HOUSE RENTAL SYSTEM FOR UMP STUDENT.

MUHAMMAD ASLAM BIN MAT ASRI

Bachelor of Computer Science (Software Engineering) with Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name	:	MUHAMMAD ASLAM BIN MAT ASRI
Date of Birth	:	
Title	:	HOUSE RENTAL SYSTEM FOR STUDENT
Academic Session	:	2022/2023

I declare that this thesis is classified as:

□	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
□	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)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

_______________________	_______________________
(Student's Signature)	(Supervisor's Signature)
___	Azma Abdullah
New IC/Passport Number	Name of Supervisor
Date:	Date: 28 July 2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

UMP
Universiti
Malaysia
PAHANG
Engineering • Technology • Creativity

SUPERVISOR’S DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of Bachelor of Computer Science (Software Engineering) with Honours

____

(Supervisor’s Signature)

Full Name : Ts, Azma Binti Abdullah

Position : Lecturer

Date : 28 July 2023

STUDENT’S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

(Student’s Signature)

Full Name : MUHAMMAD ASLAM BIN MAT ASRI

ID Number : CB20156

Date :

HOUSE RENTAL SYSTEM FOR UMP STUDENT (UNIHOMES)

MUHAMMAD ASLAM BIN MAT ASRI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

July 2023

ACKNOWLEDGEMENTS

In the accomplishment of this project successfully, many people have been backbone for me to complete this project. I would like to take opportunity to thanks all the people that is directly or indirectly that contributed to the project

Primarily I would like to thank to Allah for being able me and allow me to finish this project. Then, I also would like to thanks my supervisor, Ts Azma Binti Abdullah for guiding me during this project. Her comments and contribution play a big part in completing this project.

Besides that, I also would like to express my gratitude to my parents that have been supporting me. Not to mention, my friend that encourage me during this project.

ABSTRAK

UniHomes ialah penyelesaian berasaskan web yang telah direka bentuk untuk menyediakan pusat sehenti untuk pelajar mencari rumah atau bilik untuk disewa. Masalah semasa yang dihadapi oleh pelajar ketika mereka mencari penginapan memakan masa kerana mereka perlu pergi ke platform yang berbeza, kekurangan ketelusan yang boleh membawa kepada penipuan dan akhir sekali terdapat beberapa tuan rumah yang sanggup menyewakan harta mereka kepada pelajar. Untuk menangani cabaran ini, UniHomes dibangunkan sebagai sistem berasaskan web yang menggunakan teknologi semasa dengan antara muka mesra pengguna. Pendekatan ini memudahkan akses mudah untuk tuan tanah, pelajar dan PETAKOM mudah untuk menggunakan sistem walaupun mereka mempunyai pengetahuan teknikal yang terhad. Memandangkan sistem dibangunkan sebagai berasaskan web, ia menjadikannya lebih mudah untuk diakses di mana-mana tempat pada bila-bila masa menggunakan mana-mana peranti yang menyokong pelayar. UniHomes telah diuji dengan sekumpulan pelajar dan tuan tanah dan telah menerima maklum balas positif dengan keputusan Ujian Penerimaan Pengguna kerana 10 daripada 10 penguji berpuas hati. Sistem ini telah meningkatkan kecekapan dan ketelusan pengurusan harta tanah. lebih-lebih lagi, UniHomes telah direka bentuk untuk memenuhi keperluan khusus pengurus hartanah dan tuan tanah, dan hasilnya, ia telah diterima baik oleh mereka yang telah mengujinya. Kesimpulannya, UniHomes ialah alat yang berharga untuk pelajar dan tuan tanah untuk menyelaraskan proses pengurusan harta mereka dan meningkatkan kecekapan dan ketelusan operasi mereka. Sistem h berpotensi untuk merevolusikan cara pengurusan hartanah sewaan. UniHomes dilihat menjadi penyelesaian terbaik untuk pelajar dan tuan rumah yang ingin menambah baik cara mereka menguruskan hartanah sewa mereka.

ABSTRACT

UniHomes is a web-based solution that has been designed to provide an one stop center for the student to search for house or rooms to rent. The current problem that is faced by the student when they are searching for accommodations are time consuming as they need to go to different platforms, lack of transparency that can lead to scam and lastly there few landlords that are willing to rent their property to the students. To tackle these challenges, UniHomes is develop as a web-based system that utilize the current technology with a user friendly interface. This approach facilitates easy access for the landlords, students and PETAKOM easy to use the system even they have limited technical knowledge. As the system is develop as web based it make it easier to be access at any place at any time using any devices that support browsers. The UniHomes has been tested with a group of students and landlords and has received positive feedback with the User Acceptance Test result in as 10 out of 10 testers are satisfied. The system has improved the efficiency and transparency of property management. moreover, UniHomes has been designed to meet the specific needs of property managers and landlords, and as a result, it has been well received by those who have tested it. In conclusion, UniHomes is a valuable tool for students and landlords to streamline their property management process and improve the efficiency and transparency of their operations. The system h has the potential to revolutionize the way rental properties are managed. UniHomes is seen to become the go-to solution for students and landlords who want to improve the way they manage their rental properties.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

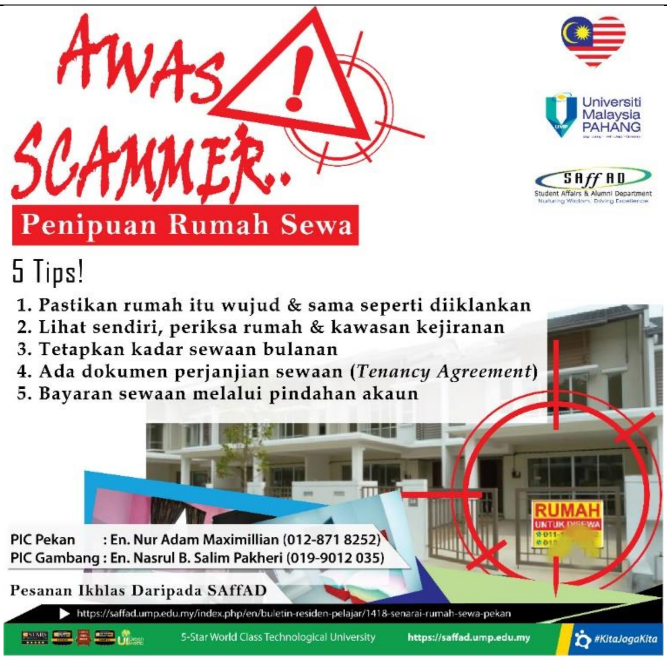
One of the basics need of a human being is shelter. Everyone needs to have a place where they can rest or head to whenever they want to take a break. For university students it is a common practice where the first year student are provided with hostel by the university. However, some university can't afford to provide the students hostel inside or outside of the campus for their entire study because of some limitation and challenge such as room capacity. Therefore, the students that is not allegeable to apply for hostel need to find their own house or room to be rent for the to be able to attend to classes or activity inside the campus.

Searching for a house or room for rent can be quite a hassle for students because they need to consider a lot of things such as budget, distance for campus, transports, accommodation, safety and location of the place itself. In Addition to the post COVID-19 pandemic the number of students that will be going back to campus is higher than before. So, the demand for the house or room increases. The amount of effort and time need to be spent on house hunting is tiring as the student need to go to different housing area to find a house or room that is suitable for them to rent. Some of them also use the web to search for house in Facebook, Mudah.my, PropertyGuru and other platform.

Moreover, most of the landlord usually are reluctant to make student as their tenant because of their past experience or others telling stories that student does not take care of the house such as not paying the monthly utilities bills, damaging the house and not keeping the house clean (*Landlord Reveals Worst Student Housing Problems, And Why It's A Never-Ending Nightmare | TRP*, n.d.). So, the project is be developed to counter this issue by becoming the one stop centre for the students to search for house and room to rent handled by the PETAKOM of Faculty of Computing as an Admin that will verify the application or the houses and rooms that will be advertised on the website.

1.2 Problem Statement

In this era of globalization, people usually use their smartphone or computer to shop, find articles, read news, watch entertainment and any other task. In this study aspect, the student utilizes the technology to find their shelter using the social media such as Facebook, Mudah.my and PropertyGuru website. However, most of the advertised house or room are not in the range of the student budget because most advertisements target family, working adults or anyone that can pay the respectful rent and can sign the lease for the longer period of time.


AWAS SCAMMER..
Universiti Malaysia PAHANG
SAffAD
Student Affairs & Alumni Department
Nurturing Wisdom, Driving Excellence
RUMAH
UNTUK DISEWA
5-Star World Class Technological University
#KitaJagaKita

Penipuan Rumah Sewa

5 Tips!

1. Pastikan rumah itu wujud & sama seperti diiklankan
2. Lihat sendiri, periksa rumah & kawasan kejiranan
3. Tetapkan kadar sewaan bulanan
4. Ada dokumen perjanjian sewaan (*Tenancy Agreement*)
5. Bayaran sewaan melalui pindahan akaun

PIC Pekan : En. Nur Adam Maximillian (012-871 8252)
PIC Gambang : En. Nasrul B. Salim Pakheri (019-9012 035)

Pesanan Ikhlas Daripada SAffAD

https://saffad.ump.edu.my/index.php/en/buletin-residen-pelajar/1418-senarai-rumah-sewa-pekan

https://saffad.ump.edu.my

Figure 1.1 UMP announcement

First problem statement is the student can be exposed to scam. As been reported by the Berita Harian, the case of scamming student for house to be rent increase since the student are allowed by the government to be back to campus for face to face class. The victims losses about RM26,700 because of the scam (*"Scam" Rumah Sewa,* *Penuntut IPT Jadi Sasaran*, n.d.) . This also quiet a concern for the student when searching for a house to be rented as they need to be careful such as meet with the owner face to face and make sure that the house does exist. The figure 1.1 shown the announcement from the Student Affairs & Alumni Department (SAffAD) of Universiti Malaysia Pahang to aware the student about the scam.

Second problem is the student also needs to search for advertised house or room in various platform such as social media. There is no dedicated platform that the students can go to find accommodations. Thus, this project aims to solve this problem that have been faced by the students to ease their efforts to find a place to stay by providing trusted platform for them to search for it.

Lastly, most of the landlords are reluctant to rent their house or rooms the students because they have bad experience with the students before this(*Seven out of 10 Landlords Won't Rent out Their Property to Students | The Independent | The Independent*, n.d.). Therefore, the students are having difficulties in finding landlord that trust the student and willing to rent their property to them.

1.3 Objective

i. To study the requirements for house rental systems.
ii. To develop a prototype system in student house rent management system.
iii. To validate the proposed prototype system in web based.

1.4 Scope

a. Student

UMP student is the target audience for using the web based system.

b. Landlord

The landlord can advertise their houses or rooms to the student using the web based system.

c. Admin PETAKOM

The admin will be the one who verified houses and rooms that is advertised on the website.

i. Functions

Table 1.1 Modules

Module	Description
Student	1. The student can search for house and room based on location they prefer. 2. The student can view the detail about the property such as: ▪ Images of the interior and exterior of the property. ▪ Description of the house such as room number, fully furnished or not and any related attributes. ▪ Find contact number of the landlord. 3. The student can view the availability of the property. 4. The student can write a review about the property.

PETAKOM	1. The PETAKOM can manage the advertisement of the rooms and houses that is advertised in the system.
Landlord	1. The landlord can apply to advertise their houses and room that available for rent.

ii. Development Platform

The system is developed for web based as the web based application is more flexible and can be use on any computer through the internet.

1.5 Thesis Organization

The report will contain three chapter which are Introduction, Literature Review, and Methodology.

The first chapter contain the discussion about the project by describing the current issues and problem. It consists of problem statement, objectives, and scope.

Chapter 2 is about Literature Review of existing system which are mudah.my, PropertyGuru and RumahSiswa. This chapter contain the comparison about each advantage and disadvantages about the existing system.

Chapter 3 is about the methodology that is used to develop the project. The chapter contain the methodology, project requirement, propose design, proof of initial concept and testing.

Chapter 4 discuss about the implementation that have been done during the development of the project. this chapter also contain the testing that have been done

Chapter 5 conclude the entire project and also discuss about the constraint and limitation regarding the project and future work for the system.

CHAPTER 2

LITERATURE REVIEW

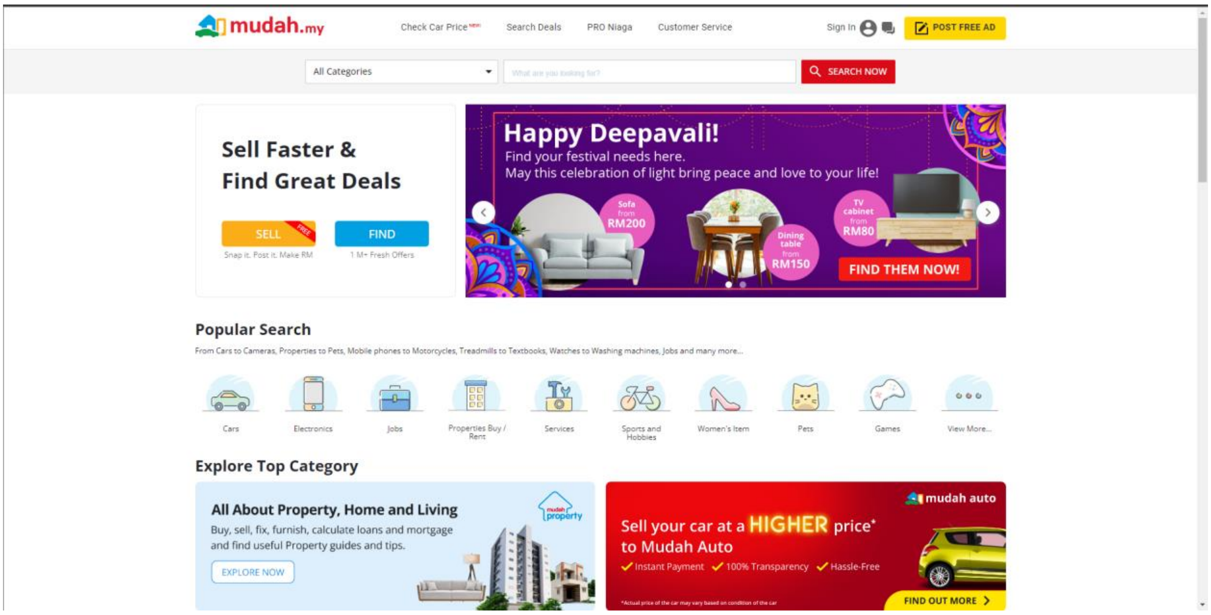
2.1 Introduction

This chapter discuss about the literature overview that can help to understand more about the requirements of the system by comparing the existing system that is used commonly by the students to search for house or room to rent. Mudah.my, PropertyGuru and RumahSiswa websites will be compared to each other to make further improvement for the system that will be develop.

2.2 Existing System

To conduct a comprehensive analysis, we have evaluated three existing systems, namely Mudah.my, PropertyGuru, and RumahSiswa. By comparing these platforms, we can identify their strengths and weaknesses, paving the way for improvement in our UniHomes system.

2.2.1 Mudah.my



Sell Faster & Find Great Deals

Snap it. Post it. Make RM

1 M+ Fresh Offers

Happy Deepavali!

Find your festival needs here.

May this celebration of light bring peace and love to your life!

Popular Search

From Cars to Cameras, Properties to Pets, Mobile phones to Motorcycles, Treadmills to Textbooks, Watches to Washing machines, Jobs and many more...

