APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 5 No. 2 (2024) 1065-1083
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Rental House System for UTHM Students

Yap Kit Yeng[1], Mohd Hamdi Irwan Hamzah[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: hamdi@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2024.05.02.056

Article Info

Received: 24 July 2024
Accepted: 21 October
Available online: 15 December 2024

Keywords

Rental Management System, Web Based System, Interative Model

Abstract

In Parit Raja, Johor, UTHM students who are not living in the university dormitory will need to rent houses around UTHM for academic purposes. The existing rental process is time-consuming since students need to go out off campus to find suitable house and may face problem of incomplete detail of the house and the landlords. The objectives of this web-based platform are to design a rental house system using an object-oriented approach, to develop a rental house system using a web-based approach and to test the functionality and usability of the developed rental house system. This study used an iterative model of the software development life cycle to systematically design and develop the system. By comparing the system with existing systems, the system has unique features that set it apart from existing products. Ultimately, the system will enhance the housing search experience for UTHM students while providing administrative convenience for landlords and administrators. The Rental House System for UTHM Students successfully achieves the objectives and solved the problem facing by the stakeholders.

1. Introduction

A rental house system is a web-based project that allows users to easily access all rental housing near their location [1]. Finding suitable and affordable housing near a university can be a hard task, especially for students unfamiliar with the surrounding area. In the current process of students renting a house or a room, students need to go out and find the house that is available to rent then contact the owner or agent of the house. Then, if the students are satisfied with renting the house, the students will inform the owner to discuss payment details and the date of entry into the rental house[2]. UTHM students are often challenged with limited housing market options and obstacles to the rental process.

The Rental House System for UTHM Students comes may provide UTHM students with a digital solution that simplifies and streamlines their housing search. On this platform, students can browse verified rental information, including comprehensive descriptions, visuals, and the ability to schedule an appointment with a landlord. With this system, students can search for housing based on their financial constraints, preferred type of housing, and desired amenities, thus increasing the efficiency and transparency of the entire process. However, the advantages of the system go beyond convenience. The system addresses the anxiety students face by providing important information about potential landlords and properties. Verified listings and secure communication channels replace deceptive practices and untrustworthy agents as a thing of the past. Students can choose legitimate agents with confidence.

This web-based system is developed with an object-oriented approach and provides UTHM students with the necessary resources to make informed decisions about their living arrangements. With a user-friendly

interface, secure payment options, and a specialized complaint system, this rental system provides a comprehensive solution for all stakeholders - students, landlords, and administrators. The system specifically addresses the challenges UTHM students face in the housing market, providing them with a more connected, knowledgeable and less stressful housing experience. For UTHM students, finding more than just an apartment is a comfortable home.

This paper is divided into several sections. Section 1 is the introduction to the project. Section 2 explores related work. By gaining a deeper understanding of the project, Section 2 explores the literature review and comparison with other existing systems. In Section 3, the methodology used for system development is scrutinized by describing each phase and its corresponding deliverables. Section 4 will provide a comprehensive analysis and design of the system through the implementation of Unified Modeling Language (UML) diagrams. Finally, Section 5 will elaborate on the conclusions of this document, summarizing the main findings of the project and providing recommendations for further system improvements.

2. Related Work

In this section, to provide a deeper understanding of the developed system, the current process of rental property will be discussed. Besides that, since this is a web-based system, then the web-based system concept will be explained in this section. Next, the Rental House System for UTHM system will be developed with the study and analysis of the existing related systems. Thus, the comparison between the existing systems with the developed system is conducted and displayed in the table.

2.1 Project Domain of rental property in Parit Raja

There are very few online resources for UTHM students looking for available rent houses near UTHM in the Parit Raja area. In the existing rental house system such as Mudah.my iProperty and PropertyGuru, there are very few houses displayed in Parit Raja area since there are very few landlords who advertised their properties in these systems. Thus, many UTHM students experience difficulties in finding rental property online. Thus, they often need to drive and find a rental property near the campus by themselves. There is no centralized system that manage the rent house near the UTHM campus. After that, when these students find an available rental property, they need to contact the landlord or agent for further information such as monthly fee, deposit, maximum number of tenants and so on. After negotiations, they will book a time to view the actual conditions of the house.

There are various problems of existing process of rental, the most significant problems are the process of driving to search for available rentals is very time-consuming and there is a risk of being scammed as the banners do not show the details of the house and the owner and the information is not verified. As a result, a web-based rental house system is developed to solve the existing problem that may be faced by the UTHM students in the current process in this project.

2.2 Web-based system

In the current renting process, the UTHM students exchange all related information with landlord directly and this kind of communication is time-consuming. Then, web technology can help people find information easier rather than traditional way of finding information[3]. Thus, a web-based system can help UTHM students to find the information regarding the rent house near them. They can search, view, and filter the rent houses that fulfilled their criteria. Web-based systems can function on devices that connect with the internet without carrying out any downloading. A web-based system is usually developed by using HTML, CSS, and JavaScript since these three languages are the most common and easiest to use language used in developing web-based system.

2.3 Study of the Existing Related Systems

In this section, the Rental House System for UTHM system will be developed with the study and analysis of the existing related systems. The following sections will give a deep detailed description of related systems which are Mudah.my iProperty and Property Guru. The comparison between the existing systems with the developed system is conducted based on the capability of the system to identify the similarities and differences between the existing systems. Table 1 shows the comparison between the selected systems which are Mudah.my, iProperty, PropertyGuru and Rental House System for UTHM students. After that, a description of the table will be explained after Table 1

Table 1 *Comparison between Existing Systems and Developed System*

Modules	Mudah.my	iProperty	PropertyGuru	Rental House System for

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1065-1083

				UTHM Students
User Login and registration	✓	✓	✓	✓
Manage account details	✓	✓	✓	✓
Manage house	✓	✓	✓	✓
Rent house	✓	✓	✓	✓
Manage payment and agreement	X	X	X	✓
Manage complaint	X	X	X	✓
Generate report	X	✓	✓	✓
Platform	Web-based and mobile application	Web-based and mobile application	Web-based and mobile application	Web-based

Based on Table 1, there are several modules in the existing system same as the developed system. The same modules can be used as references while developing the developed system. However, there are also some different modules between the existing systems and the developed system. For example, the three existing systems do not provide the manage payment and agreement module and manage complaint module. Thus, the Rental House System for UTHM Students is different from the existing system by providing manage payment and agreement module and manage complaint module. These modules can help to enhance the connection between the students and landlords. As a conclusion, the Rental House System for UTHM Students can help the students, administrators, and the landlords in handling the rental process. The platforms of the three existing systems are web and mobile application whereas the platform for the developed system only has a web-based platform.

3. Methodology

Methodology is a description of the underlying assumptions that led to the selection of specific methods to carry out the project [4] A methodology description usually begins by defining and describing the methodology that has been chosen to study a particular research question. It then discusses the steps used in selecting, collecting, and processing data in an organized manner. This section explains the methodology that has been conducted for developing the developed system, Rental House System for UTHM Students. Figure 1 shows the Iterative model and Table 2 shows task and deliverables in each of the development phases in the Iterative model.

