can fill details information	/		1. must send data to admin not agent.
b)	Customer can check eligibility	/		
c)	Customer can view last calculation	/		
d)	Customer can save calculation	/		
e)	Customer able to see recommendation	/		2. add per month house payment
7.	<b>Chat</b>			
a)	User can ask general question using AI chatbot	/		
b)	User can send message	/		
c)	Agent receive message	/		
d)	Agent can reply message	/		
e)				
8.	<b>Report</b>			
a)	Admin can access reports	/		
b)	Admin can print reports	/		
c)	Admin can copy reports	/		
d)				
e)				

I am \_\_\_\_\_ hereby declare that the information provided is true and correct.

Agreed by: \_\_\_\_\_ Witnessed by: \_\_\_\_\_

Name: MUHAMMAD AIMAN HARITH KAMARUZAMAN      Name: SITI HANIK  
Position: EEO MR PROPERTYMAN      Position: DEVELOPER  
Date: 8/6      Date: 8/6

**USER ACCEPTANCE TESTING (UAT)**  
MRPROPERTYMAN PROPERTY MANAGEMENT SYSTEM

Name : WAN NORHAISRA BINTI WAN OMAR  
Date : 8/6/2025  
Phone Number : 016-2018042  
Company : MR PROPERTY MAN

No.	Module/Acceptance Criteria	Test Result		Remark (optional)
		Pass	Fail	
1.	<b>Register and Login</b>	Pass	Fail	
a)	There is a register and login page	/		
b)	Able to register and login	/		
c)	Redirect to homepage after login	/		
d)	Redirect to login after register	/		
e)	Error message shown if wrong credentials entered	/		
f)				
g)				
h)				
i)				
2.	<b>Profile</b>			
a)	Customer and agent can view profile information	/		
b)	Customer and agent can update profile information	/		
c)	Customer and agent can send feedbacks	/		
d)				
3.	<b>Manage Agent</b>			
a)	Admin can view list of agents	/		
b)	Admin can delete agents	/		
c)	Admin can search agents	/		
d)				
e)				
4.	<b>Manage Property</b>			
a)	Agents can add property	/		1. please add whatsapp link
b)	Agents can submit property	/		to specific agent
c)	Agents can edit property	/		
d)	Agents can delete property	/		2. The property data some not familiar
e)	User can search and filter property	/		
f)	User can view property	/		
g)				
h)				