Explore Top Category

All About Property, Home and Living

Buy, sell, fix, furnish, calculate loans and mortgage and find useful Property guides and tips.

Sell your car at a HIGHER price* to Mudah Auto

✓ Instant Payment ✓ 100% Transparency ✓ Hassle-Free

*Actual price of the car may vary based on condition of the car

Figure 2.1 mudah.my

Figure 2.1 show the main interface of Mudah.my, which is an advertisement website that is mainly market in Malaysia. It is the result of a partnership between Singapore's 701Search (a joint venture of SPH and Schibsted) and Norway's Telenor ASA. Mudah.my is an online platform where the user can sell and find variety of products, services and other stuff such as real estate automotive, and career all across Malaysia. This service allows the user to sell and buy product without needing a physical shop(*Malaysia's Largest Marketplace - Buy & Sell Your New and Preloved Items - Mudah.My*, n.d.).

2.2.1.1 Features

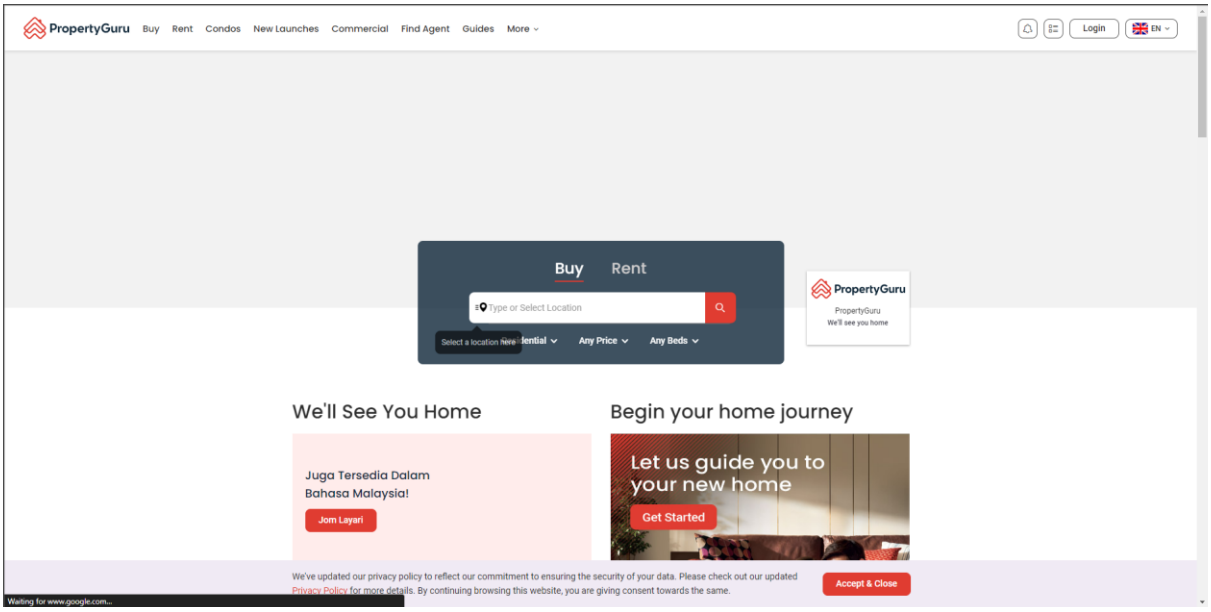
Mudah.my provides various function to the user such as they can search the item that they want based on categories such as electronics, property and many other categories. Mudah.my divided the user into two user which are seller and buyer. The user can choose either they want to search for product or sell them. The website allows the user to filter based on their location. They can filter out the result to only show products that is near them or location that they can go such as nearest district.

The user also can find result based on their budget. Next, the user can see the images uploaded by the user who sell a product or services and read the description of the posting. The user can contact the owner of the posting within the app or contact them via their contact number displayed in the posting. The user interface of the website is easy to understand even for someone who is first time using it.

2.2.1.2 Advantages and Disadvantages

The first advantage of the mudah.my website is it is available on web based platform and mobile platform that make it easy to be access anywhere at any time through the internet. Next advantage is the user interface is user friendly and easy to navigate However, even though the website offers a lot of variety of products, the reviews online about the website mostly talk about the number of scammers on the website. the user has been reporting numbers of scam on the website have been increasing.

2.2.2 PropertyGuru



We've updated our privacy policy to reflect our commitment to ensuring the security of your data. Please check out our updated Privacy Policy for more details. By continuing browsing this website, you are giving consent towards the same.

Figure 2.2 PropertyGuru

Figure 2.2 shows the interface of PropertyGuru, it was established in 2007, it is the first southeast Asia digital property marketplace that is leading also in Singapore, Malaysia, Vietnam and Thailand. They currently host more than 3.5 million real estate listings monthly, serve 40 million property seekers monthly with more than 64,000 active real estate agents across five growing economies of Southeast Asia – Singapore, Malaysia, Vietnam, Thailand, and Indonesia(*Asia's Largest Online Property Portal Group | PropertyGuru Group*, n.d.).

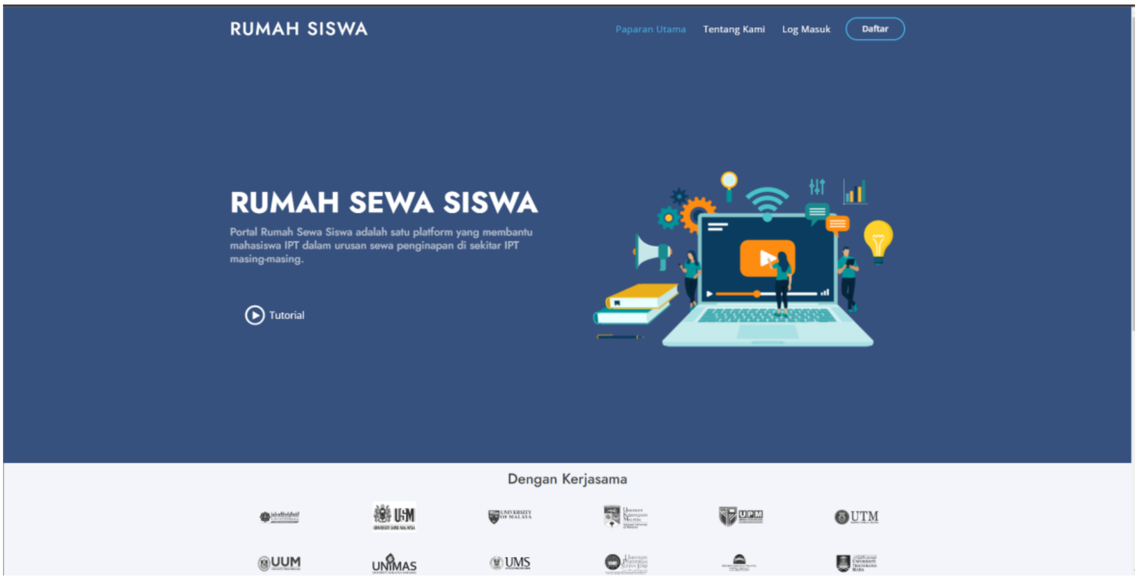
2.2.2.1 Features

PropertyGuru offer the user range of property to buy or rent. The user can filter the result based on the residential types, prices and number of beds. The user can view the house or room description and images in the posting. This website offers price insight that enable the user to view the market price of the house over the years. PropertyGuru provided the user with article and guide that is related to the property management such as tax management, agreement templates and guides, home improvement and home financing. The user interface of the website is very user friendly and consistent with its competitor mudah.my

2.2.2.2 Advantage and Disadvantage

The advantage of the PropertyGuru website is the user interface is user friendly and easy to use as the interface does not really different with other website that make the user struggle to use the website.

2.2.3 RumahSiswa



RUMAH SEWA SISWA

Portal Rumah Sewa Siswa adalah satu platform yang membantu mahasiswa IPT dalam urusan sewa penginapan di sekitar IPT masing-masing.

Figure 2.3 Rumah Siswa

Figure 2.3 shows the main page of Rumah Siswa, a portal that have been developed by the Ministry of High Education to help the student in accommodations by avoiding them involved in scam and make their process for searching a house is more organized and efficient(*Rumah Siswa*, n.d.).

2.2.3.1 Features

The function that is available on the website is the search function where the student enters the location that they preferred to rent, and the website will show the list of available houses and room to rent by the student.

2.2.3.2 Advantage and Disadvantage

The advantage of the Rumah Siswa website is the website target audience are the student that is looking for accommodations. The website only shows the result based on the institution that the student registered with. Moreover, the user cannot see anything other than just list of houses and room with no pictures that show the interior and exterior of the house.

2.3 Comparison of Existing System

Table 2.1 Comparison Table

	Mudah.my	PropertyGuru	Rumah Siswa
Platform	Mobile and web based	Mobile and web based	Web based
Features	1. Can filter the search result 2. Contain images of the property in the posting 3. Can contact the owner within the system	1. Can filter the search result 2. Contain images of the property in the posting 3. Can contact owner of within the system	1. Can filter the search result 2. Does not contain images of the property in the posting 3. Cannot contact the owner within the system

	Mudah.my	PropertyGuru	Rumah Siswa
	4. User friendly user interface	4. User friendly user interface	4. User friendly user interface
Advantage	1. Provides both mobile and web version. 2. User friendly interfaces. 3. Able to contact owner of the property within the system.	1. Provides both mobile and web version. 2. User friendly interfaces.	1. Developed as a one stop centre for high education students to find accommodations.
Disadvantage	1. A lot of users reported high number of scammers.	1. Focus more on property management	1. Not well maintained.

2.4 Relevance Comparison with Project

Table 2.2 Comparison with Project Table

Characteristic	Mudah.my	PropertyGuru	Rumah Siswa	UniHomes
Web	√	√	√	√
Filter function	√	√	X	√
Images of the property	√	√	X	√
Way of contact the landlord	√	√	√	√
Location	X	X	X	√

2.5 Summary

From the literature review in chapter 2, it can conclude that the user mostly cares about the filter function when the user using the search function so that they can view the results based on their preferences such as type of property, prices range and numbers of room available. The user also concerns about the user interface of the system that they use. A user-friendly user interface attract the user to come back to use the app even more. Therefore, the findings that been discovered in this chapter are implemented and be used to develop the project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In the previous chapter 2, literature review has been made to identify the advantages and disadvantages of the existing system. The findings in chapter 2 will be propose to the project. This chapter also contain the methodology that is used to develop the project which is waterfall model.

3.2 Waterfall model

The Waterfall model was the first SDLC methodology to be applied to software development. It is really easy to comprehend and utilise. In a waterfall model, there is no overlap between stages and each phase must be finished before the subsequent step can start. The waterfall Model illustrates the software development process as a sequential, linear process. This indicates that any step of the development process may start only after the one before it has finished. The stages do not overlap in this waterfall model (*SDLC - Waterfall Model*, n.d.).

The Waterfall model is suitable for this project because short time frame for the project to be complete. This methodology is easy to manage as each process will have a certain deliverable and a review process. As this project is considered as a small project and the requirement are well understood make the Waterfall model is suitable for developing the project.

The project is breakdown into several phase. Begin by gathering detailed requirements from stakeholders, followed by creating a system design based on those requirements. Then, implement the software, and conduct testing to ensure it meets the requirements, and proceed to deployment. After deployment, provide maintenance and support. Careful planning and documentation are crucial in this model, as changes become challenging once a phase is completed. As can be seen in the Appendix A: gantt chart.

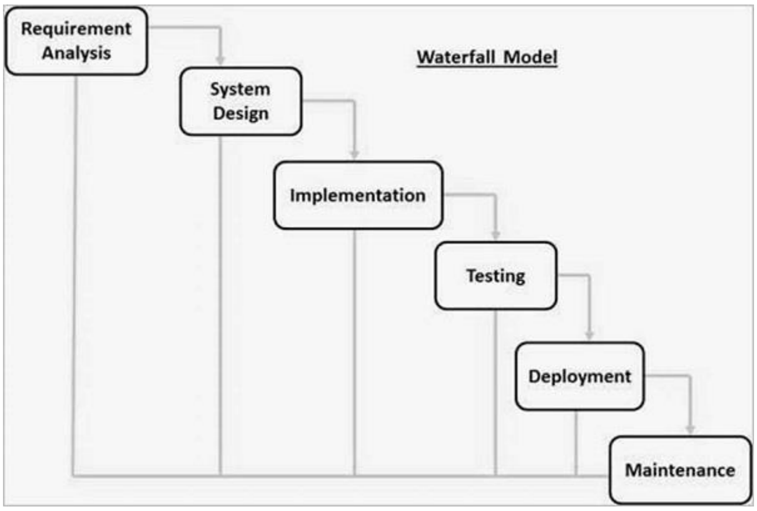


Figure 3.1 Waterfall model

3.2.1 Requirement Analysis

In this phase, the requirement for the project is gathered by taking surveys from the landlords and students using Google Form to collect information and to determine the target audience for the system. After the requirement has been determined, the plan for the next phases can be planned correctly. All the requirements frequently will be documented in the Software Requirement Specification in Appendix B

3.2.2 System Design

In this phase, the required specifications from the previous phase are examined, and the system design is created. This system design aids in determining the overall system architecture as well as the hardware and system requirements (*What Is WaterFall Model in Software Developement Life Cycle | SDLC*, n.d.). The system will be using MVC architecture that will have three interconnected elements that are related to the program logic. In this phase also the database design of the system is done to know what data that will be included in the system. The interface design of the system is designed using Figma to give an overview about how the system works in real scenario. All the design in documented in Software Design Document in Appendix C.

3.2.3 Implementation

In this phase, the deliverables from analysis and design phase are executed to develop the project. In the implementation phase we also develop the source code for the project by referring to the requirements and specification that is identified from the previous phases. The source code will be write using VS code in this phase covers the front end and the back-end language for the system. XAMPP is used setup the local environment for the development before the system is deploy in live server to be use by the users.

3.2.4 Testing

Testing phase is where we do the software verification and validation to make sure that the project is developed fulfil all the requirement that have been agreed in the previous phases. The testing will be using User Acceptance Test (UAT) to validate the system(*What Is User Acceptance Testing (UAT)? - Definition from Techopedia,* n.d.). Several users will be using the system with few test cases to test the system to identify faults and bugs in the system. The feedbacks from the testing will be collect and then changes will be made to the system to improve the system The UAT result is attach in Appendix D

3.2.5 Deployment

This phase is where the system is deployed in real environment that can be use by the user after going through verification and validation process in the previous phase. The system will be deployed in a real server that can be access by the public to be use.

3.2.6 Maintenance

The process of upgrading, modifying, and updating software to keep it current with user demands is known as software maintenance. After a product has been released, software maintenance is carried out for a variety of purposes, such as to enhance the programme generally, to fix problems or bugs, to improve performance, and more (*4 Types of Software Maintenance & What Is Software Maintenance?*, n.d.) .

3.3 Project Requirement

3.3.1 Functional Requirement

i. The system should allow the users to login and register.

ii. The system should allow the user to search for the house and room to rent.

iii. The system should allow the user to filter the search result based on their preferences such as price.

iv. The system should have display images of the property

v. The system should allow the user to save the contact number of landlord into WhatsApp.

vi. The landlord should be able to make payment for advertising their property in the system.

vii. The system should be able to generate report in pdf format to the PETAKOM.

3.3.2 Non-Functional Requirement

i. Security

The system must apply proper security to the system such as session and password hash function to make sure that sensitive data of the user are safe.

ii. Performance

The system should load the result not more than 10 seconds.

iii. Compatibility

The system can be use in any browser on any devices regardless of its operating system.