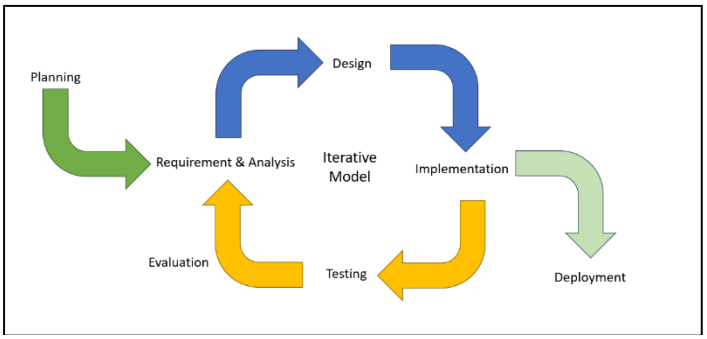


Fig. 1 *Iterative Model*

Table 2 *System development phase task and Deliverable*

Phase	Task/Activities	Deliverable
Planning	• Identify project title. • Identify problems, objectives, scopes and expected outcome. • Identify stakeholders. • Generate project workflow. • Generate project timeline in Gantt Chart • Identify current process of rental house	• Proposal • Gantt Chart

Table 2 *(cont)*

Phase	• Task/Activities	• Deliverable
Requirement Gathering and Analysis	• Interview with stakeholder • Collect the requirements from stakeholders. • Analysis and documented the requirements in SRS	• Software Requirements Specification (SRS)
	Iteration Start	
Design	• Identify the suitable software and hardware requirements. • Create the UML diagrams of the system. • Design the interface of each module. • Design the database structure. • Document the system design in SDD	• Class diagram • Use case diagram • Activity diagram • Sequence diagram • Software Design Document (SDD)
Implementation	• Develop the system module. • Develop database of the system • Develop the interface of the system	• Complete system module • System database • System interface
Testing	• Conduct testing • Generate test case. • Record and document the testing result. • Conduct debugging process. • Conduct retesting process	• Test case report • Test plan
Evaluation	• Review and evaluate the complete system by stakeholders. • Record changed or add requirement (if needed)	• User evaluation result
	Iterative End	
Deployment	• Deploy the system in real-world environment.	• Complete system

4. Result and Discussion

In this section, the data and the analysis of the study will be discussed and presented. Besides that, the system requirement analysis , system analysis and system design will also be discussed in this section.

4.1 System Requirement Analysis

One of the main difficulties in software development projects is requirement analysis because different stakeholders have different contexts in which to articulate and specify requirements [5]. However, requirement analysis needs to be carried out to ensure the developed system is developed by meeting the user requirements. Besides that, after carrying out requirement analysis, the process flow of the developed system can be understood in more detail. Functional requirements refer to the functions or features that should be implemented in the system to fulfill the users' needs. The design undergoes verification against specified requirements that outline the expected behavior and functionality of the software is called functional requirements [6]. Thus, the functional requirements should be discussed and identified before developing a new system. Next, non-functional requirements refer to quality requirements that should be implemented in the system to maintain the quality of the system. The examination of non-functional requirements and the establishment of acceptance parameters serve to effectively tackle the encountered quality challenges [7]. In addition, Table 3 displayed the functional requirement of the system whereas Table 4 displays the non-functional requirements of the system.

Table 3 *Functional requirements of the developed system*

No	Module	Description

1	Login and registration module	• The system should allow new users such as students and landlords to register by using e-mail. • The system should display error messages for any invalid input of registration detail. • The system should save the registration details into the database for future login purposes. • The system should redirect the user to the login page after registration. • The system should allow users to login using e-mail and password. • The system should only allow users to login with valid e-mail and password. • The system should display error messages for any invalid input of login detail. • The system should redirect users to main page if the e-mail and password are valid.
2	Account detail management module	• The system should allow users to view account details. • The system should allow users to edit account details such as usernames and passwords. • The system should update the latest account status to database.
3	House management module	• The system should allow landlords to create house detail. • The system should allow landlords to view house detail. • The system should allow landlords to edit house detail. • The system should allow landlords to delete house detail. • The system should allow administrators to view the overall house list.
4	Rent house module	• The system should allow students to view available house list. • The system should allow students to search the house. • The system should allow students to view appointments with the landlords. • The system should allow landlords to approve or reject the appointment. • The system should allow students and landlords do cancellation before the appointment date. • The system should be able to record the attendance. • The system should allow student to make rental request to landlord after attending the appointment • Thes system should allow landlords to approve or decline the rental requests
5	Payment and agreement management module	• The system should allow the students to select payment methods. • The system should allow the students to make payment after rental request is accepted by the landlords. • The system should display the agreement after payment is made. • The system should allow the students, the landlords and the administrator to view payment history. • The system should allow student and landlords to print the agreements.
6	Complaint management module	• The system should allow users to create complaints. • The system should allow users to view complaints. • The system should allow users to edit complaint. • The system should allow users to delete complaint.

Table 3 *(cont)*

7	Report generation module	• The system should allow the user to view payment reports. • The system should allow the user to view house reports. • The system should allow the user to view complaint reports. • The system should allow the user to print report

Table 4 Non-*Functional requirements of the developed system*

No	Module	Description
1	Operational	• The system should be able to function in any web browser such as Chrome, Edge, Firefox and so on
2	Security	• The system should only allow authenticated user to access the system.
3	Performance	• The response time of the system on the users' interaction should not more than 3 seconds.
4	Usability	• The interface of the system should be easy to use and simple to learn for the users without any technical knowledge.
5	Availability	• The system should always be readily operable and accessible.

4.1.1 Use Case Diagram

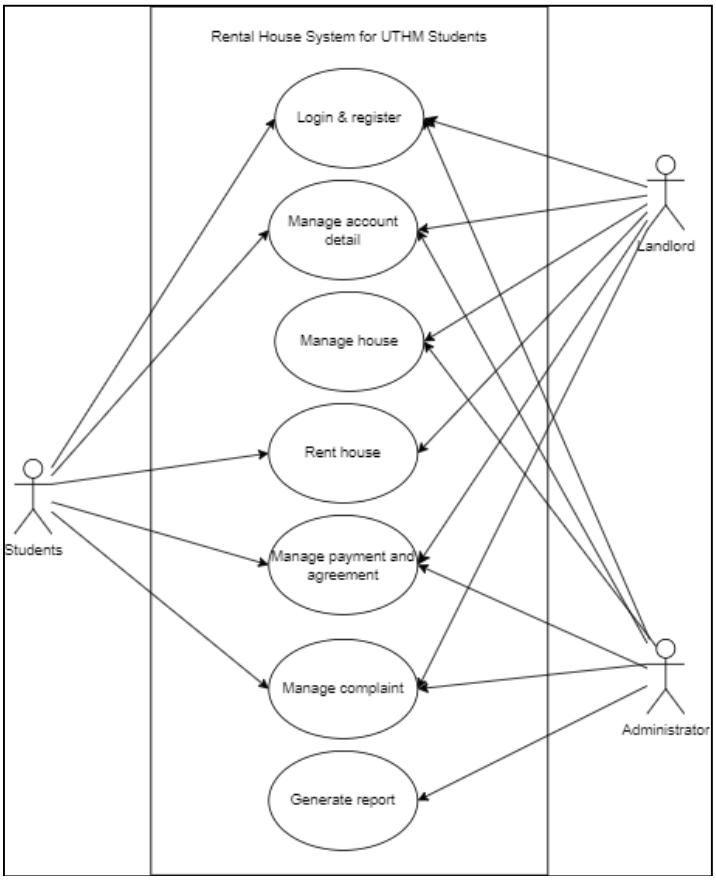


Fig. 2 *Use Case Diagram for Rental House System for UTHM Students*

Use case diagram is a graphic representation that display the system users, which also called as actors and their interactions with the system modules [8]. Besides that, the use case diagram also is in UML (Unified Modelling Language) as a modelling of the artifact and flow of the software systems. Figure 3 shows the use case diagram of Rental House System for UTHM Students. Based on Figure 2, there are three actors in this system which are Students, Landlords and Administrators. and seven modules which are login and register, manage account detail, manage house, rent house, manage payment and agreement, manage complaint and generate report.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1065-1083