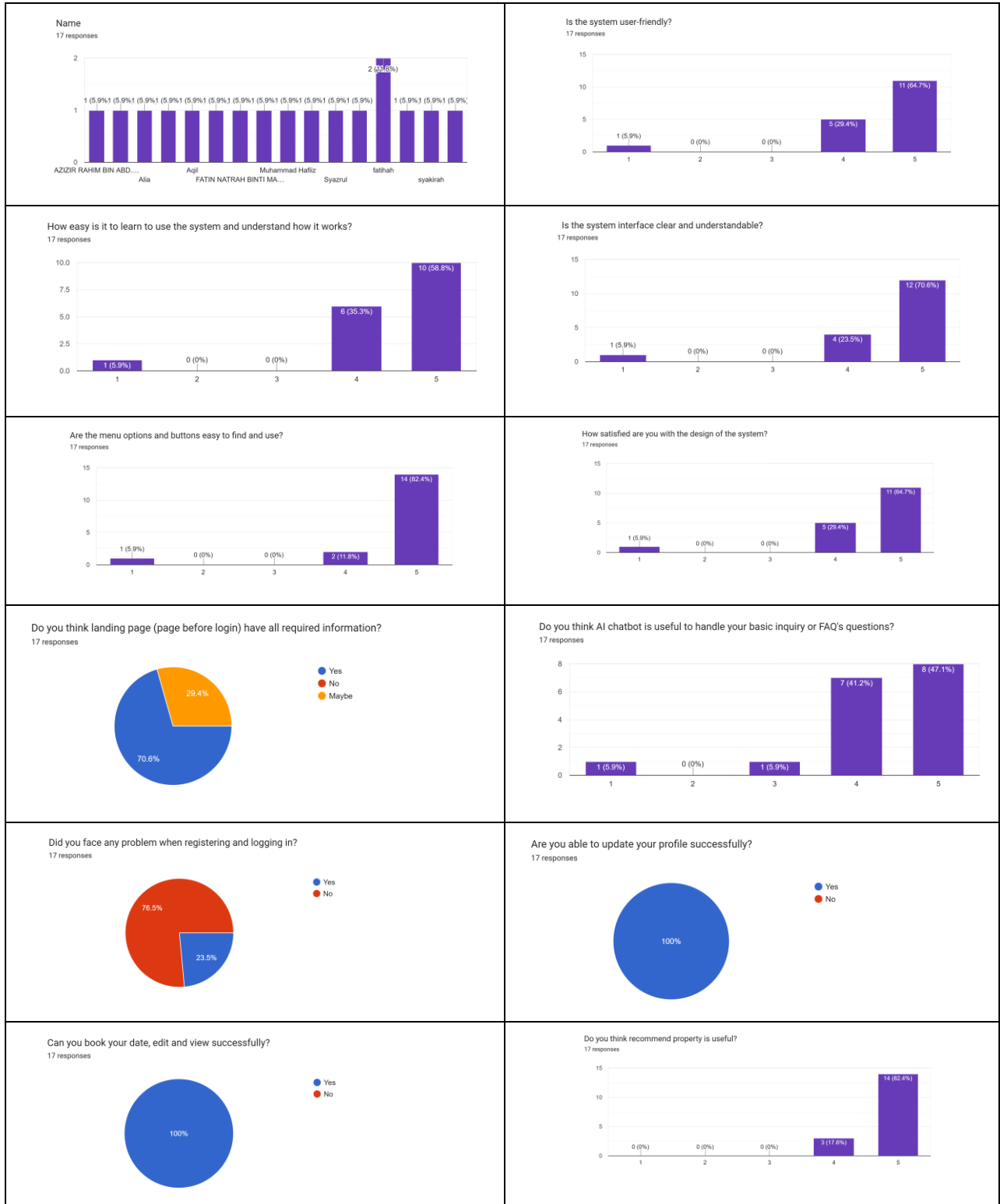
5.	<b>Booking</b>			
a)	Customer can book a property viewing	/		improve: agent can update their unavailable time.
b)	Customer can edit	/		
c)	Customer can delete	/		
d)	Agent can view booking specifically	/		
e)				
6.	<b>Budget Analysis</b>			
a)	Customer can fill details information	/		
b)	Customer can check eligibility	/		
c)	Customer can view last calculation	/		
d)	Customer can save calculation	/		
e)	Customer able to see recommendation	/		
7.	<b>Chat</b>			
a)	User can ask general question using AI chatbot	/		usually customer prefer use whatsapp.
b)	User can send message	/		
c)	Agent receive message	/		
d)	Agent can reply message	/		
e)				
8.	<b>Report</b>			
a)	Admin can access reports	/		
b)	Admin can print reports	/		
c)	Admin can copy reports	/		
d)				
e)				
f)				

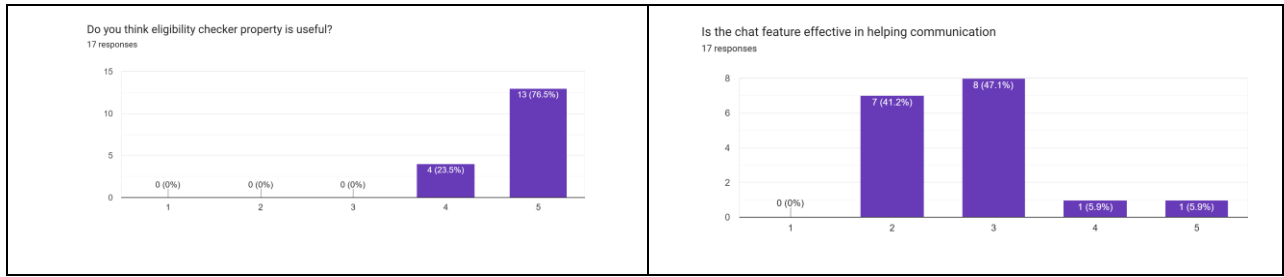
I am WAN NORHAISRA BINTI WAN OMAR hereby declare that the information provided is true and correct.