3.3.3 Constraint and Limitation

i. Time

The time to complete the project is quite short and the project may not be able to be completed on time.

ii. Resources

The project does not have enough resources to be delegate and need to be done one by one make it consume more time to finish.

iii. Quality

As the time for the project is short and the resource of the project is also limited. Hence, this factor can influence the project quality.

3.4 Propose Design

3.4.1 Context Diagram

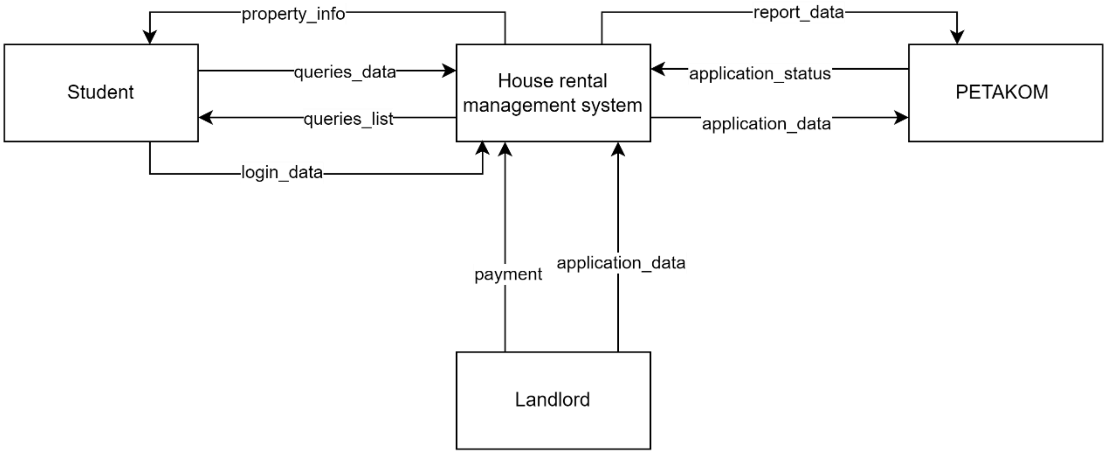


Figure 3.2 Context Diagram

The Figure 3.2 shows the context diagram for the house rental management system. the system will have three entities which are Student, PETAKOM, and Landlord. The entities will have their own functions and data flow when using the house rental management system. The student login to the system. They can also view the property info in the system. They can search for the house or room to rent in the system. The Landlord can apply for account to advertise their property in the system. They can upload their proof of payment into the system. PETAKOM can view the report data and application by the Landlord in the system. They can update the application status in the system.

3.4.2 Use Case Diagram

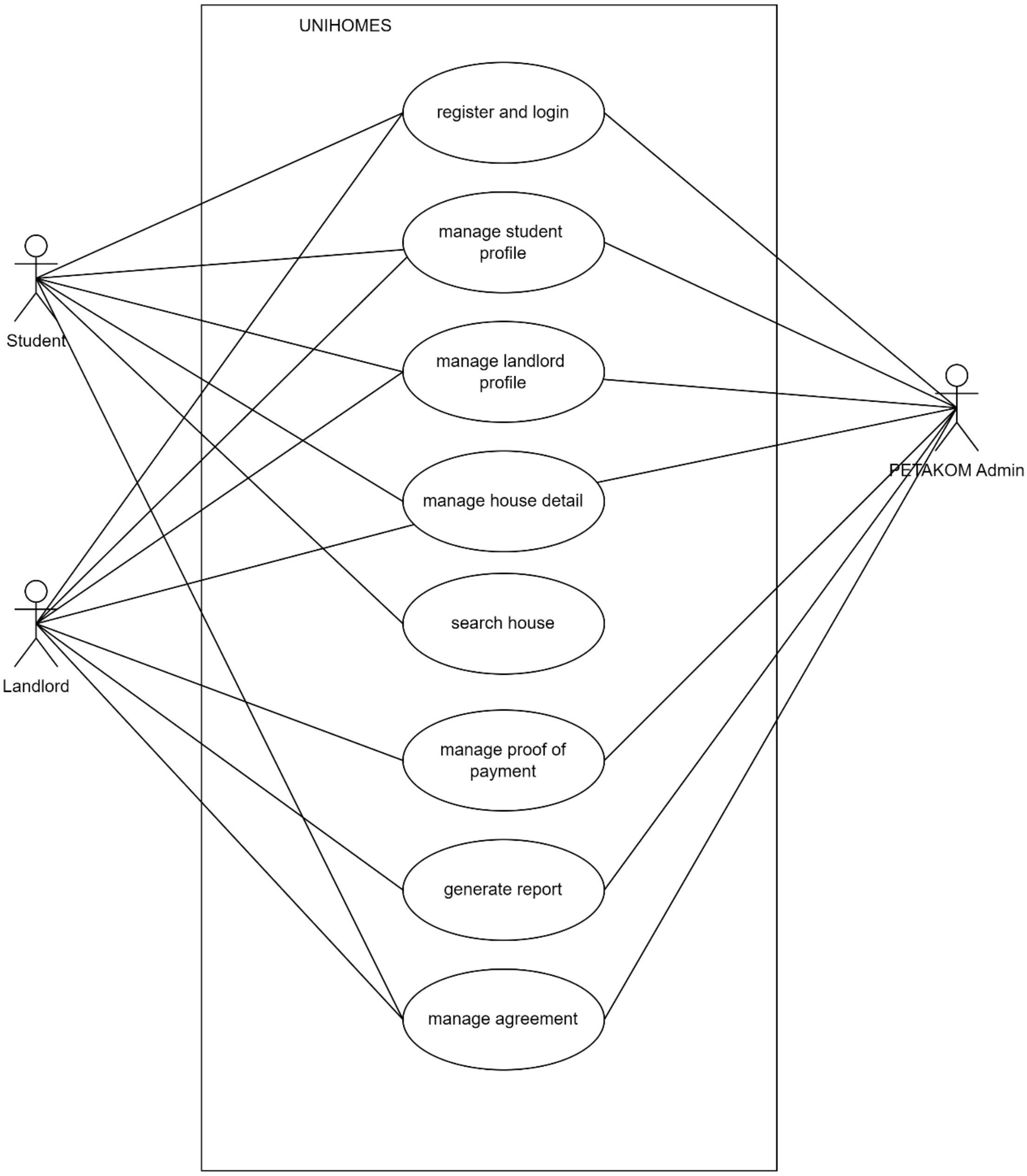
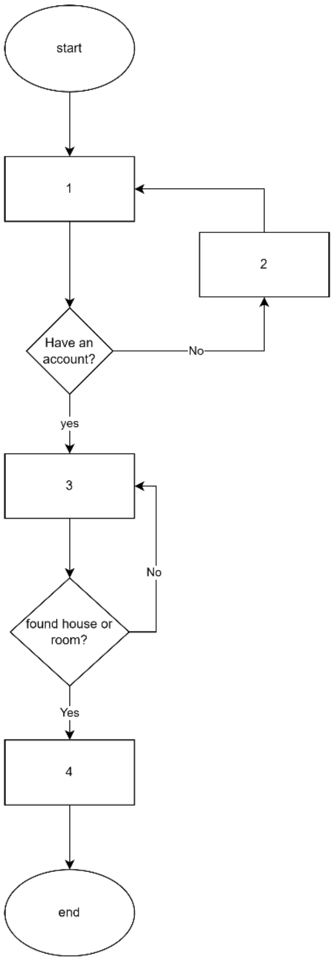


Figure 3.3 Use case diagram

Figure 3.3 shows the use case diagram for the house rental management system. The use case has three actors which are Student, Landlord and PETAKOM. There are eight use cases that will be in the house rental management system. The first use case is the register and login use case that will involve all the actors. The second use case is the manage student profile that also will involve all the actors. The student can view their profile and the landlord profile, they also can update and delete their profile. Next is the manage landlord profile. In the use case the landlord can view, update and delete their profile. The student and PETAKOM can view the landlord profile. The manage house detail use case will enable the actors to view, update, create and delete the information about the house. The use case search house is used by the student to search for house and room to rent. The manage review use case allow the user to create, view, update and delete the review about the property. The proof of payment involves the Landlord and PETAKOM. The landlord can upload the receipt of payment as a proof that they have paid.

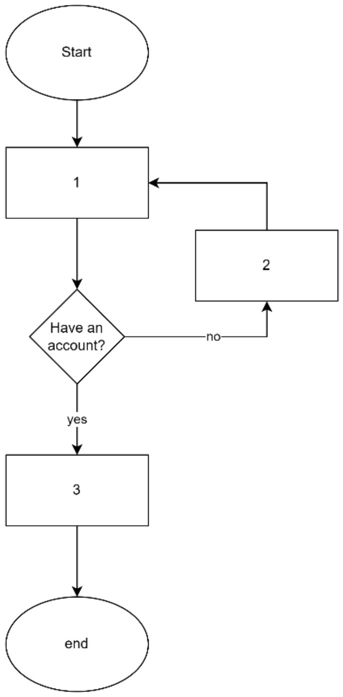
3.4.3 Flowchart

Figure 3.4 Flowchart for student

PROCESS	ACTION	RESPONSIBILITY
 start 1 2 Have an account? No yes 3 No found house or room? Yes 4 end	1. Login into the system. 2. Register to the system. Does the user have an account? 3. Search for room Does the user found the house or room? 4. Contact the landlord	Student Student Student Student

The figure 3.4 shows the flow for the student when they are using the system. the flow starts with the student login to the system. if they do not have account yet, they need to register first if they want to use the system. Once they are logged in the can search for house or room that they want to rent.

Figure 3.5 Flowchart for landlord

PROCESS	**ACTION**	**RESPONSIBILITY**
 Start 1 2 Have an account? no yes 3 end	1. Login into the system. 2. Register to the system. Does the user have an account? 3. Post house or room to rent	Landlord Landlord Landlord

The figure 3.5 shows the flowchart for the landlord when they want to post houses or rooms that they want to rent out to the students. The landlord flow starts with login into the system. if they do not have an account, they need to register one. Once they registered, they can start posting house or room to be rent.

Figure 3.6 Flowchart for PETAKOM

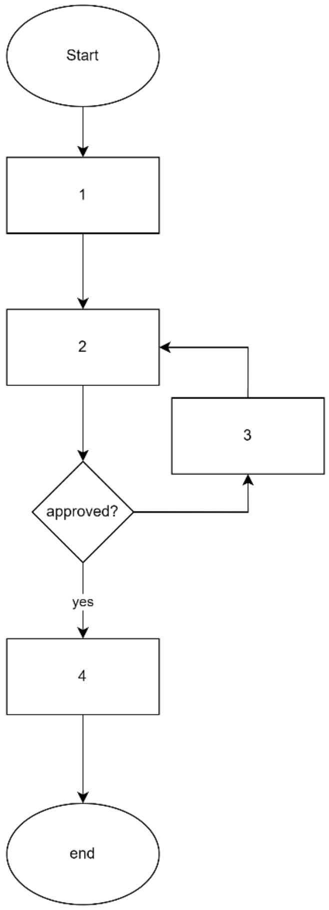
PROCESS	ACTION	RESPONSIBILITY
	1. Login into the system. 2. Manage user 3. Delete a user? Does the user deleted? 4. User deleted	PETAKOM PETAKOM PETAKOM Landlord

Figure 3.6 shows the flowchart for the PETAKOM. The PETAKOM must login to the system. Once they are logged into the system, they can manage post and manage user.

3.4.4 Activity Diagram

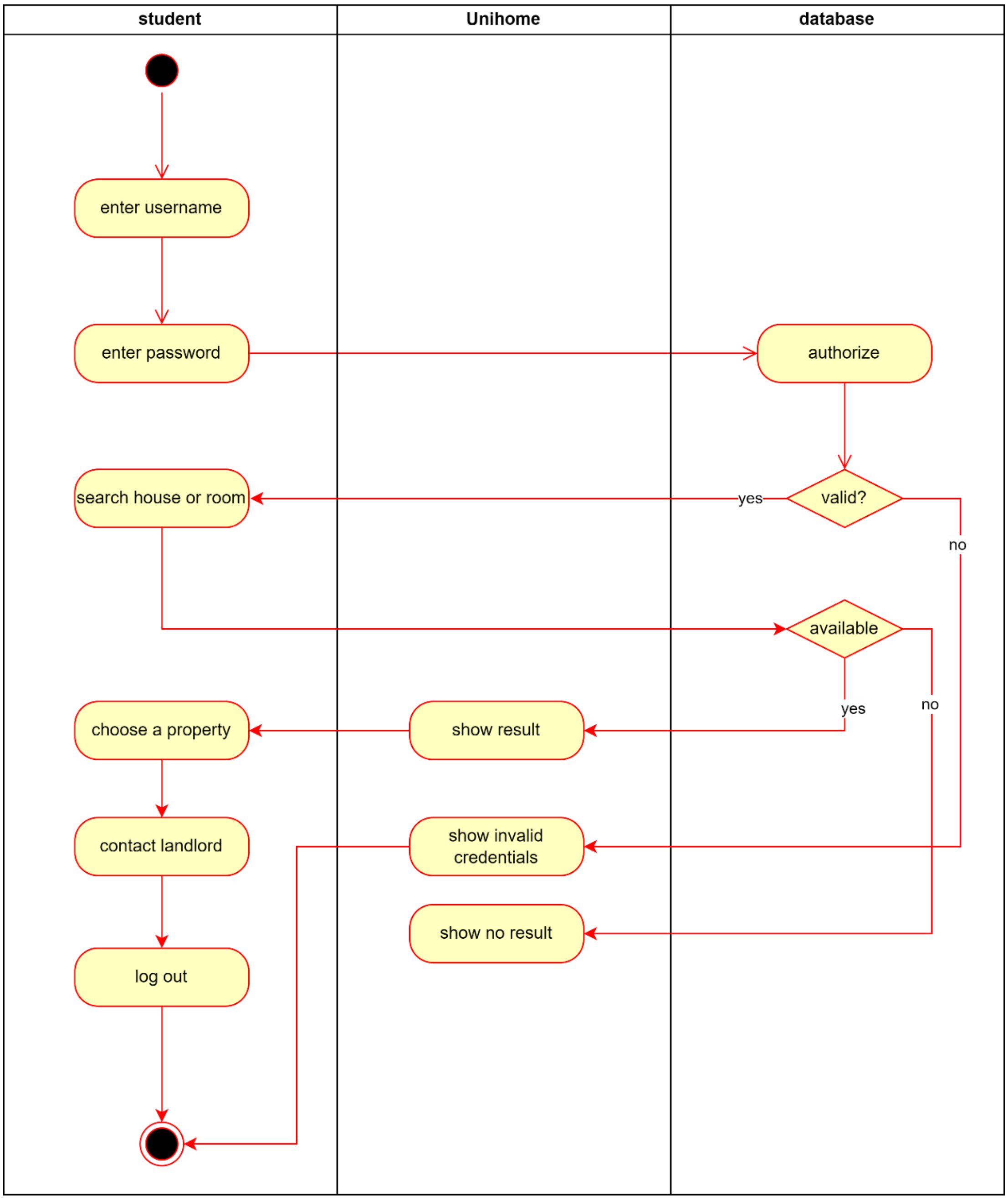


Figure 3.7 Activity Diagram for student

The activity diagram in figure above shows the flow for the student. They start by entering their username and password. The username and password then are validated to determine either it is in databased or not. If they credential matched, they can start searching for house or room to be rented. The system will display the search result. Then the student choose with property to rent and start to contacting the landlord.