4.1.2 Class Diagram

The creation of a class diagram helps facilitate the coding process for system development. The diagram includes the names of the classes, their attributes, and the relationships between the classes. It is important to note that the above class diagram is built on the principles of object-oriented programming and can be used in different phases such as analysis, design and implementation. Therefore, the class diagram will further elaborate on the nature of relationships involved in class names, attribute lists, function lists, and system development of the Rental House System for UTHM Students. Figure 3 shows the class diagram of the system.

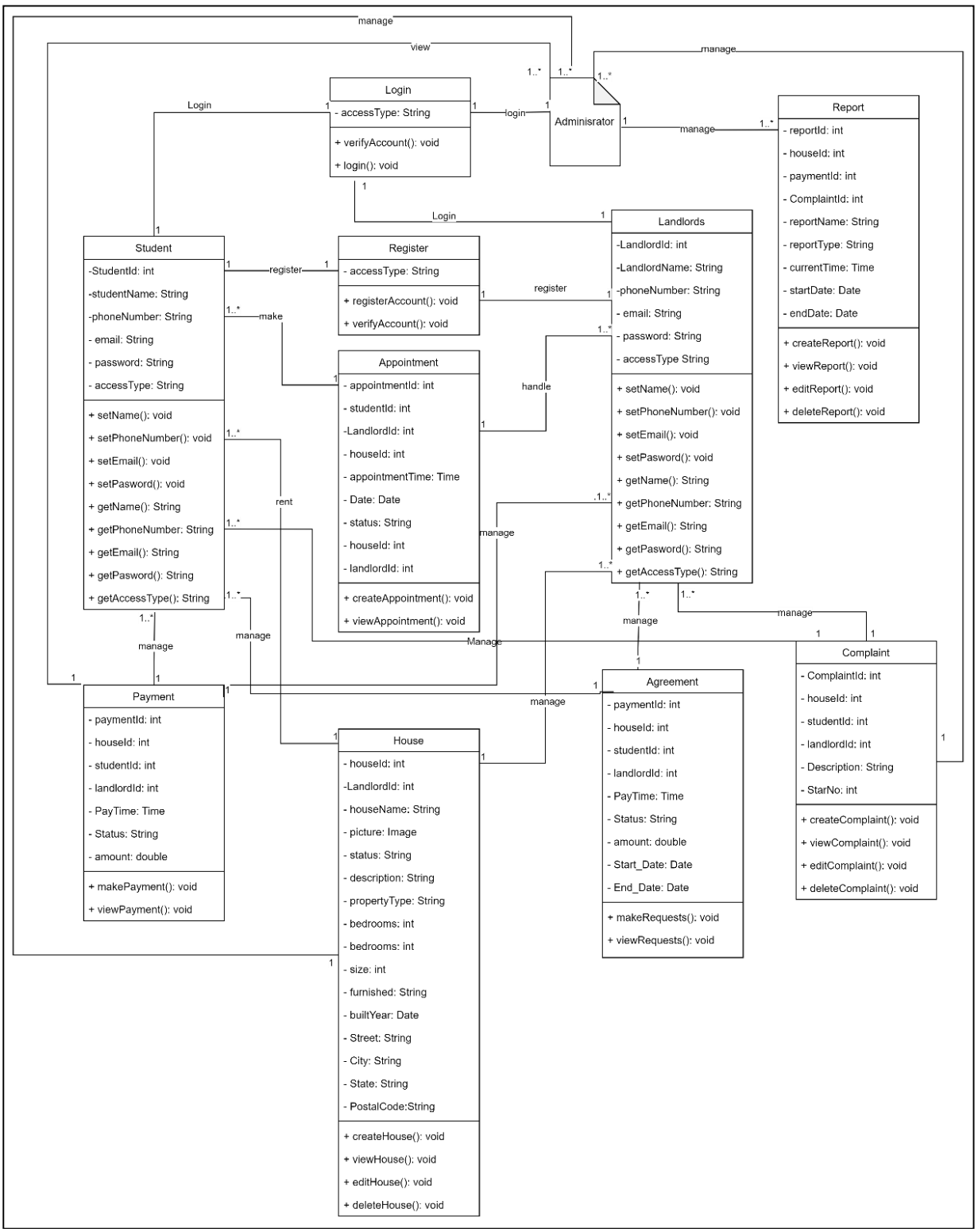


Fig. 3 *Class Diagram for Rental House System for UTHM Students*

4.2 Implementation

The modules of the system are implemented and developed as the system design. The tools use in development of this web-based system are Visual Studio Code as source-code editor and MySQL in XAMPP as the database management tool. The user interface of the system in each module will be explained in this section.

4.2.1 User Login and Registration Module

Figure 4 shows the interface of register account interface for landlords and students. The student and landlord can register an account by inputting their name, email, phone number, password and then choose the user type and upload the profile image. After that, they need to verify their email address by clicking the verifying link in the email.

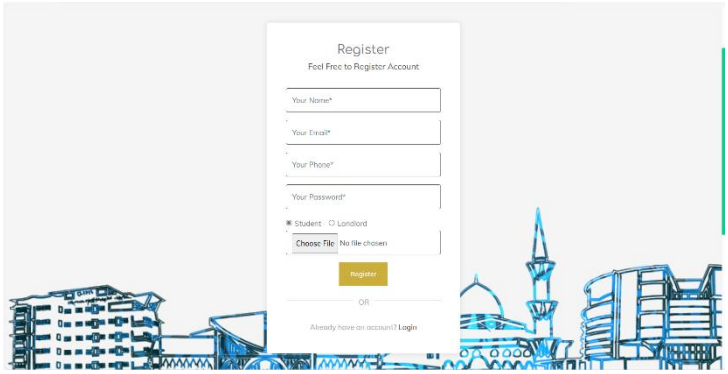
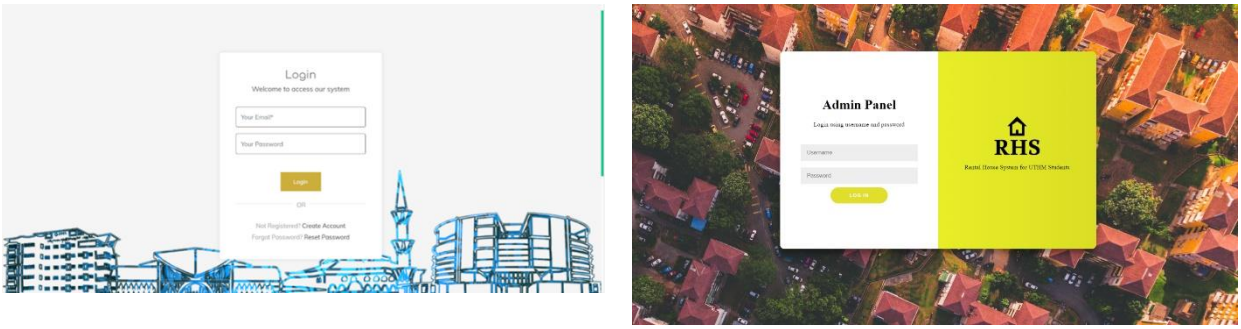


Fig. 4 *Register Interface for Students and Landlords*

Figure 5 display the login page for students, landlords and administrator. The student and landlords can login to the system by entering their email and password. If they forgot password, they could click the "Reset password" for getting a reset password email so they can click the link in the email to reset password. If the email or password is not matched, the user is not allowed to access the full functionality of the system.