Agreed by: \_\_\_\_\_ Witnessed by: \_\_\_\_\_

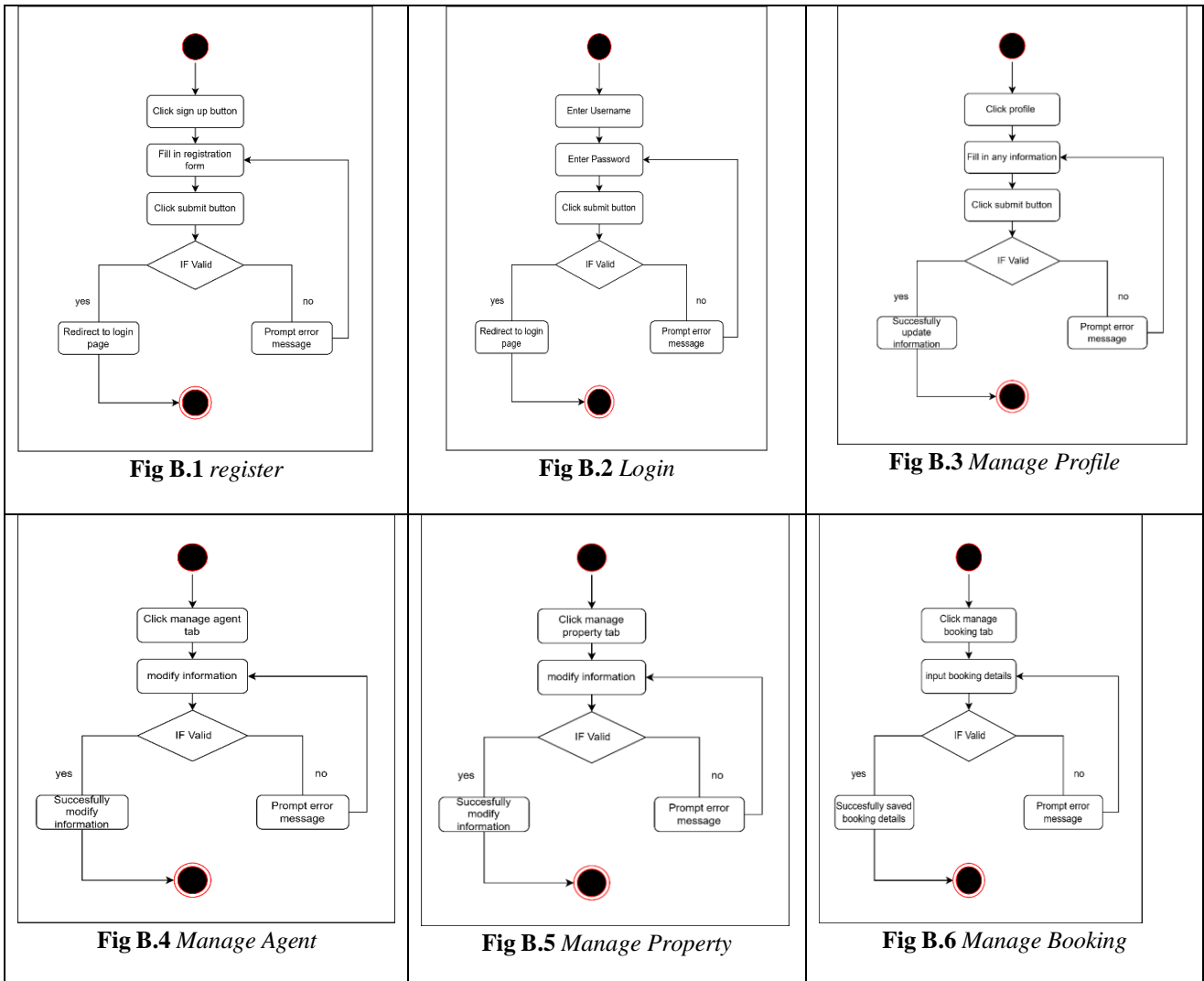
Name: WAN NORHAISRA BINTI WAN OMAR      Name: SITI HANIK  
Position: AGENT      Position: DEVELOPER MR PROPERTYMAN  
Date: 8/6/2025      Date: 8/6