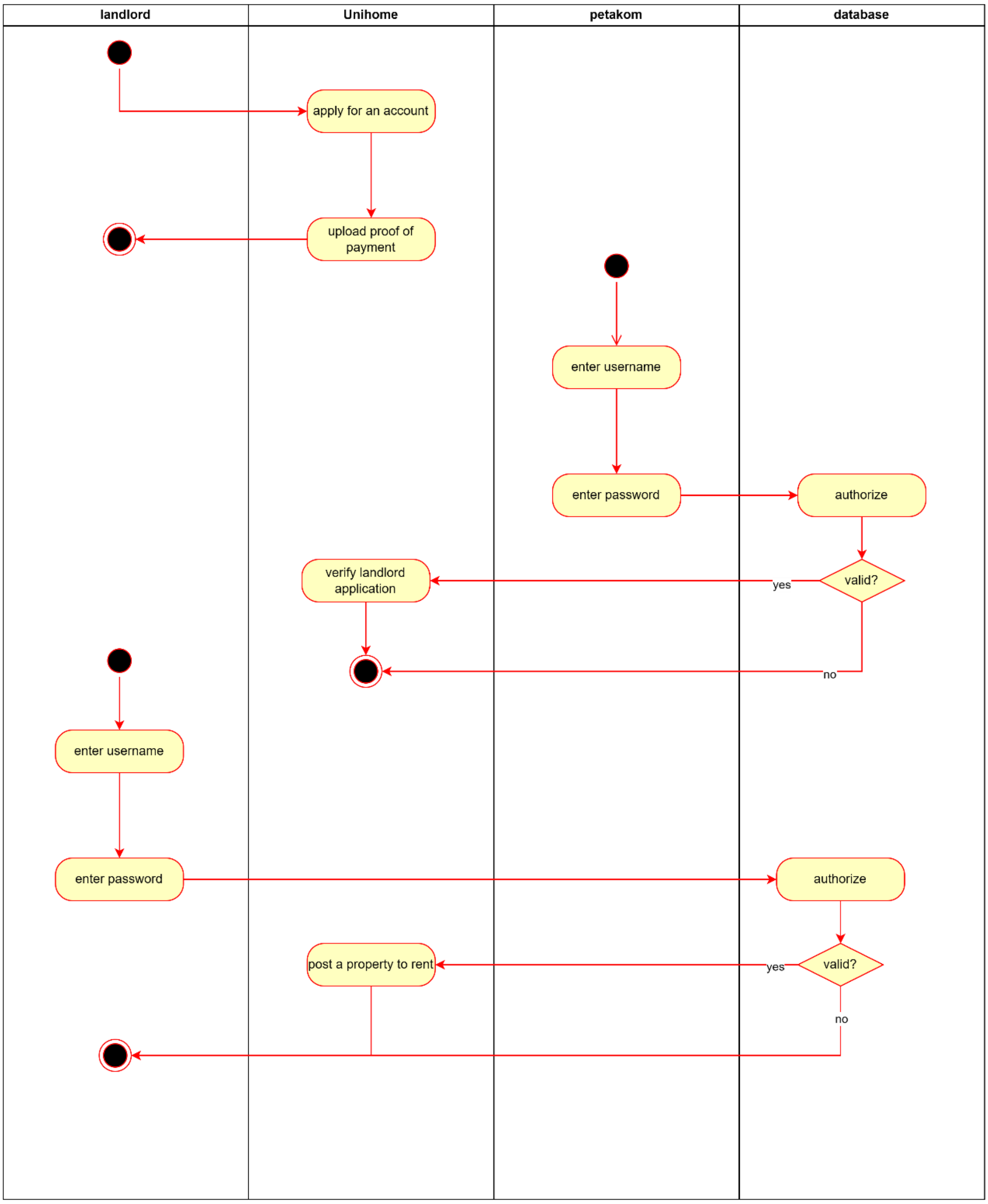


Figure 3.8 Activity Diagram for Landlord and PETAKOM

The figure above show the activity diagram for landlord and PETAKOM. The landlord can register for account and wait for their registration is approved by

PETAKOM. Once their registration is approve, they can login to the system to post their property.

3.5 Data Design

3.5.1 Entity Relationship Diagram

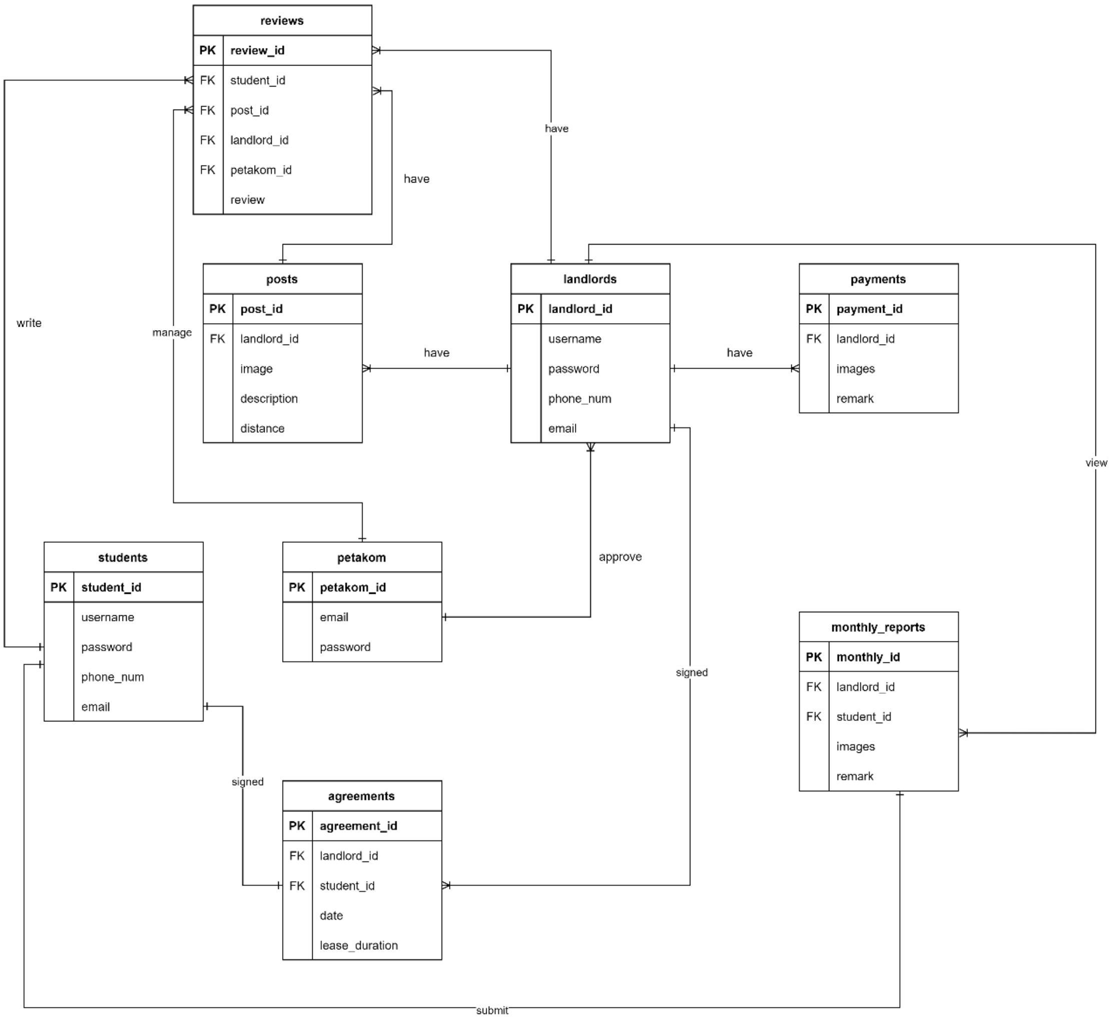


Figure 3.9 ERD for UniHomes

3.5.2 Data Dictionary

Table 3.1 users

Field name	Data type	Length	constraint	Description
id	Integer	250	Primary key	Student id , auto generate
name	Varchar	250	Not null	User name
email	Varchar	250	Not null	Email for login
password	Varchar	250		User password or login
remember_token	Varchar	250		Token for session
created_at	timestamp			Time and date when the user is created
updated_at	timestamp			Time and date when the user is updated

Table 3.2 posts

Field name	Data type	Length	constraint	Description
id	Integer	20	Primary key	Post id
user_id	Integer	20	Foreign key	User id
house_name	Integer	250		Name of the house
price_per_month	varchar	250		Price per month
phone_number	varchar	250		Landlord phone number
description	Varchar	250		Description of the house
address	varchar	250		Address of the house
number_of_room	varchar	250		Number of rooms in the house
house_capacity	Varchar	250		Capacity of the house
image_path	varchar	250		Path to store image
status	int			Status of the post
latitude	varchar	250		

longitude	varchar	250		
created_at	timestamp			
updated_at	timestamp			

Table 3.3 payments

Field name	Data type	Length	constraint	Description
payment_id	Integer	250	Primary key	payment id , auto generate
landlord_id	Integer	250	Foreign key	Landlord id
image	varchar	250		Image of payment receipt
remark				

Table 3.4 agreements

Field name	Data type	Length	constraint	Description
agreement_id	Integer	250	Primary key	payment id , auto generate
landlord_id	Integer	250	Foreign key	Landlord id
student-id	varchar	250	Foreign key	Image of payment receipt
date	Date	250		Agreement date
lease_duration	Varchar	250		Lease duration

3.6 Proof of Initial Concept

This part contains the initial concept of early work which is the prototype of the project. The prototype shows the interface for the PETAKOM, landlord and student as below.

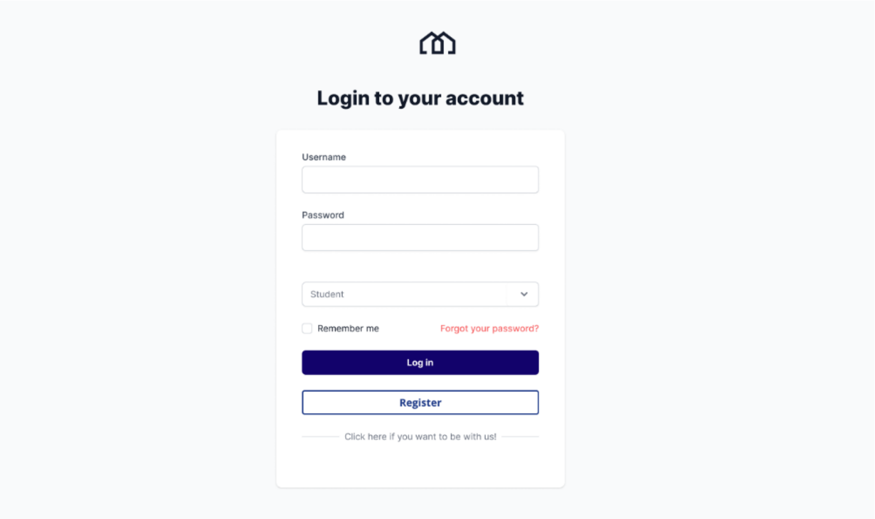


Figure 3.10 Login interface

The figure above shows the user interface for the user to login. They need to login using their email as username and key in the password.

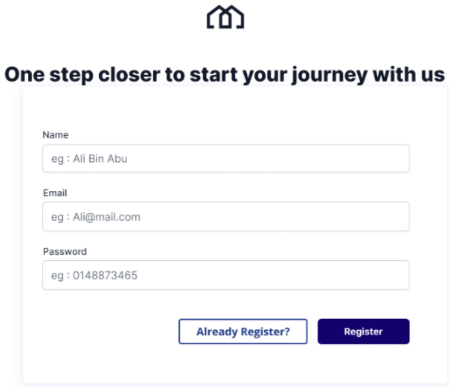
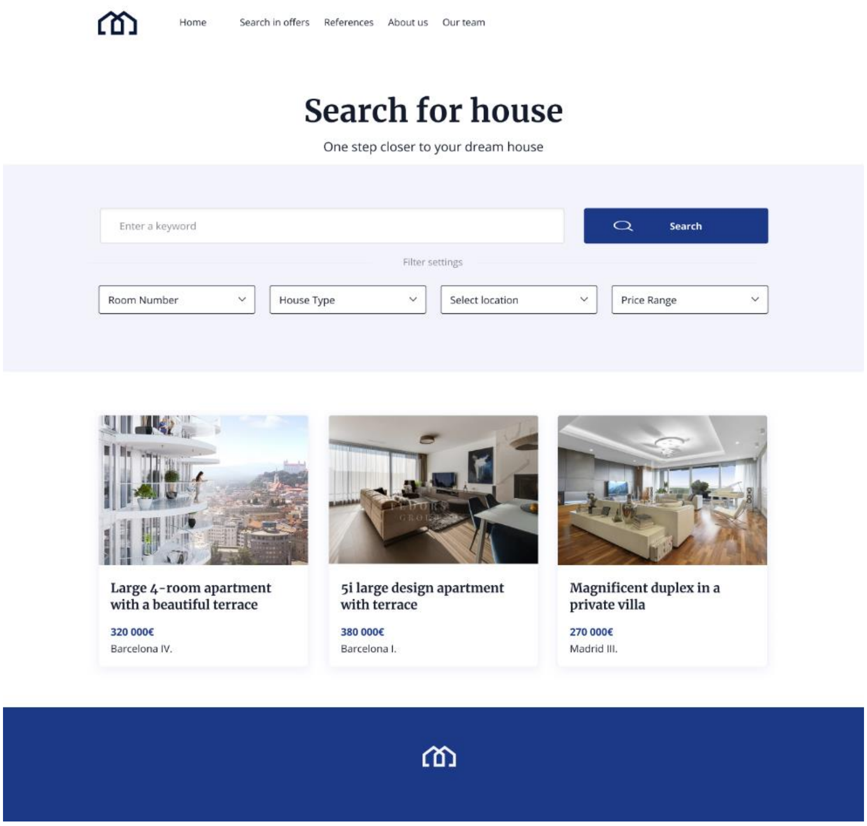


Figure 3.11 Register interface

The figure 3.11 shows the registration form for the user to register for an account. The user needs to input their name, email, and password.



Search for house

One step closer to your dream house

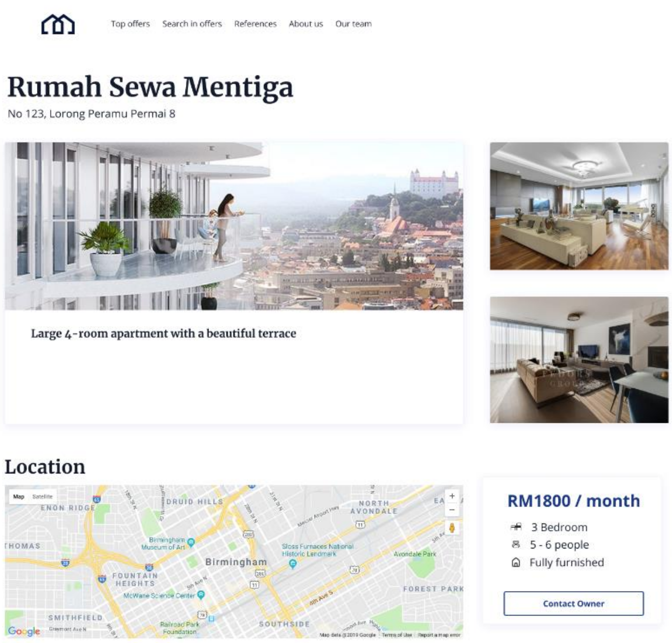
Large 4-room apartment with a beautiful terrace
320 000€
Barcelona IV.