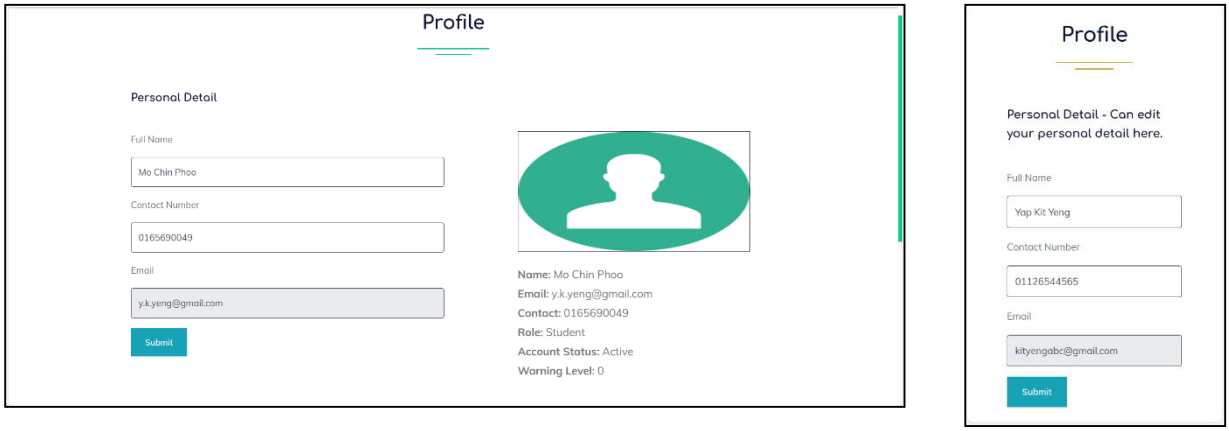
(a) **(b)**

Fig. 5 *Login Interface for (a) Students and Landlords; (b)Administrator*

4.2.2 Manage Users Details Modules

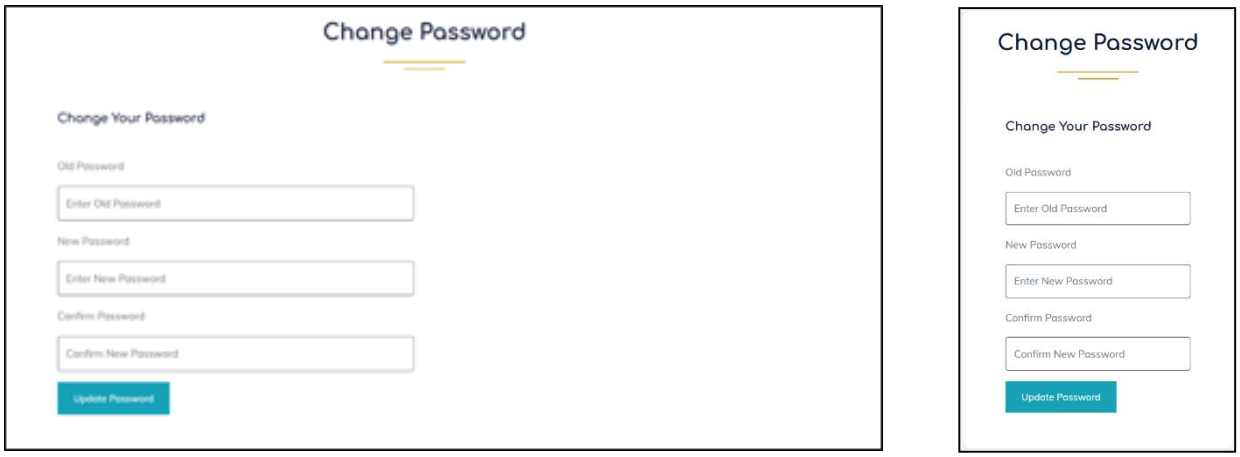
Figure 6 shows the interface of edit profile Interface of Students and Landlords in normal view and mobile view. The student and landlord can change their username and phone number in the filed then click submit button to update the detail.

Figure 7 display the login page for students, landlords and administrator. The student and landlords can login to the system by entering their email and password. If they forgot password, they could click the "Reset password" for getting a reset password email so they can click the link in the email to reset password. If the email or password is not matched, the user is not allowed to access the full functionality of the system.



(a) **(b)**

Fig. 6 *Edit Profile Interface of Students and Landlords in (a) Normal View (b) Mobile View*



(a) **(b)**

Fig. 7 *Change Password Interface of Students and Landlords in (a) Normal View (b) Mobile View*

Figure 8 shows the change password interface of administrator. The administrator can change the password by inputting the old password, new password and confirm password. Besides that, the administrator can view the personal details by clicking the personal details tab.

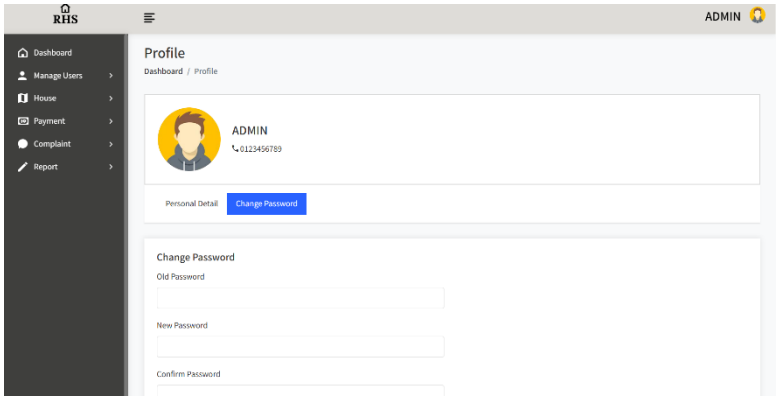
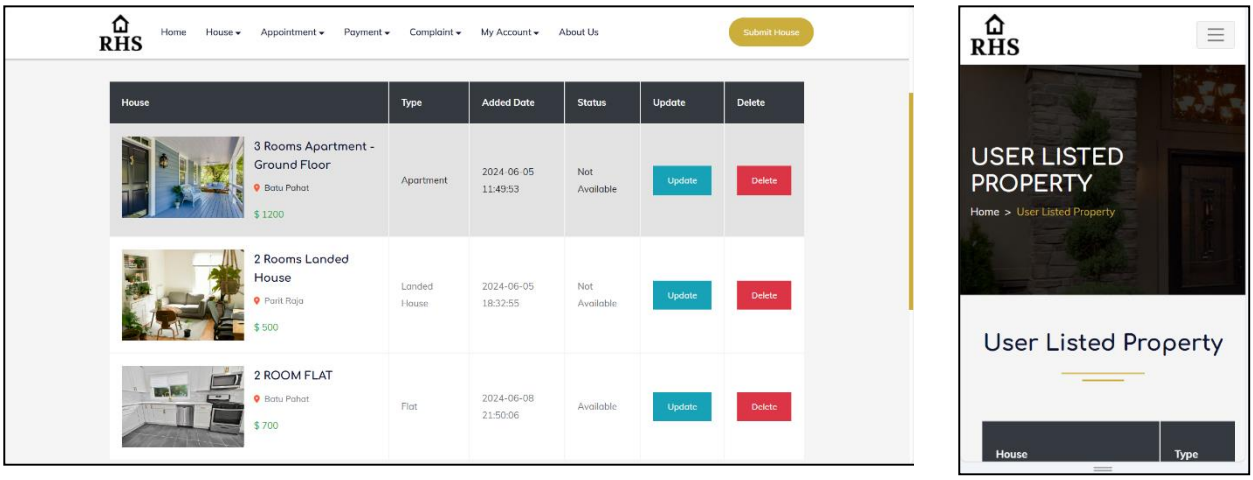