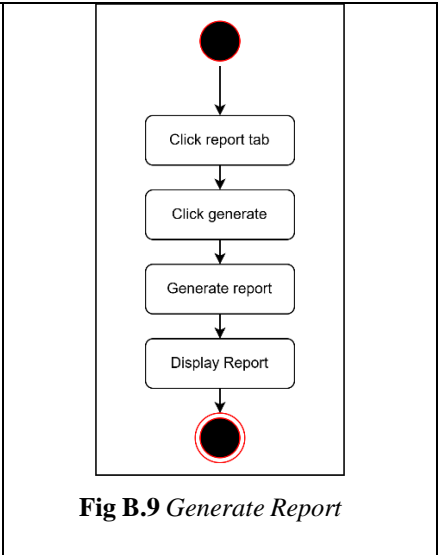
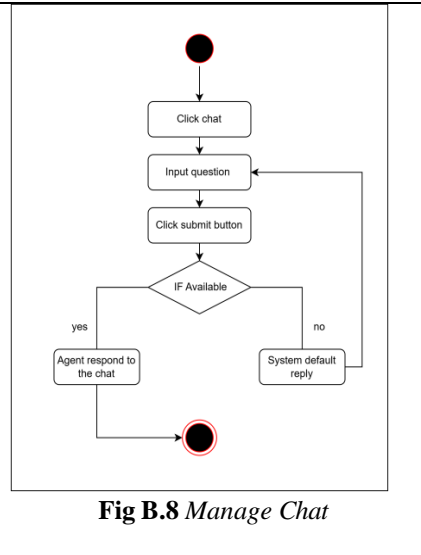
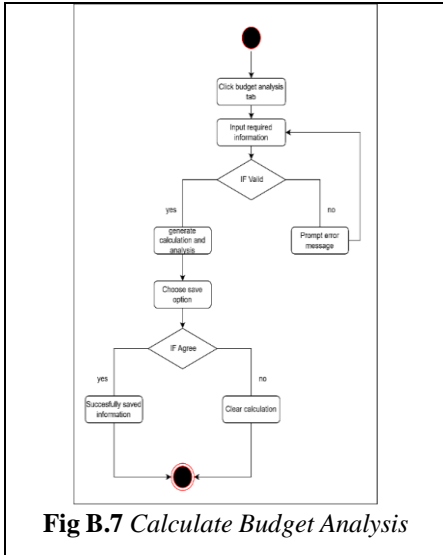
### Appendix C: User Acceptance Testing (Customer)