5i large design apartment with terrace
380 000€
Barcelona I.

Magnificent duplex in a private villa
270 000€
Madrid III.

Figure 3.12 student Homepage

The figure 3.12 shows the prototype of the homepage for the system. the homepage will show list of available rooms and house to be rent.



Rumah Sewa Mentiga

No 123, Lorong Peramu Permai 8

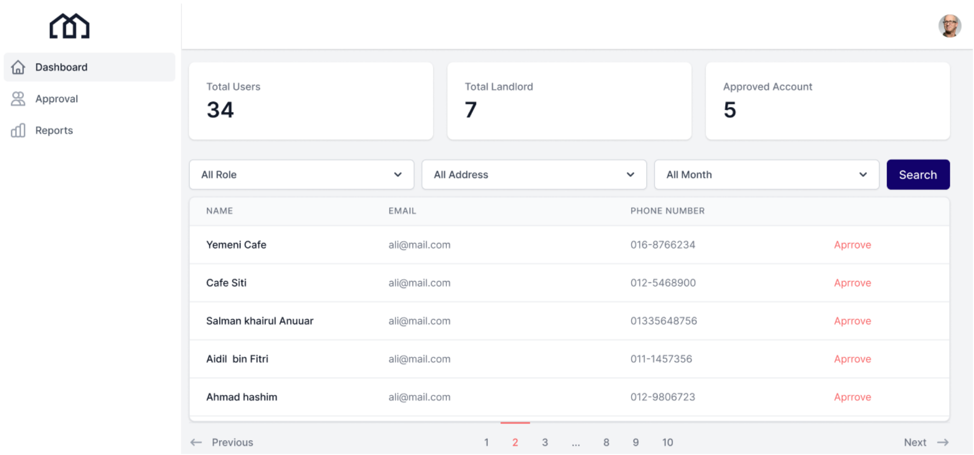
Large 4-room apartment with a beautiful terrace

RM1800 / month

3 Bedroom
5 - 6 people
Fully furnished

Figure 3.13 Post Interface

The figure 3.13 shows the user interface for each post. The student can view the interior images of the house, price per month, house information and button to contact the landlord.



Total Users 34

Total Landlord 7

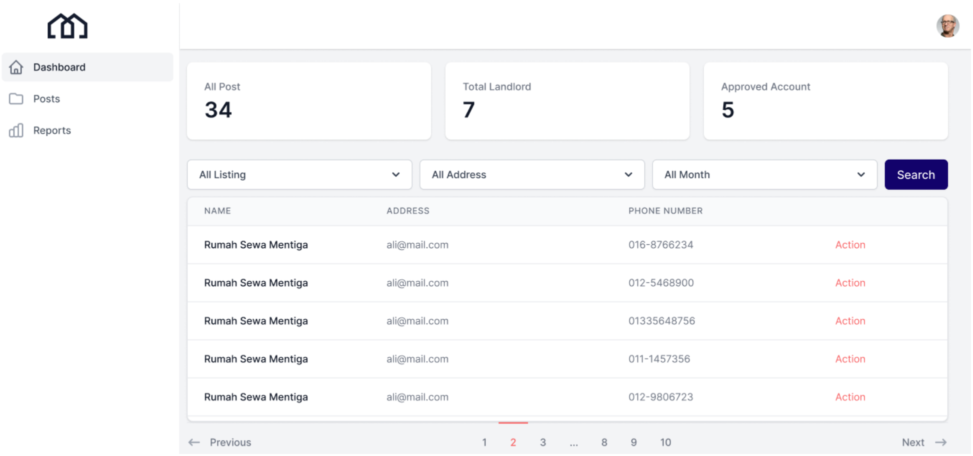
Approved Account 5

All Role | All Address | All Month

NAME	EMAIL	PHONE NUMBER	
Yemeni Cafe	ali@mail.com	016-8766234	Aprrove
Cafe Siti	ali@mail.com	012-5468900	Aprrove
Salman khairul Anuuar	ali@mail.com	01335648756	Aprrove
Aidil bin Fitri	ali@mail.com	011-1457356	Aprrove
Ahmad hashim	ali@mail.com	012-9806723	Aprrove

Figure 3.14 PETAKOM Dasboard

The figure 3.14 shows the dashboard for the PETAKOM. In this interface PETAKOM can manage posts, and view reports.



All Post 34

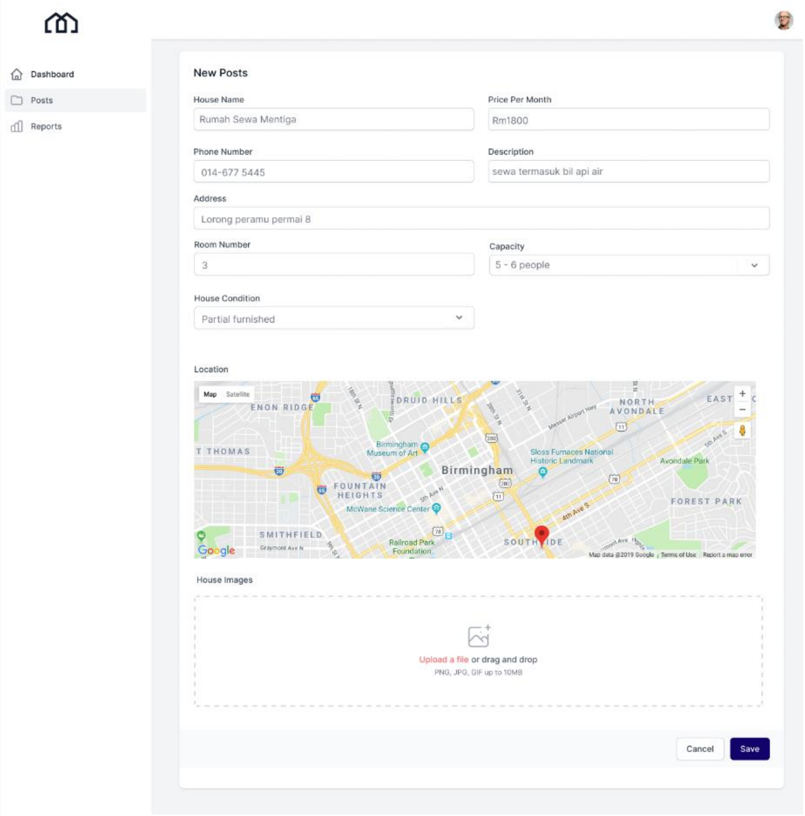
Total Landlord 7

Approved Account 5

NAME	ADDRESS	PHONE NUMBER	
Rumah Sewa Mentiga	ali@mail.com	016-8766234	Action
Rumah Sewa Mentiga	ali@mail.com	012-5468900	Action
Rumah Sewa Mentiga	ali@mail.com	01335648756	Action
Rumah Sewa Mentiga	ali@mail.com	011-1457356	Action
Rumah Sewa Mentiga	ali@mail.com	012-9806723	Action

Figure 3.15 Landlord Dashboard

The figure 3.15 shows the dashboard for the landlord. In this interface PETAKOM can manage posts, and view reports.



New Posts

House Name: Rumah Sewa Mentiga

Price Per Month: Rm1800

Phone Number: 014-677 5445

Description: sewa termasuk bil api air

Address: Lorong peramu permai 8

Room Number: 3

Capacity: 5 - 6 people

House Condition: Partial furnished

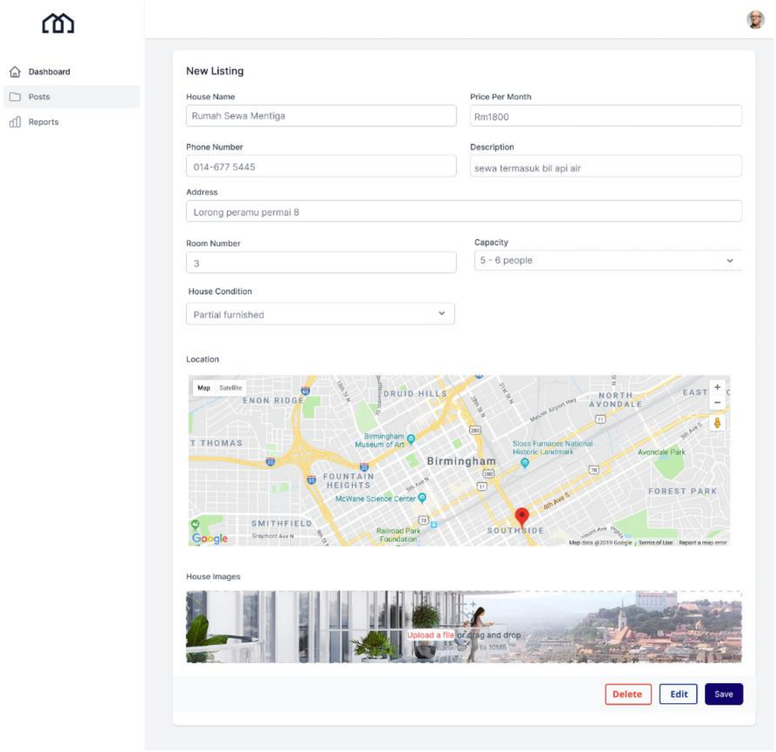
Location

House Images

Upload a file or drag and drop
PNG, JPG, GIF up to 10MB

Figure 3.16 Create Interface

The figure 3.16 shows the interface when the landlord wants to create a post about their house or room to be rent to the students.



New Listing

House Name: Rumah Sewa Mentiga

Price Per Month: Rm1800

Phone Number: 014-677 5445

Description: sewa termasuk bil api air

Address: Lorong peramu permai 8

Room Number: 3

Capacity: 5 - 6 people

House Condition: Partial furnished

Location

House Images

Figure 3.17 View Interface

The figure 3.1y shows the interface when the landlord wants to view a post about their house or room to be rent to the students.

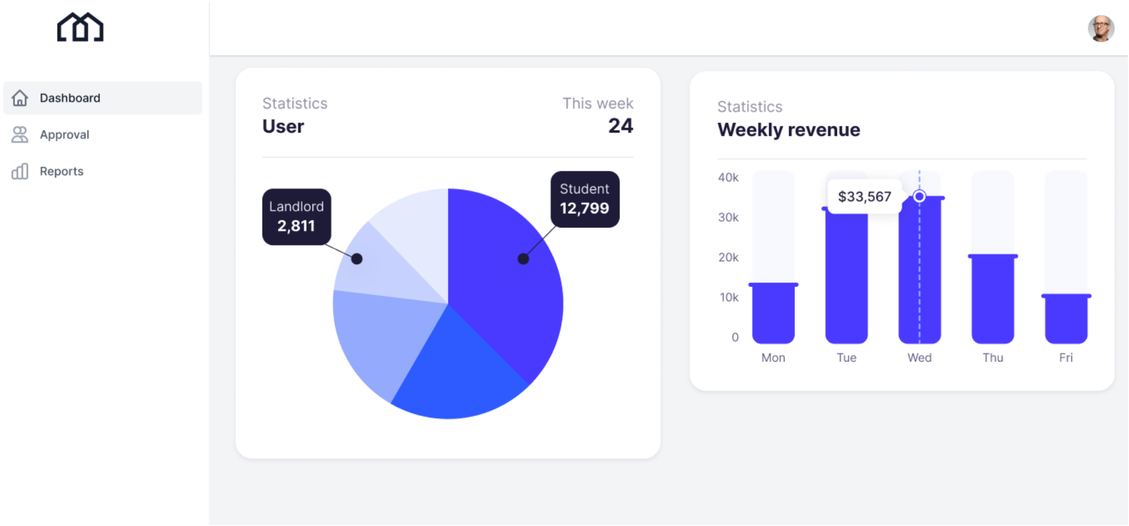
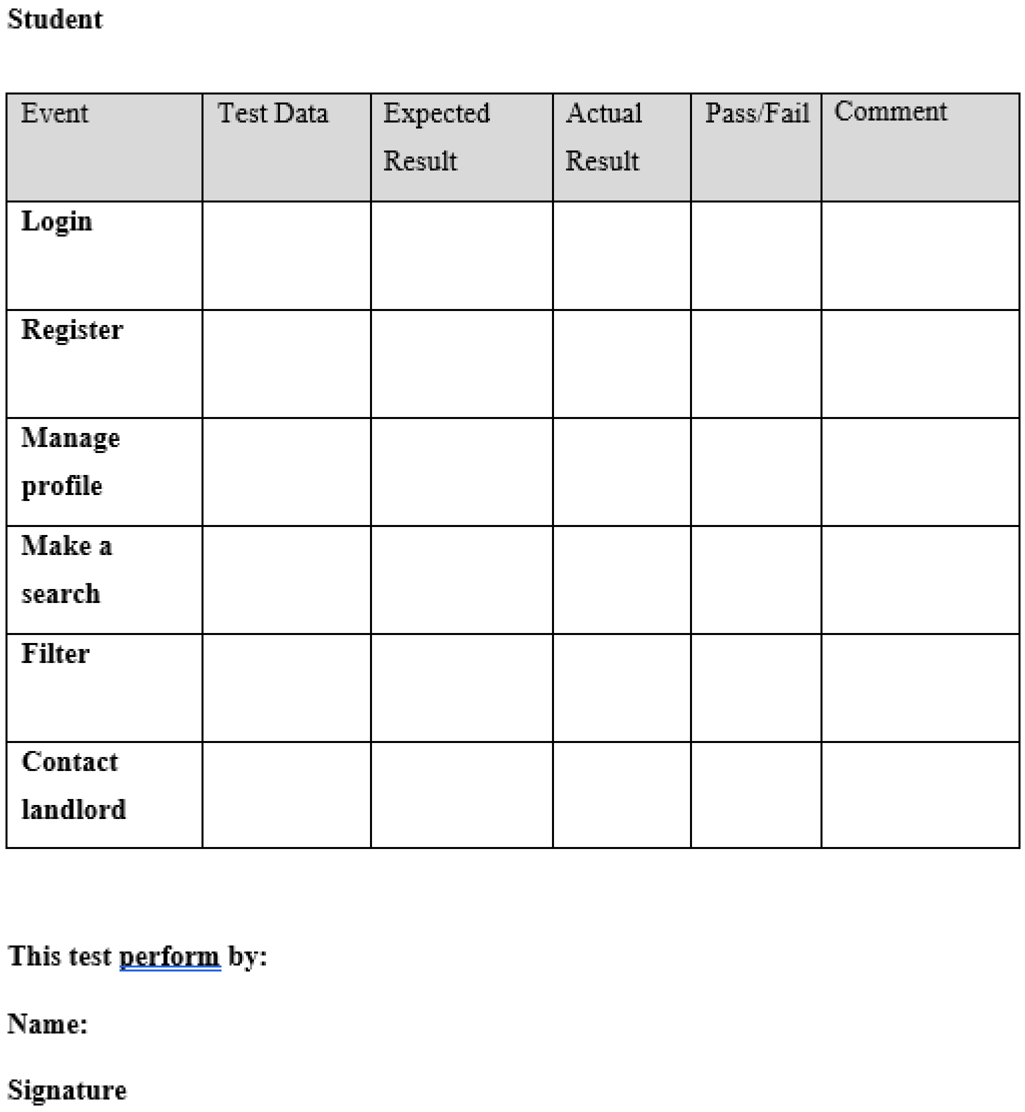


Figure 3.18 Report Interface

The figure 3.18 shows the report interface. In this interface the user can view the information regarding user and posts in graphical.