Fig. 8 *Change Password Interface of Administrator*

4.2.3 Manage House Modules

Figure 9 shows the manage house interface of the landlords in normal view and mobile view. The landlords can submit their house by clicking the "Submit House" button and insert the house detail. Then, landlords also can update house detail and delete the house by clicking the "Update" button and "Delete" button.



(a) **(b)**

Fig. 9 *Manage house Interface of Landlords in (a) Normal View (b) Mobile View*

Figure 10 shows the Manage House Interface of Administrator. the administrator can manage the house in this page, they can view the house detail and delete house.

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1065-1083

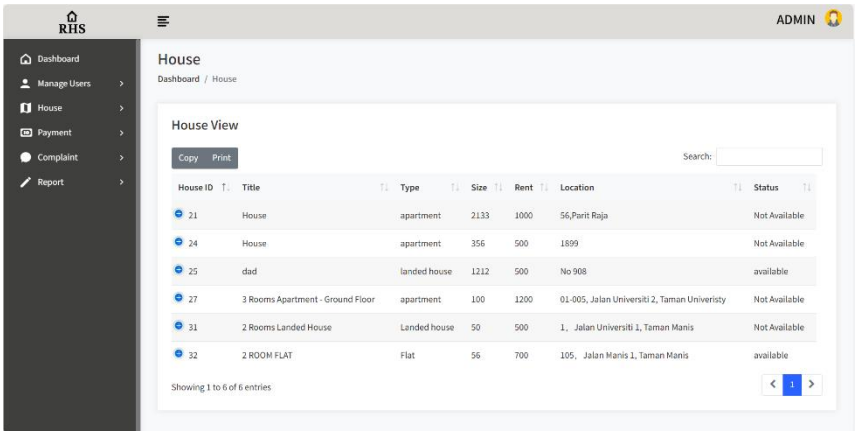
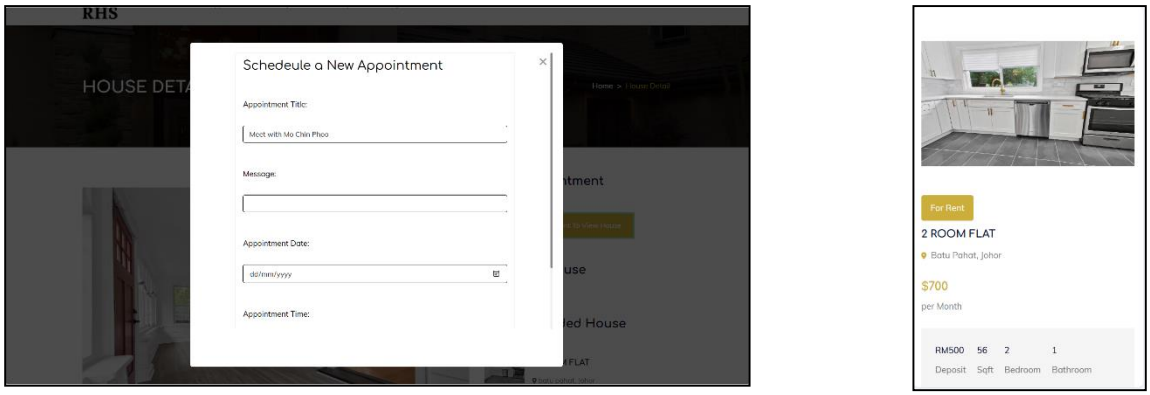


Fig. 10 *Manage House Interface of Administrator*

4.2.4 Rent House Modules

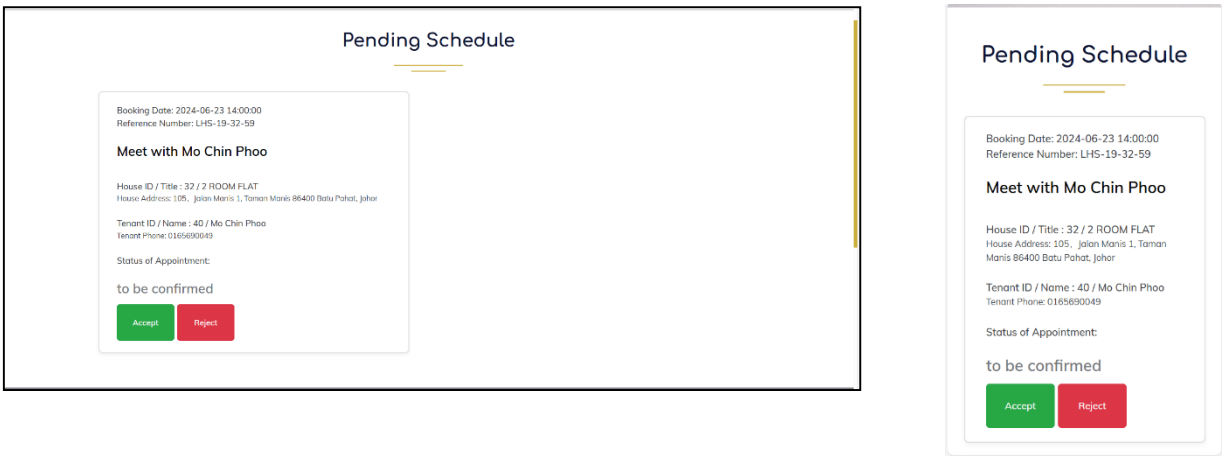
Figure 11 shows the rent house interface of the students in normal view and mobile view. The students can view the available list and the details of the available house. In the house detail page, the students can schedule appointment with the landlords to view the house by clicking the "schedule appointment" button and then filling in the appointment detail such as appointment title, message , date and time.



(a) **(b)**

Fig. 11 *Rent house Interface of Students in (a) Normal View (b) Mobile View*

Figure 12 shows the rent house interface of landlords in normal view and mobile view. In this page, the landlords can accept or reject the appointment. If the appointment is accepted, the landlord can generate the QR code for the students to scan for the attendance.