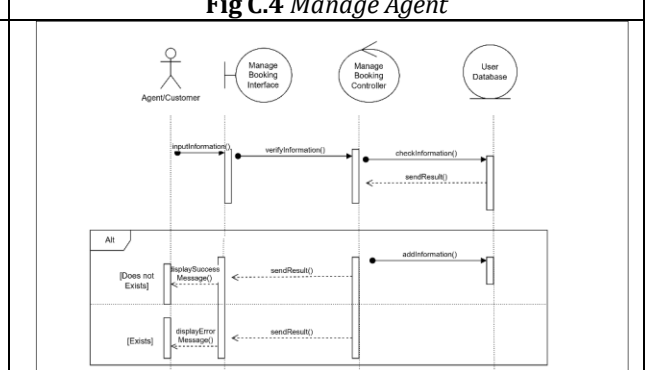
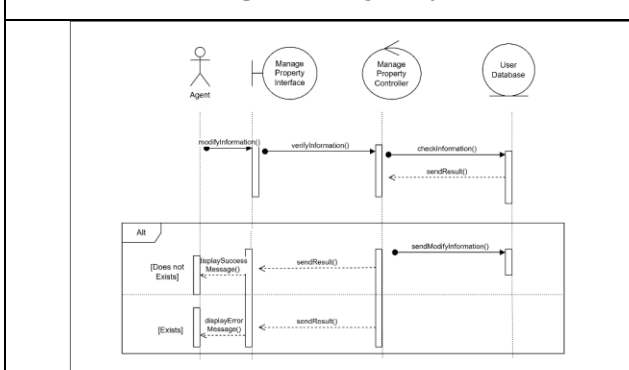
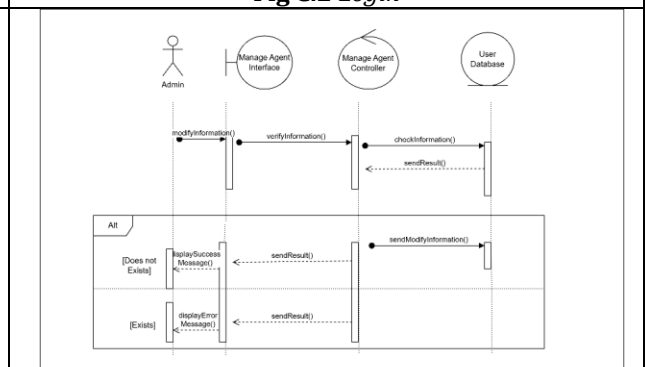
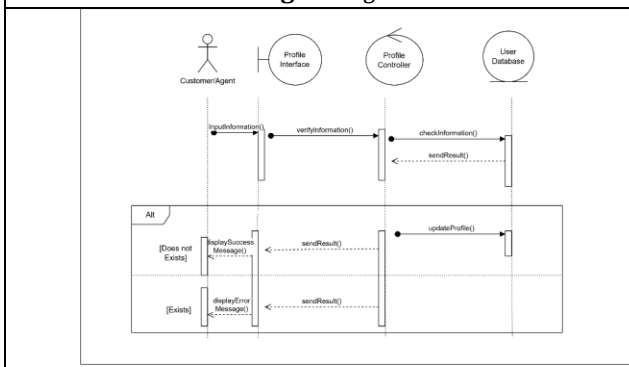
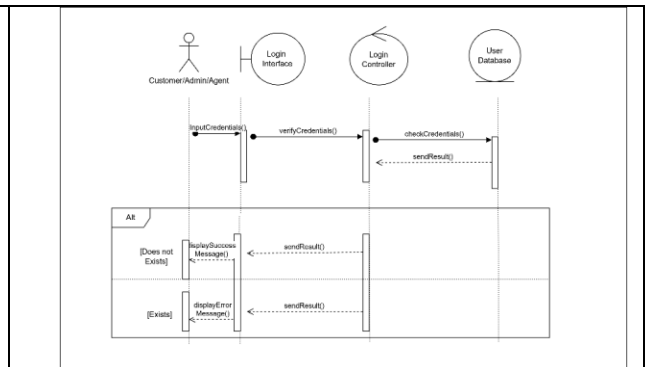
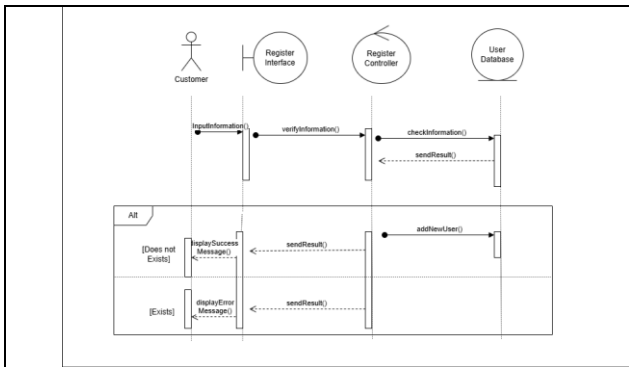