3.7 Testing and Validation Plan



Student

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login					
Register					
Manage profile					
Make a search					
Filter					
Contact landlord					

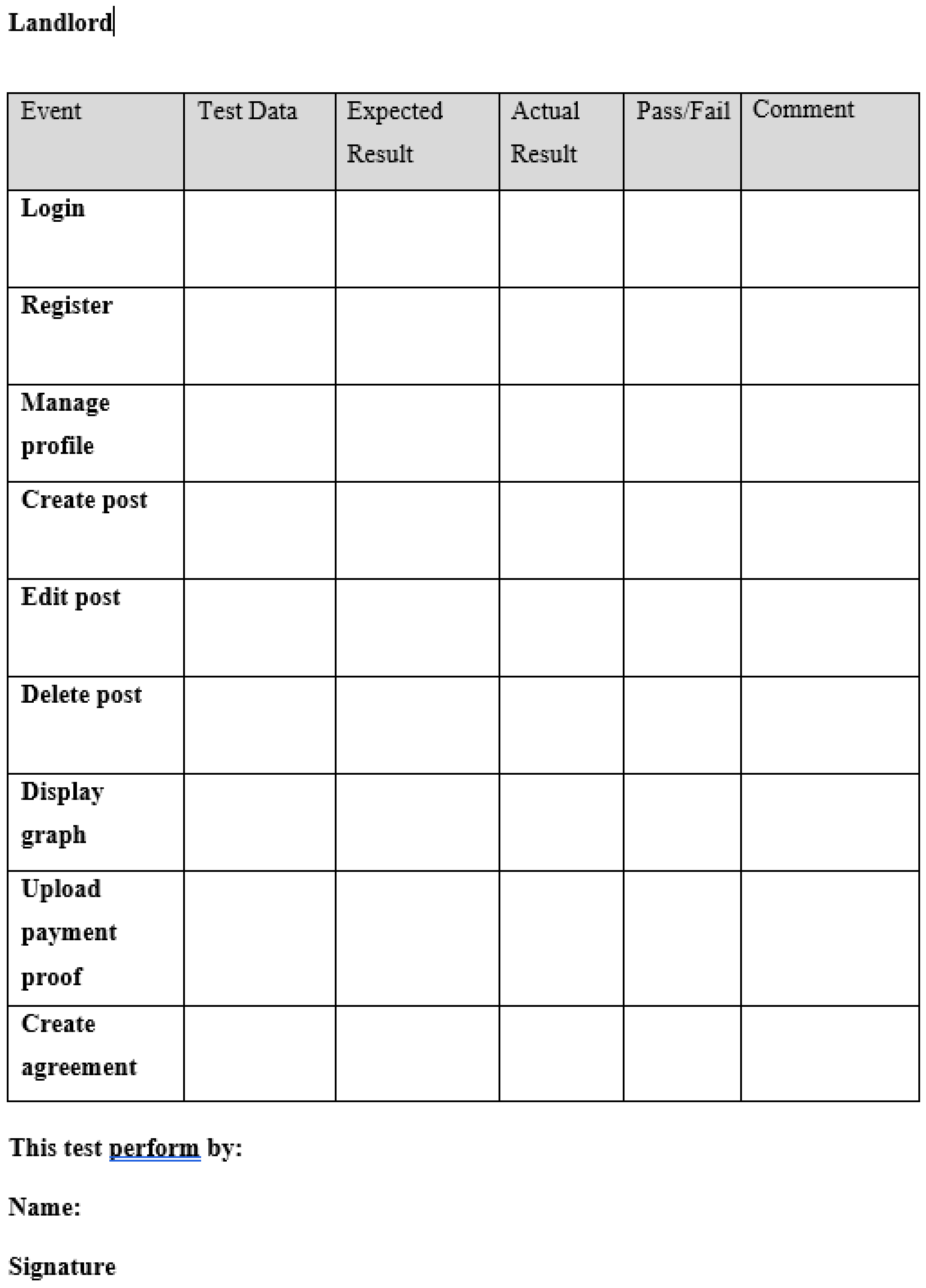
This test perform by:

Name:

Signature

Figure 3.19 UAT for student

Figure 3.19 shows that UAT form for student. The modules that are being test during the User Acceptance Test are the login module, register, manage profile, search function, filter function and contact landlord.



Landlord

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login					
Register					
Manage profile					
Create post					
Edit post					
Delete post					
Display graph					
Upload payment proof					
Create agreement					

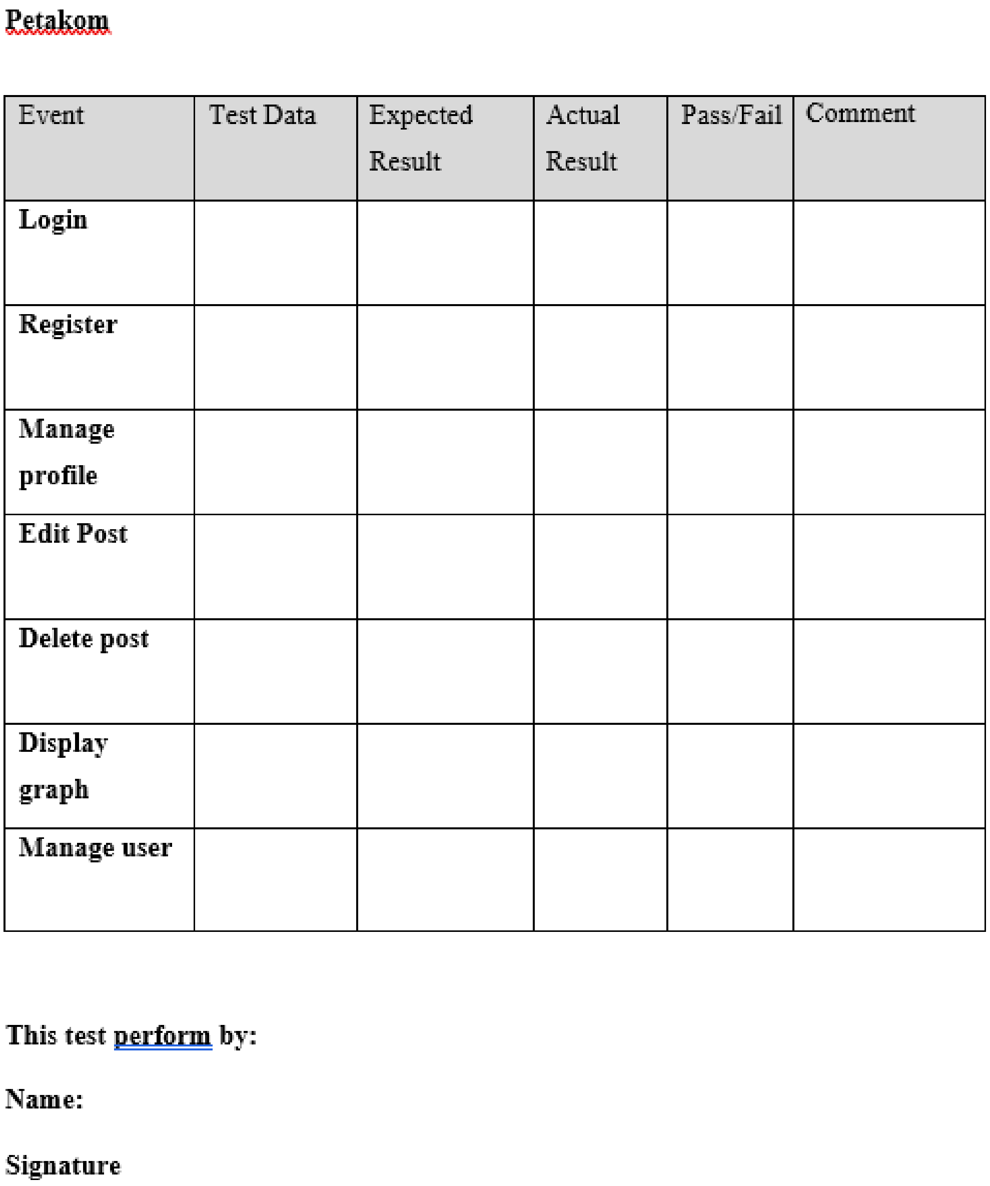
This test perform by:

Name:

Signature

Figure 3.20 UAT for landlord

Figure 3.20 shows that UAT for for landlord. The modules that are being test during User Acceptance Test for landlord are login, register, manage profile, create post, edit post, delete post, display graph, upload proof of payment and create agreement.



Petakom

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login					
Register					
Manage profile					
Edit Post					
Delete post					
Display graph					
Manage user					

This test perform by:

Name:

Signature

Figure 3.21 UAT for PETAKOM

Figure 3.21 shows that UAT for for PETAKOM. The modules that are being test during User Acceptance Test for landlord are login, register, manage profile, edit post, delete post, display graph, manage user.

3.8 Potential Use of Proposed Solution

The UniHomes can become one stop centre for student to search for their accommodation during their study. The web-based system will eliminate the risk of the student being scammed. UniHomes also can act as a source of income for PETAKOM to collect fund for the organization.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter discuss about the development, implementation, and testing of Unihome. This web-based system is implemented to the student at Universiti Malaysia Pahang. The system is developed using Laravel Framework, Visual Studio Code, MySQL, and Figma. The testing of the system is performed to identify any errors and bug and immediately fix it.

4.2 Implementation Process

The development of Unihomes starts with outlining the database of the system. Entity Relationship Diagram (ERD) is created to determine the entities that are related to the system. The ERD helps to specify what data that will be store, the entities and their attribute and also how the entities are relate to each other. The ERD then is implemented in phpMyAdmin by creating the database and tables for the Unihome system.

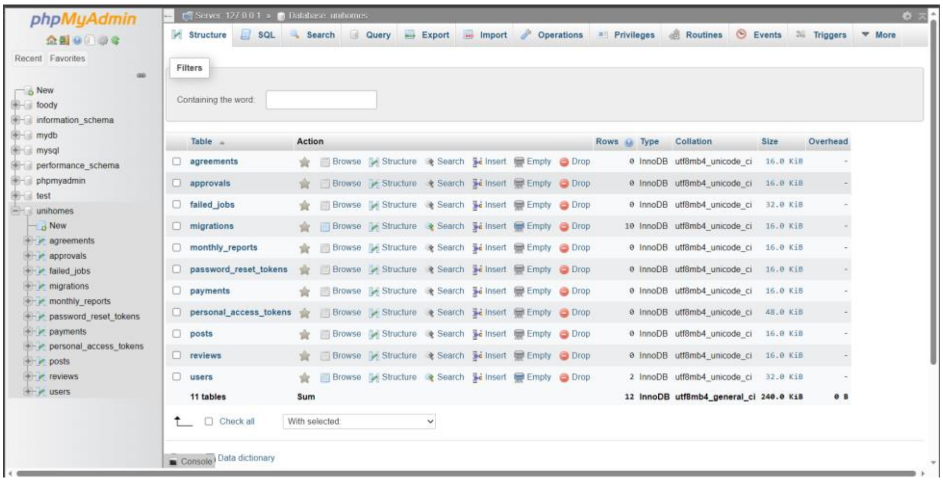


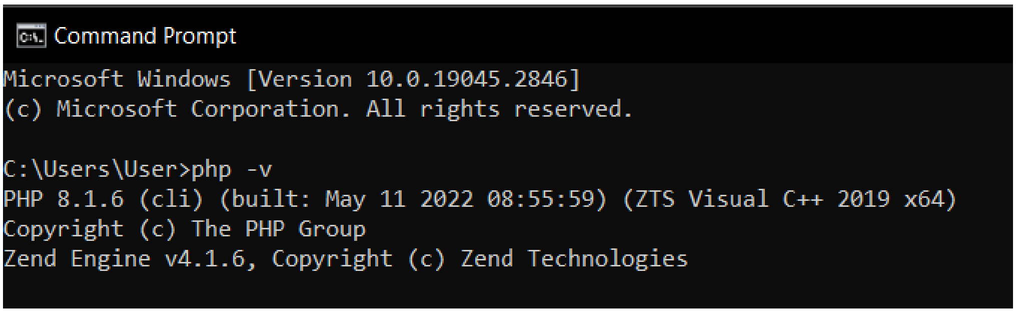
Table	Action	Rows	Type	Collation	Size	Overhead
agreements	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
approvals	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
failed_jobs	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
migrations	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
monthly_reports	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
password_reset_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
payments	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
personal_access_tokens	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	48.0 KiB	-
posts	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
reviews	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	-
11 tables	Sum	12	InnoDB	utf8mb4_general_ci	240.0 KiB	0 B

Figure 4.1 Tables in Unihome database.

The Unihomes is developed using Laravel Framework. The Laravel framework is web application framework with expressive, elegant syntax. It provides powerful features such a thorough dependency injection, impressive database abstraction layer, queues and scheduled job, unit and integration testing and more.

Before creating Laravel project, it requires:

i. PHP installed on your computer



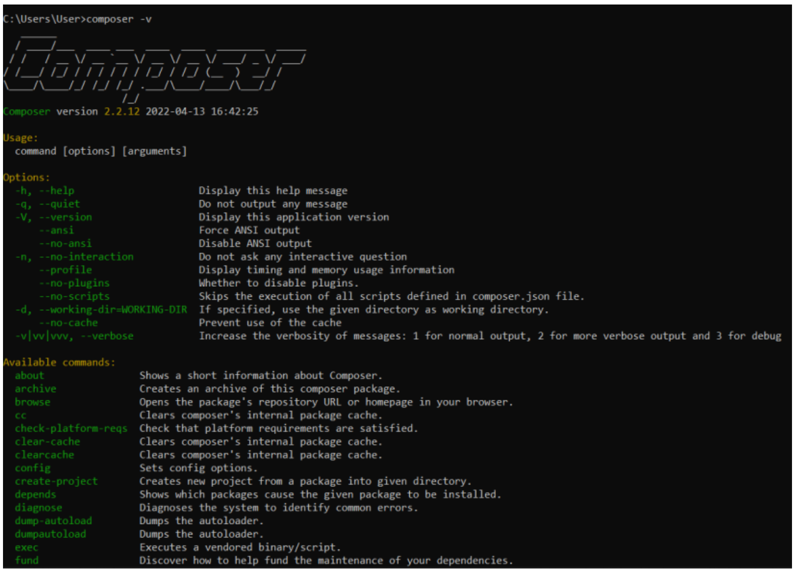
Command Prompt

```
Microsoft Windows [Version 10.0.19045.2846]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>php -v
PHP 8.1.6 (cli) (built: May 11 2022 08:55:59) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.1.6, Copyright (c) Zend Technologies
```

Figure 4.2 PHP version

Figure 4.2 shows that PHP version 8.1.6 is installed on the computer.

ii. Composer is installed.

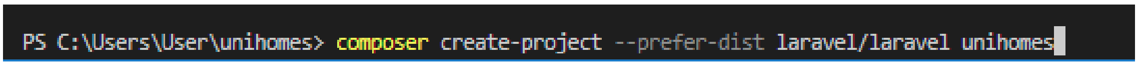


```
C:\Users\User>composer -v
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 2.2.12 2022-04-13 16:42:25

Usage:
  command [options] [arguments]

Options:
  -h, --help                     Display this help message
  -q, --quiet                    Do not output any message
  -V, --version                  Display this application version
      --ansi                     Force ANSI output
      --no-ansi                  Disable ANSI output
  -n, --no-interaction           Do not ask any interactive question
      --profile                  Display timing and memory usage information
      --no-plugins               Whether to disable plugins.
      --no-scripts               Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR  If specified, use the given directory as working directory.
      --no-cache                 Prevent use of the cache
  -v|vv|vvv, --verbose           Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                Shows a short information about Composer.
  archive              Creates an archive of this composer package.
  browse               Opens the package's repository URL or homepage in your browser.
  cc                   Clears composer's internal package cache.
  check-platform-reqs  Check that platform requirements are satisfied.
  clear-cache          Clears composer's internal package cache.
  clearcache           Clears composer's internal package cache.
  config               Sets config options.
  create-project       Creates new project from a package into given directory.
  depends              Shows which packages cause the given package to be installed.
  diagnose             Diagnoses the system to identify common errors.
  dump-autoload        Dumps the autoloader.
  dumpautoload         Dumps the autoloader.
  exec                 Executes a vendored binary/script.
  fund                 Discover how to help fund the maintenance of your dependencies.
```

Figure 4.3 Composer

Figure 4.3 shows that the composer have been installed in the computer.