(a) **(b)**

Fig. 12 *Rent house Interface of Landlords in (a) Normal View (b) Mobile View*

4.2.5 Manage Payment and Agreement Module

Figure 13 shows the make payment page of the student, the student can key in their debit and credit card in the field to pay for the deposit. Beside that, they also can use Link to pay the deposit and first month rent to the landlord to generate rental agreement with landlord. After payment is successfully made, the payment history is recorded down in payment history page of students and landlords as in Figure 14. The student and the landlords can view the agreement by clicking the "agreement" button.

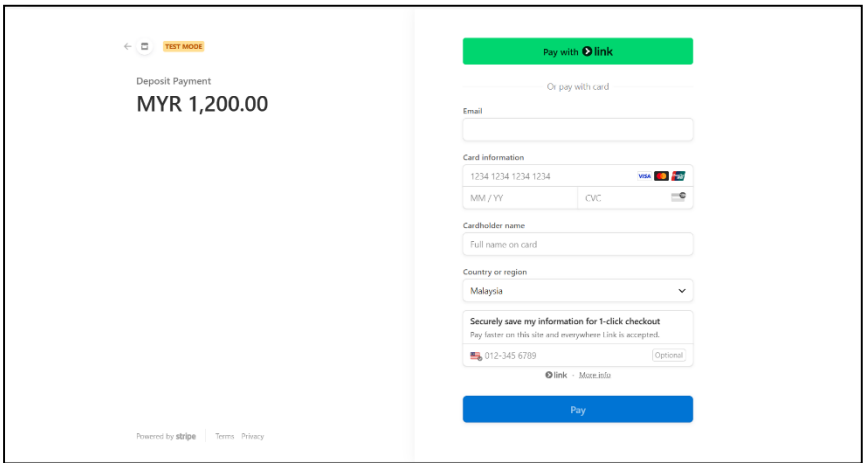


Fig. 13 *Payment interface of Students*

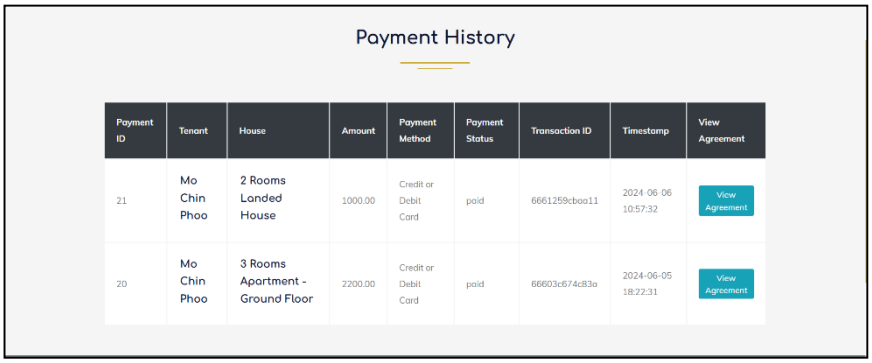


Fig. 14 *Payment History Interface of Students and Landlords*

Figure 15 shows the payment history page of adminsitrator, the adminsitrator can view the payment detail auch as payment ID, transaction ID in the table and the agreement detail between the landlord and student.

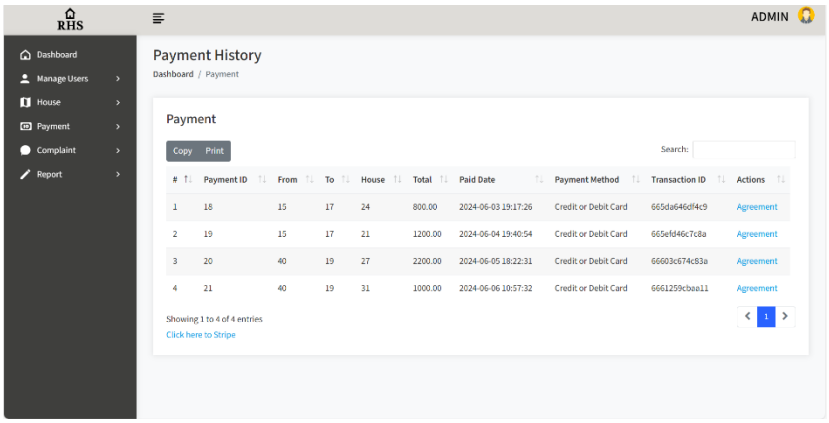
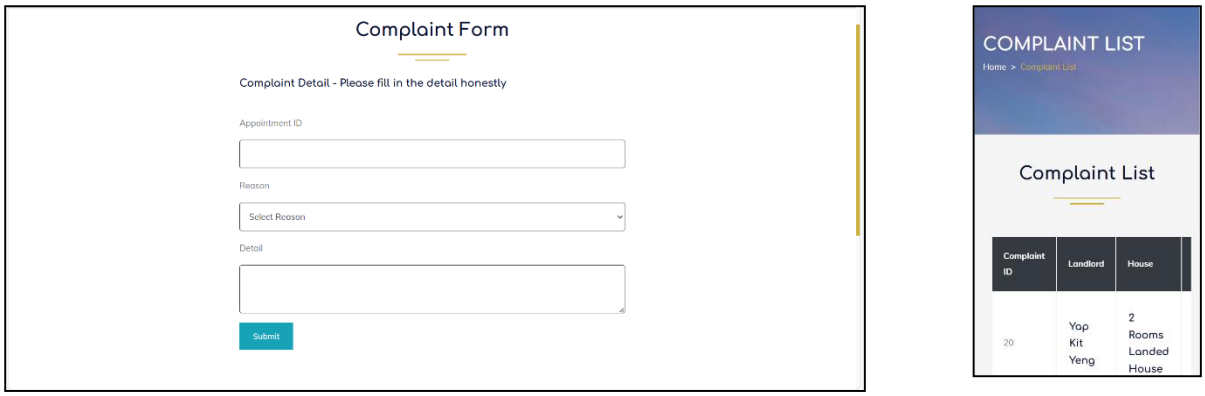


Fig. 15 *Payment History Interface of Administrator*

4.2.6 Manage Complaint Modules

Applied Information Technology and Computer Science Vol. 5 No. 2 (2024) p. 1065-1083

Figure 16 shows the manage complaint interface of students and landlords in normal view and mobile view. The students and the landlords can fill in the related appointment ID which is absent by one of the parties and fill in the reason and the detail of the complaint then click the "Submit" button to make complaint. After compliant is made, the complaint will be recorded in the complaint list to follow the further update of the complaint.



(a) **(b)**

Fig. 16 *Manage Complaint Interface of Students and Landlords in (a) Normal View (b) Mobile View*

Figure 17 shows the manage complaint module for the administrator. The administrator can view the complaint detail by clicking the detail button. Besides that, if the complaint is solved successfully, the administrator can choose to close the complaint. If the complaint is cancelled or closed, the administrator can delete the complaint history.