### Appendix D: Activity Diagram





**Appendix E: Sequence Diagram**



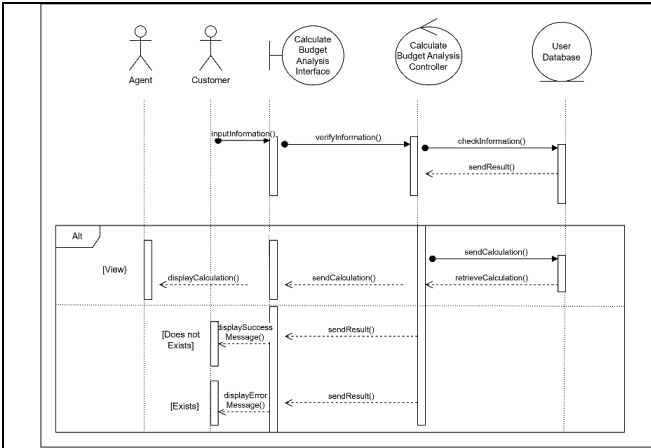


Fig C.7 Calculate Budget Analysis

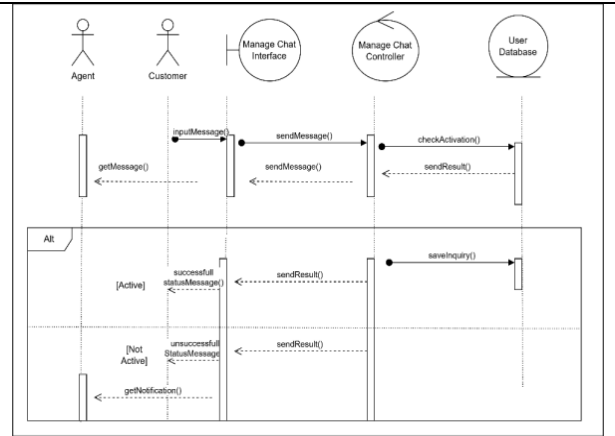


Fig C.8 Manage Chat

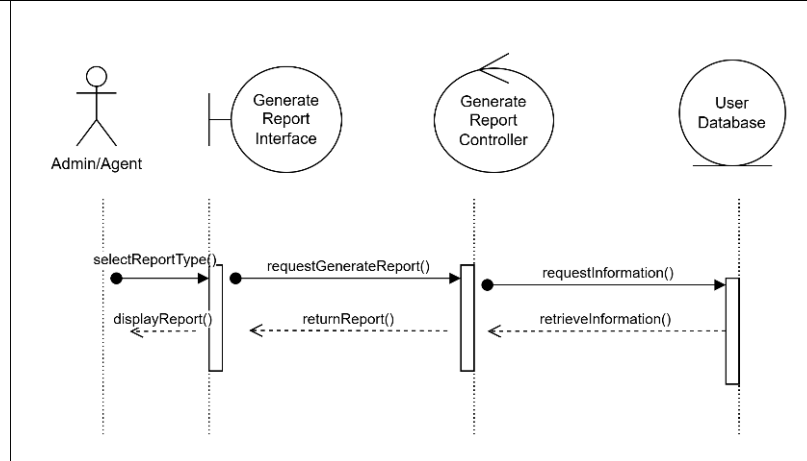


Fig B.9 Generate Report