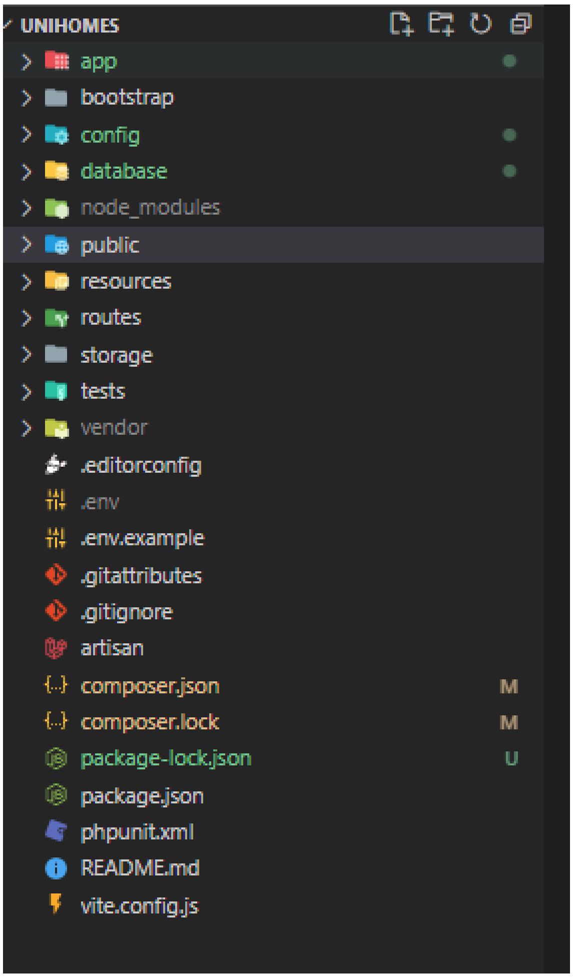
Create Laravel project



```
PS C:\Users\User\unihomes> composer create-project --prefer-dist laravel/laravel unihomes
```

Figure 4.4 Command to create Laravel project

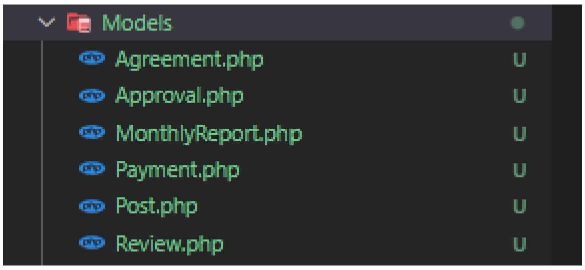
Figure 4.4 shows the command that need to be run in the terminal in Visual Studio Code to create the UniHomes using Laravel Framework.



```
UNIHOMES
> app                      ●
> bootstrap
> config                   ●
> database                 ●
> node_modules
> public
> resources
> routes
> storage
> tests
> vendor
  .editorconfig
  .env
  .env.example
  .gitattributes
  .gitignore
  artisan
  composer.json            M
  composer.lock            M
  package-lock.json        U
  package.json
  phpunit.xml
  README.md
  vite.config.js
```

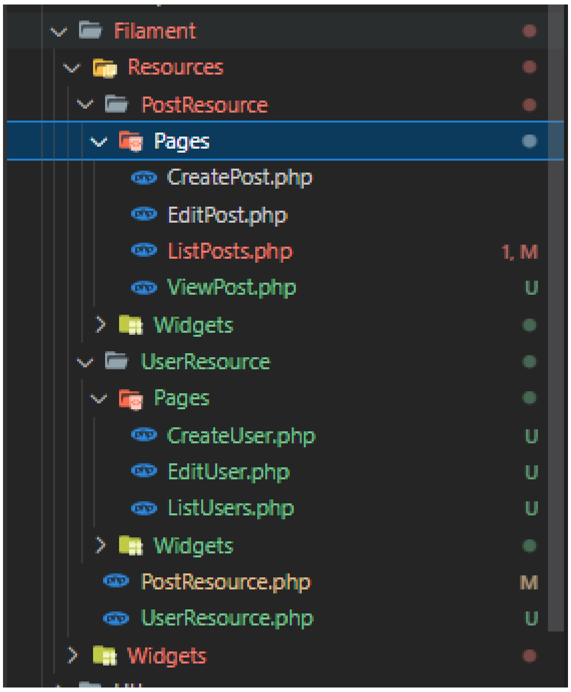
Figure 4.5 Laravel file structure

Figure 4.5 shows the file created in the Unihomes project. The Laravel project use the MVC architecture.



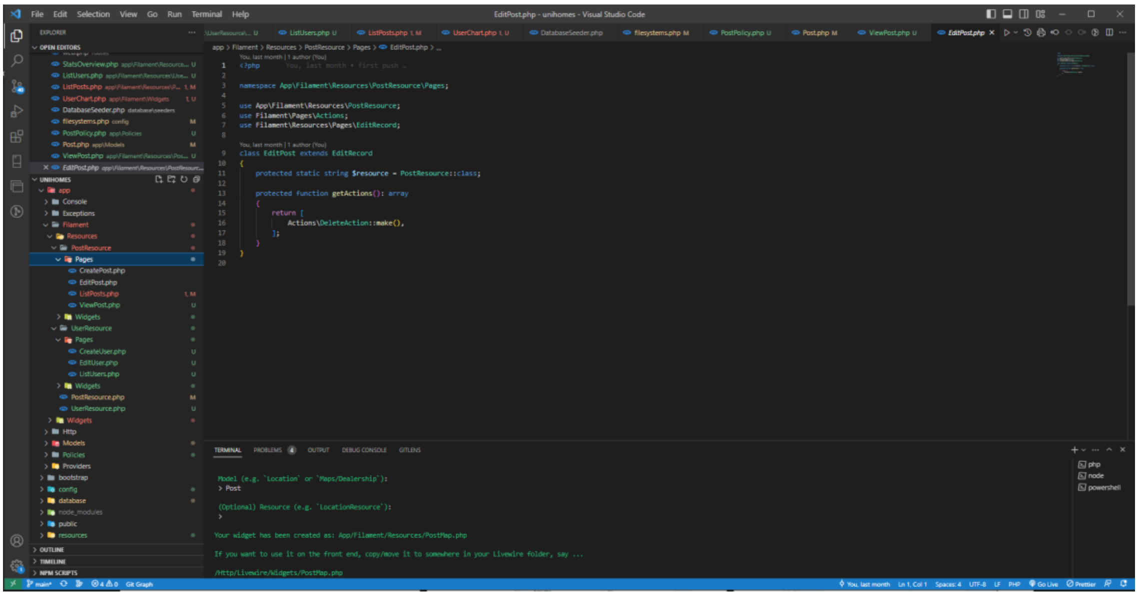
```
Models
  Agreement.php       U
  Approval.php        U
  MonthlyReport.php   U
  Payment.php         U
  Post.php            U
  Review.php          U
```

Figure 4.6 Models



```
Filament
  Resources
    PostResource
      Pages
        CreatePost.php
        EditPost.php
        ListPosts.php       1, M
        ViewPost.php        U
      Widgets
    UserResource
      Pages
        CreateUser.php      U
        EditUser.php        U
        ListUsers.php       U
      Widgets
    PostResource.php        M
    UserResource.php        U
  Widgets
```

Figure 4.7 Resources

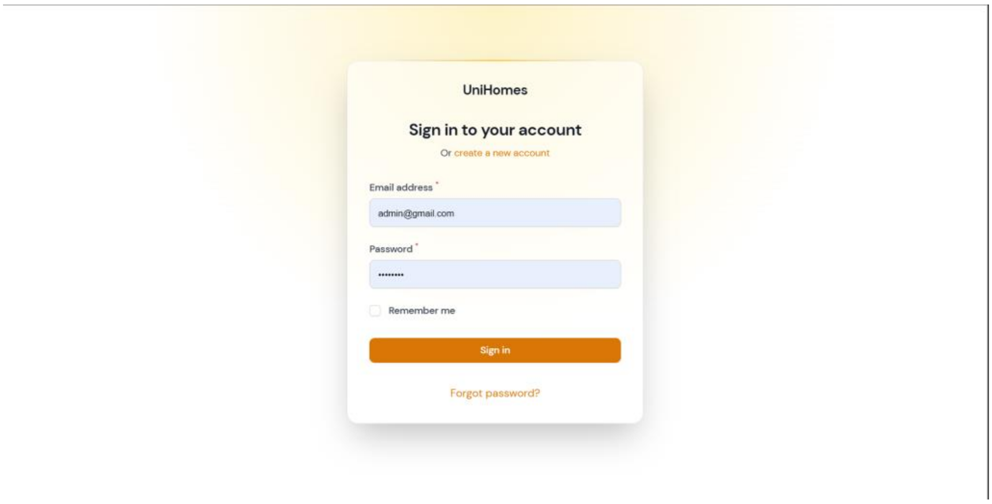

EditPost.php - unihomes - Visual Studio Code
app > Filament > Resources > PostResource > Pages > EditPost.php > ...
TERMINAL
PROBLEMS 4
OUTPUT
DEBUG CONSOLE
GITLENS

```php
<?php

namespace App\Filament\Resources\PostResource\Pages;

use App\Filament\Resources\PostResource;
use Filament\Pages\Actions;
use Filament\Resources\Pages\EditRecord;

class EditPost extends EditRecord
{
    protected static string $resource = PostResource::class;

    protected function getActions(): array
    {
        return [
            Actions\DeleteAction::make(),
        ];
    }
}
```

```
 Model (e.g. `Location` or `Maps/Dealership`):
 > Post

 (Optional) Resource (e.g. `LocationResource`):
 >

Your widget has been created as: App/Filament/Resources/PostMap.php

If you want to use it on the front end, copy/move it to somewhere in your Livewire folder, say ...

/Http/Livewire/Widgets/PostMap.php
```

Figure 4.8 Vs code interface

The code editor used to develop the system is Vs code. It is a versatile source code editor developed by Microsoft. It offers a wide range of features and extensions that enhance developer productivity and support multiple programming languages and platforms.


UniHomes

Sign in to your account

Or create a new account

Email address *

admin@gmail.com

Password *

••••••••

□ Remember me

Sign in

Forgot password?

Figure 4.9 Login

Figure 4.8 shows the login interface where the user will insert their username and password to login into Unihomes. If the username and password match the user can log in into the system.

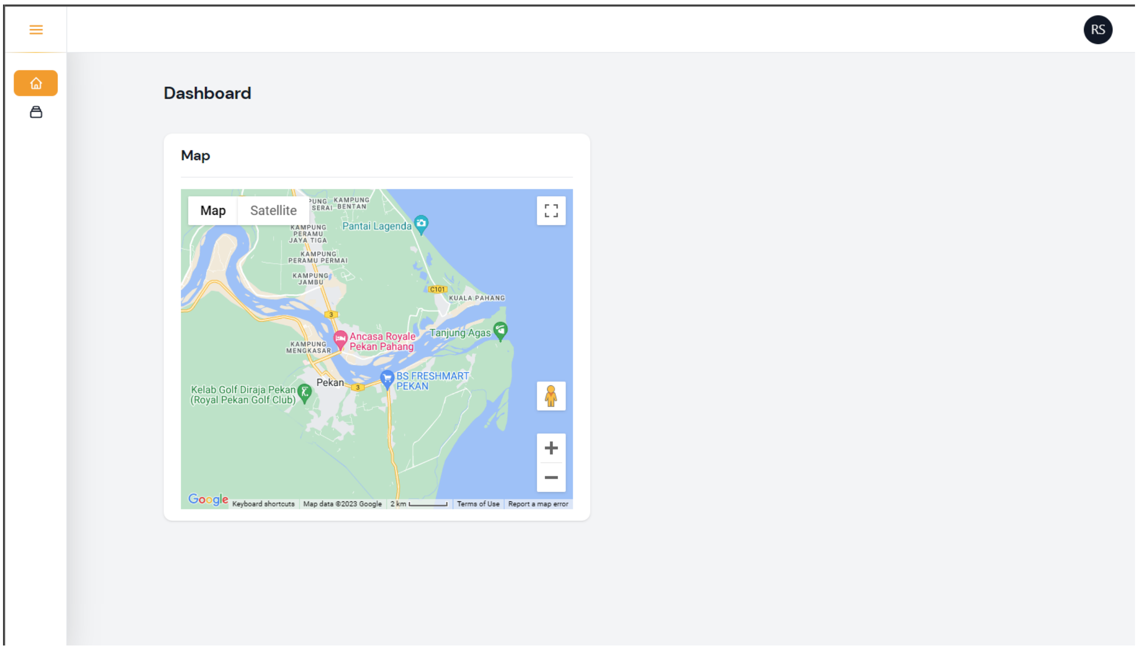
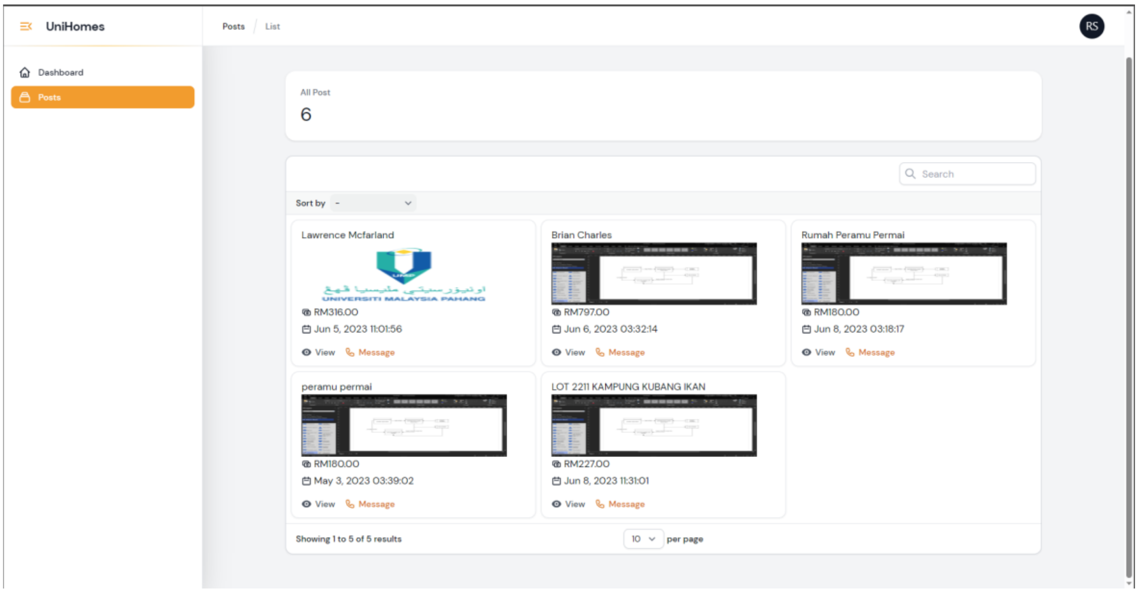


Figure 4.10 Google Map widget

The user will have the Google Map widget to allow them to see how many properties are available to be rent.