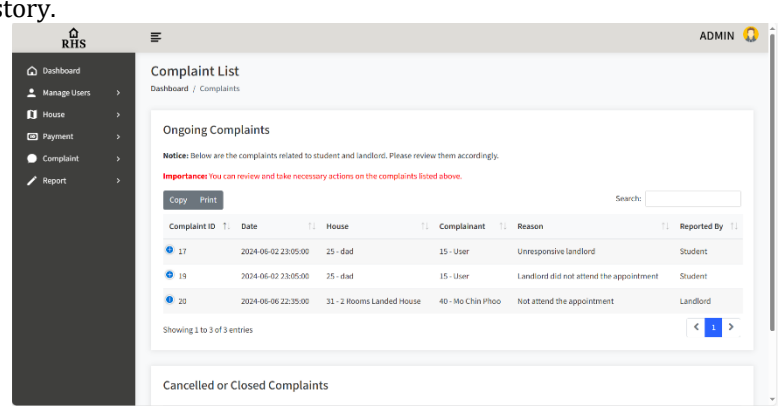


Fig. 17 *Manage Complaint Interface of Administrator*

4.2.7 Generate Report Modules

Figure 18 shows the generate report interface. There are bar chart indicating the number of the user based on the user type and pie chart that visualizing the number of houses based on the house type. Besides that, there are a line chart that indicating the payment history made by the student to pay for the deposit. The administrator can select the start date and end date of the report. After that, the administrator can print the report.

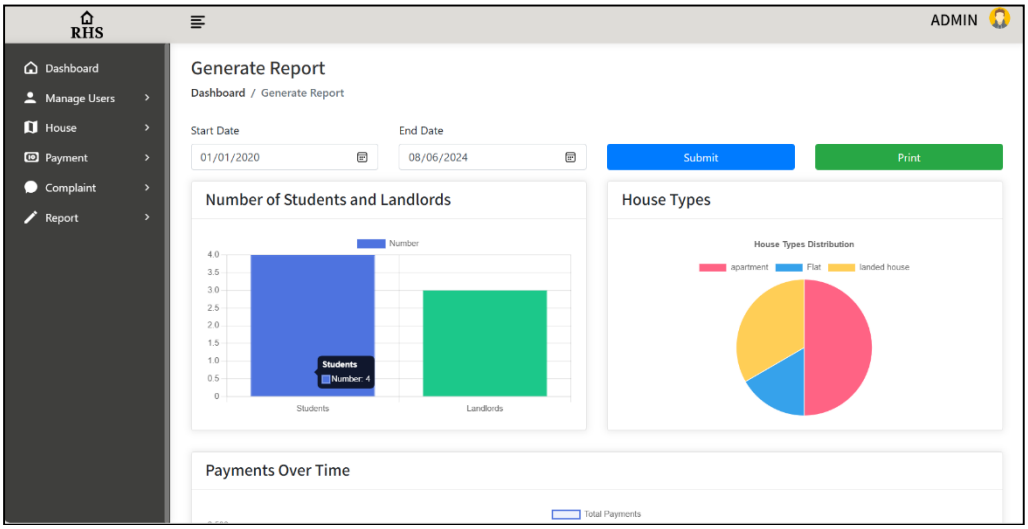


Fig. 18 *Generate Report Interface of Administrator*

4.3 Testing

Testing always be carried out after the development of the system. The goal of the testing is to discover if there are any errors or bugs in the system. If any bugs or errors is found, they will be fixed to ensure the system can function as expectation. In this section, the system testing, which is to test the system functionality, and user acceptance testing to collect feedback from the stakeholders will be discussed.

4.3.1 System Testing

The objective of carrying out system testing is to make sure that the developed system fulfills the functional requirements and non-functional requirements of the proposed plan. In the system testing, the whole system will be tested such as the integration of the system modules to find if there are any bugs or errors. Table 5 shows the test case and the result of the test case.

Table 5 *System Testing Results*

Test Case ID	Requirement ID	Description	Test Status
TC_100	**REQ_100**	**Register**	**-**
TC_100_01	REQ_101	The system shall display the registration page	PASS
TC_100_02	REQ_102	The system shall allow the user to enter registration details	PASS
TC_100_03	REQ_103	The system shall verify the password format	PASS
TC_100_04	REQ_104	The system shall display error message if the registration details are invalid	PASS
TC_100_05	REQ_105	The system shall save the registration details input by the users	PASS
TC_100_06	REQ_106	The system shall allow the user to enter login details	PASS
TC_100_07	REQ_107	The system shall display error message if the login details are invalid	PASS
TC_100_08	REQ_108	The system shall redirect to the main page based on the user types	PASS
TC_100_09	REQ_109	The system shall not allow registration requests that use the same email address	PASS
TC_100_10	REQ_110	The system shall be used on any web browser	PASS
TC_100_11	REQ_111	The response time of the system should be less than 3 seconds	PASS
TC_200	**REQ_200**	**Manage Account Detail**	-
TC_200_01	REQ_201	The system shall display manage account page.	PASS
TC_200_02	REQ_202	The system shall allow the user to edit the account details.	PASS

TC_200_03	REQ_203	The system shall display error message if the new details are invalid.	PASS
TC_200_04	REQ_204	The system shall verify the new password format.	PASS
TC_200_05	REQ_205	The system should deny the edition of the invalid new details.	PASS
TC_300	**REQ_300**	**Manage House**	**-**
TC_300_01	REQ_301	System should display manage house interface.	PASS
TC_300_02	REQ_302	System should allow users to view selected house.	PASS
TC_300_03	REQ_303	System should display house details if the house is found.	PASS
TC_300_04	REQ_304	System should display error message for house not found.	PASS
TC_300_05	REQ_305	System should display create new house interface.	PASS
TC_300_06	REQ_306	System should allow users to add new house.	PASS
TC_300_07	REQ_307	System should display error message for incomplete house details.	PASS
TC_300_08	REQ_308	System should display edit house detail interface.	PASS
TC_300_09	REQ_309	System should update existing house.	PASS
TC_300_10	REQ_310	System should delete existing house.	PASS
TC_400	**REQ_400**	**Rent House**	**-**
TC_400_01	REQ_401	System should display appointment list interface	PASS
TC_400_02	REQ_402	System should allow users to view selected appointment.	PASS
TC_400_03	REQ_403	System should display appointment details if the appointment is found.	PASS
TC_400_04	REQ_404	System should display error message for appointment not found.	PASS
TC_400_05	REQ_405	System should display create new appointment interface.	PASS
TC_400_06	REQ_406	System should allow users to add appointment and rental request.	PASS
TC_400_07	REQ_407	System should display error message for incomplete appointment details.	PASS
TC_400_08	REQ_408	System should display approve or reject appointment interface.	PASS
TC_400_09	REQ_409	System should update appointment status.	PASS
TC_400_10	REQ_410	System should allow users to cancel appointment.	PASS
TC_400_11	REQ_411	System should allow users to view selected appointment.	PASS
TC_400_12	REQ_412	System should allow users to approve or reject selected rental request.	PASS
TC_500	**REQ_500**	**Manage Payment and Agreement**	**-**
TC_500_01	REQ_501	System should display manage payment interface	PASS
TC_500_02	REQ_502	System should allow users to view selected payment.	PASS
TC_500_03	REQ_503	System should display payment details if the payment is found.	PASS
TC_500_04	REQ_504	System should display error message for unsuccessful payment.	PASS
TC_500_05	REQ_505	System should display make new payment interface.	PASS
TC_500_06	REQ_506	System should allow users to choose payment type.	PASS
TC_500_07	REQ_507	System should display manage rental request interface.	PASS
TC_500_08	REQ_508	System should allow users to make rental requests.	PASS
TC_600	**REQ_600**	**Manage Complaint**	**-**
TC_600_01	REQ_601	System should display manage complaint interface	PASS

TC_600_02	REQ_602	System should allow users to view selected complaint.	PASS
TC_600_03	REQ_603	System should display complaint details if the complaint is found.	PASS
TC_600_04	REQ_604	System should display error message for complaint not found.	PASS
TC_600_05	REQ_605	System should display create new complaint interface.	PASS
TC_600_06	REQ_606	System should allow users to add new complaint.	PASS
TC_600_07	REQ_607	System should display error message for incomplete complaint details.	PASS
TC_600_08	REQ_608	System should display edit complaint detail interface.	PASS
TC_600_09	REQ_609	System should update existing complaint.	PASS
TC_600_10	REQ_610	System should delete existing complaint.	PASS
TC_700	**REQ_700**	**Generate Report**	**-**
TC_700_01	REQ_701	The system shall display the report details.	PASS
TC_700_02	REQ_702	The system shall allow the user to print the report.	PASS
TC_700_03	REQ_703	The system shall display the report in less than 3 seconds.	PASS

4.3.2 Overall Test Result

In overall, there are a total of 59 test cases carried out for the system testing. As the result, the system functionality fulfills the functional requirements and non-functional requirements since the tesing result is all successful.

Table 6 *Overall Test Result*

Test Case	Total Test Cases	Total Success	Total Failed
TC_100	11	11	0
TC_200	5	5	0
TC_300	10	10	0
TC_400	12	12	0
TC_500	8	8	0
TC_600	10	10	0
TC_700	3	3	0
Total	59	59	0

Figure 19 show the pie chart the indicate the overall test result. The result is all the test case is passed. This means that the system has met the requirements as expected in the planning phase.

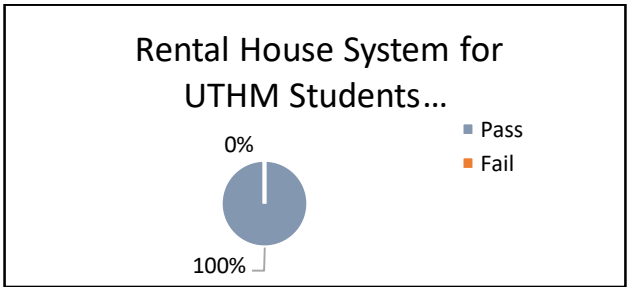


Fig 18 *Pie Chart of Overall System Testing Result*

4.3.3 User Acceptance Testing

User Acceptance Testing (UAT) is a testing that is usually used in the software development lifecycle for the end users or stakeholders to ensure it meets their requirements and functions as expected. It serves as the final validation before the software is deployed into production. For collecting responses from stakeholders, the user acceptance form is created and given to the tester who is the stakeholder of the system to evaluate the system. The evaluate form is attached to the Appendix section. In the evaluation form, there are several modules that should be tested by the testers which are login and registration module, manage account detail module, rent house module, manage payment and agreement module and manage complaint module. The tester can tick pass if the module meets their expectation whereas they can tick failed if they feel unsatisfied of the module. If there are any feedback from user, thy can fill in the note field.

The result of the evaluation form is the tester ticked the pass as the result of all modules. This indicates the developed system meet the stakeholder expectation. It is also means that the developed system fulfil the functionality and non-functionality requirements as planned.

5. Conclusion

In conclusion, the Rental House System for UTHM Students presents a unique and user-friendly solution to the challenges faced by students in finding suitable rental housing near their campus. This web-based platform offers several significances such as streamlined rental process and enhanced transparency and trust. This system simplifies the process of housing search for students by providing verified listings, allowing them to choose based on preferences and budget, and eliminating the need for time-consuming physical searches. For further enhancement of the system, mobile application development is one of the recommendations for future work. A mobile app can be created to increase accessibility and convenience for users, allowing them to manage their rental needs on the go. Besides that, the search and filter function can be more advanced. The availability of more granular search and filtering options, such as based on proximity to specific amenities or landmarks, will cater to the varying needs and preferences of students, ensuring they find the perfect rental match.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia 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:** Yap Kit Yeng, Mohd Hamdi Irwan Bin Hamzah; **data collection:** Mohd Hamdi Irwan Bin Hamzah; **analysis and interpretation of results:** Yap Kit Yeng, Mohd Hamdi Irwan Bin Hamzah; **draft manuscript preparation:** Mohd Hamdi Irwan Bin Hamzah. All authors reviewed the results and approved the final version of the manuscript.

Appendix : User Acceptance Form

User Acceptance Testing (UAT)

Rental House System for UTHM Students

Name : YEW WAN QIAN

Phone Number : 018-3906812

Role : Student

Acceptance Criteria	Test Results		Note
	Pass	Fail	
Login and Register			
Display Login Page	/		
Able to login by entering the valid email and password	/		
Display Register Page	/		
Able to register a new account by entering the valid email	/		
Manage Account Detail			
Display edits profile page	/		
Able to edit the account details.	/		
Able to change password	/		
Rent House			
Able to schedule appointment	/		
Able to view the appointment list	/		
Able to make the rental request	/		
Able to search house	/		
Able to view house detail.	/		
Manage Payment and Agreement			
Able to make payment	/		
Able to view agreement	/		
Manage Complaint	/		
Able to make complaint	/		
Able to cancel complaint	/		
Able to reply to complaint	/		
Able to view complaint list	/		

I , Yew Wan Qian ..... hereby declare that the information provided is true and correct.

Tested by,

(YEW WAN QIAN)

Date: 9/6/2024

References

[1] R. Rastogi, "Rental House Management System: An Empirical Approach with Simulation," *Actascientific*, Feb. 20, 2023. https://actascientific.com/ASCS/pdf/ASCS-05-0434.pdf

[2] M. R. Misyam, "House Rental Management System," *Penerbit UTHM*, Sep. 16, 2021. https://publisher.uthm.edu.my/periodicals/index.php/aitcs/article/view/2488/1384 (accessed Oct. 13, 2023).

[3] J. Yao and Y. Yao, "Web-based support systems," *ResearchGate*, Oct. 29, 2003. https://www.researchgate.net/publication/2887734_Web-based_Support_Systems (accessed Nov. 17, 2023).

[4] R. V. Labaree, "Research guides: Organizing Your Social Sciences Research Paper: 6. the methodology," *Research Guides at University of Southern California*, 2023. https://libguides.usc.edu/writingguide/methodology (accessed Dec. 02, 2023).

[5] S. T. Demirel and R. Das, "Software requirement analysis: Research challenges and technical approaches," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, Mar. 2018. Accessed: Dec. 23, 2023. [Online]. Available: http://dx.doi.org/10.1109/isdfs.2018.8355322

[6] A. Ray, C. Ackermann, R. Cleaveland, C. Shelton, and C. Martin, "Functional and Nonfunctional Design Verification for Embedded Software Systems," in *Advances in Computers*, Elsevier, 2011, pp. 277–321. Accessed: Dec. 26, 2023. [Online]. Available: http://dx.doi.org/10.1016/b978-0-12-385510-7.00006-0

[7] S. Saroja and S. Haseena, "Functional and Non-Functional Requirements in Agile Software Development," *Agile Software Development*, pp. 71–86, Feb. 2023, doi: 10.1002/9781119896838.ch5.

[8] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, "The use case diagram," in *UML @ Classroom*, Cham: Springer International Publishing, 2015, pp. 23–47. Accessed: Dec. 25, 2023. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-12742-2_3