All Post
6

Sort by -

Lawrence Mcfarland
RM316.00
Jun 5, 2023 11:01:56

Brian Charles
RM797.00
Jun 6, 2023 03:32:14

Rumah Peramu Permai
RM180.00
Jun 8, 2023 03:18:17

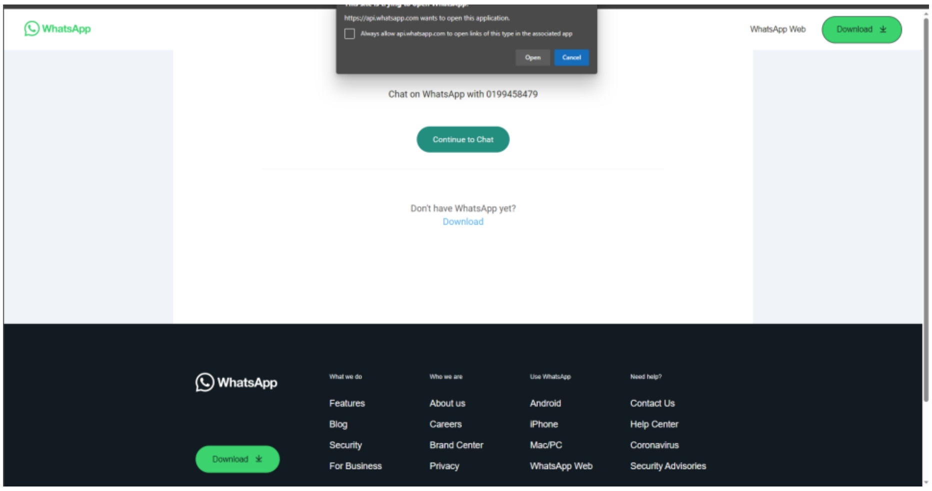
peramu permai
RM180.00
May 3, 2023 03:39:02

LOT 2211 KAMPUNG KUBANG IKAN
RM227.00
Jun 8, 2023 11:31:01

Showing 1 to 5 of 5 results
10 per page

Figure 4.11 Homepage

Figure 4.9 shows the main interface of UniHomes that the student will see after they log in into the system. in this interface there are dropdowns where they can filter out the houses or rooms that they want to rent based on their own preferences. There will be also few houses and rooms that is available to rent in the main page.



https://api.whatsapp.com wants to open this application.

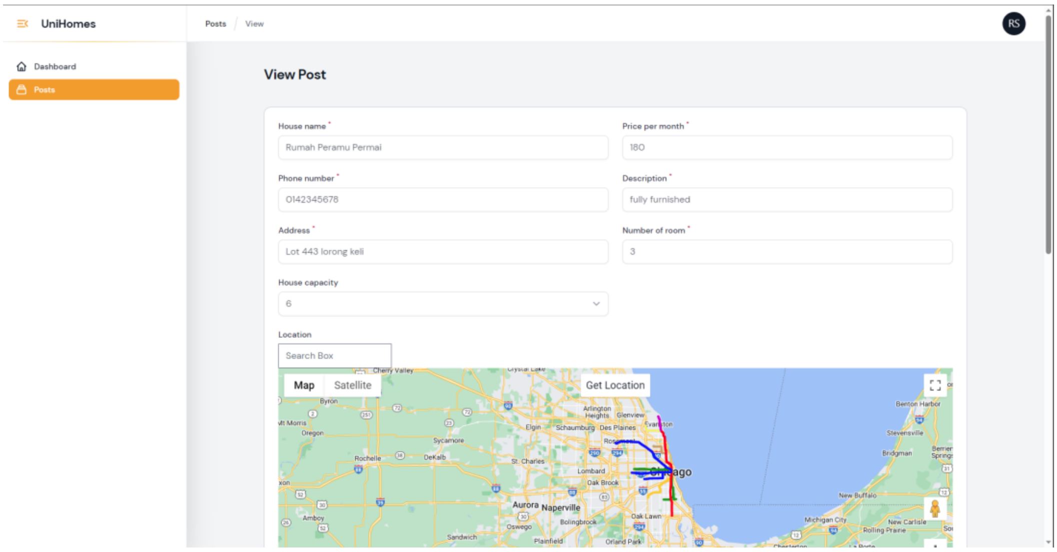
□ Always allow api.whatsapp.com to open links of this type in the associated app

Chat on WhatsApp with 0199458479

Don't have WhatsApp yet?
Download

Figure 4.12 Whatsapp the landlord

The student can click on the “message” button to contact the landlord. The button will automatically open up the whatsapp to let the student send message to ask any inquiries or book time to check out the house to rent.



View Post

House name *: Rumah Peramu Permai

Price per month *: 180

Phone number *: 0142345678

Description *: fully furnished

Address *: Lot 443 lorong keli

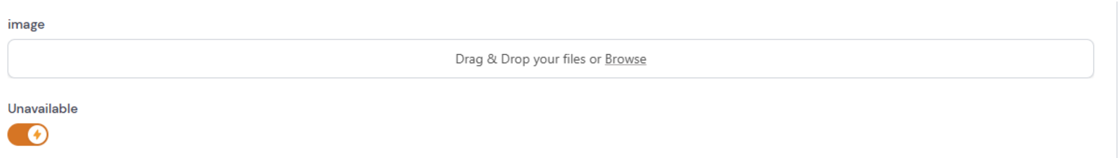
Number of room *: 3

House capacity: 6

Location: Search Box

Figure 4.13 Create Page

Figure 4.10 shows the create page where the landlord wants to create a new post to advertise their property to be rent. The need to fill in the information needed. They can pin the location of the house to allow the student to estimate the distance or see where the house is located.



image

Drag & Drop your files or Browse

Unavailable

Figure 4.14 Upload image and toggle post status

The landlord also can upload the images of their property and can toggle the status of the post either available or is unavailable.

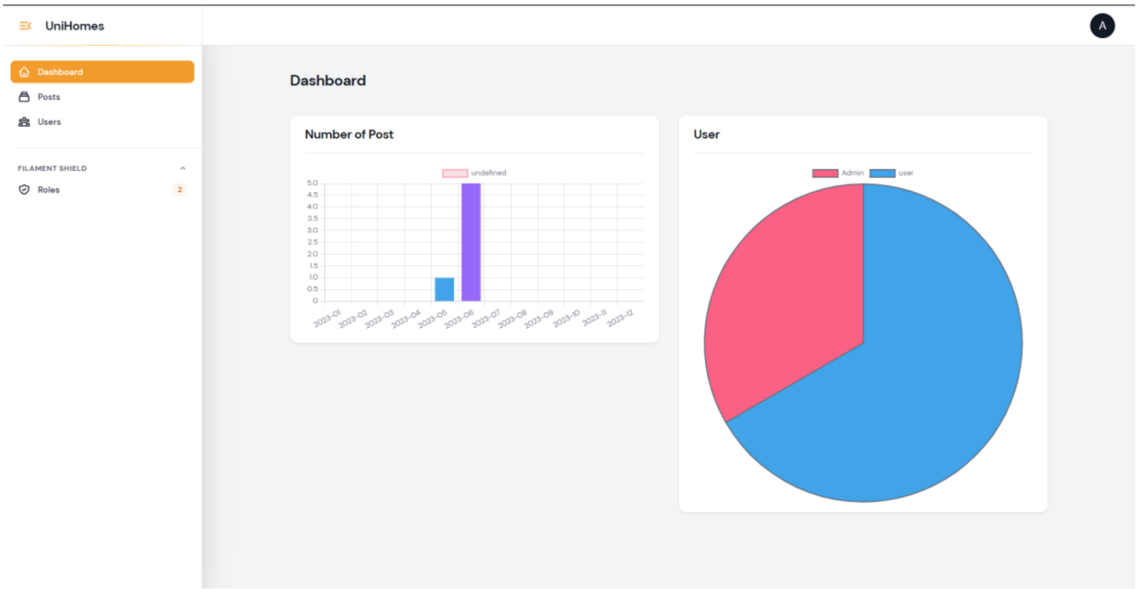
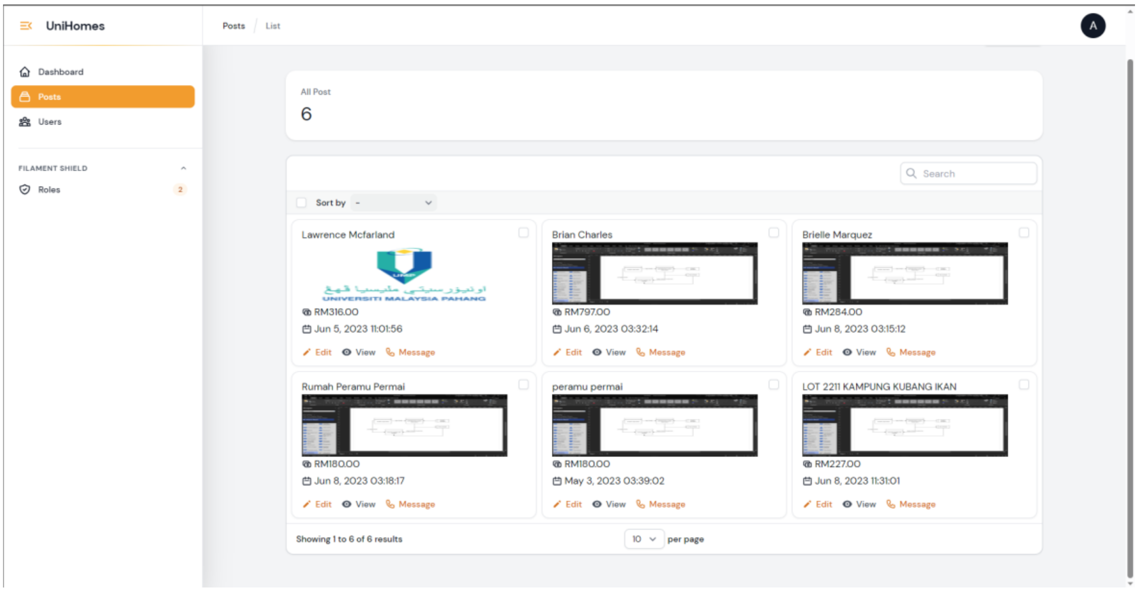


Figure 4.15 Landlord and Petakom Dashboard.

Figure 4.10 shows the dashboard of the landlord that will have list of the posts that they already submitted.



All Post
6

Lawrence Mcfarland
RM316.00
Jun 5, 2023 11:01:56

Brian Charles
RM797.00
Jun 6, 2023 03:32:14

Brielle Marquez
RM284.00
Jun 8, 2023 03:15:12

Rumah Peramu Permai
RM180.00
Jun 8, 2023 03:18:17

peramu permai
RM180.00
May 3, 2023 03:39:02

LOT 2211 KAMPUNG KUBANG IKAN
RM227.00
Jun 8, 2023 11:31:01

Showing 1 to 6 of 6 results

Figure 4.16 Post list

The landlord also can see the same view as the student. The landlord can view the posts that have been created by them before.

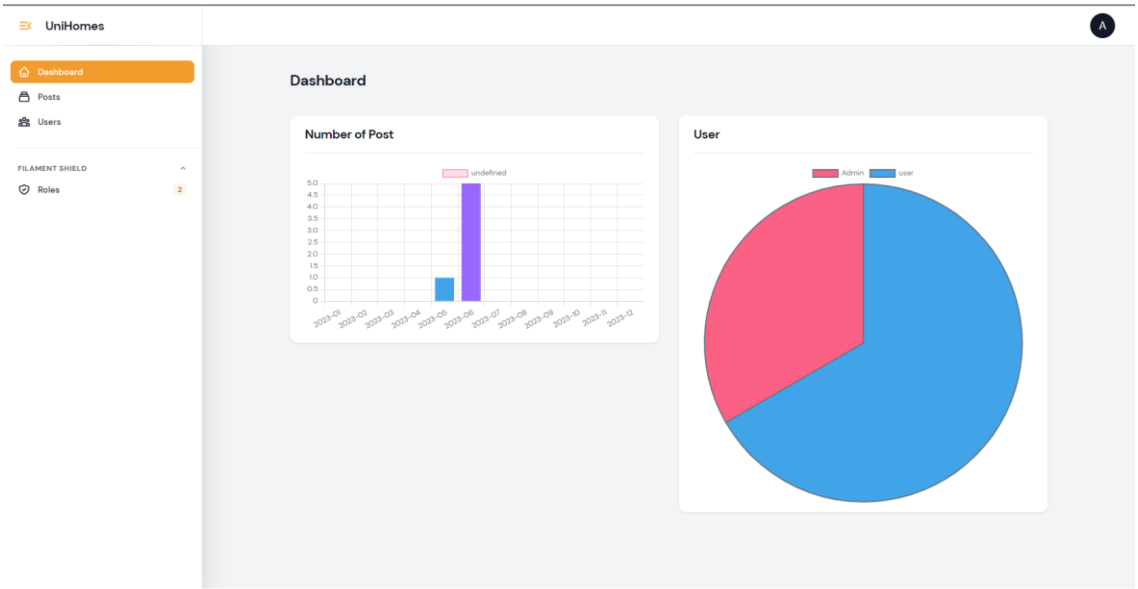
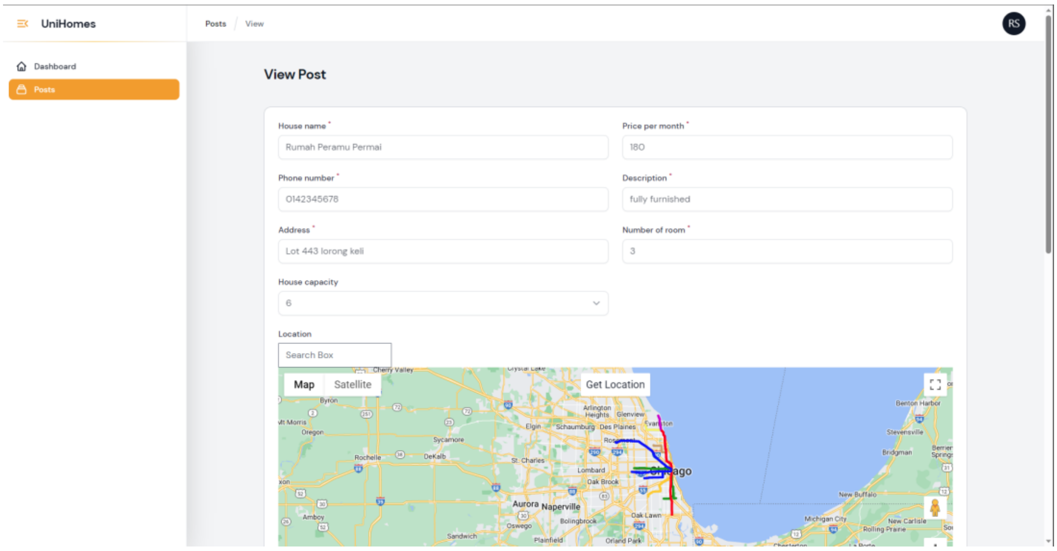


Figure 4.17 Report Interfaces

Figure 4.12 shows the report interface where the landlord can view the data of how many users viewed their post in graphical such as pie chart and bar graph.



View Post

House name * Rumah Peramu Permai

Price per month * 180

Phone number * 0142345678

Description * fully furnished

Address * Lot 443 lorong keli

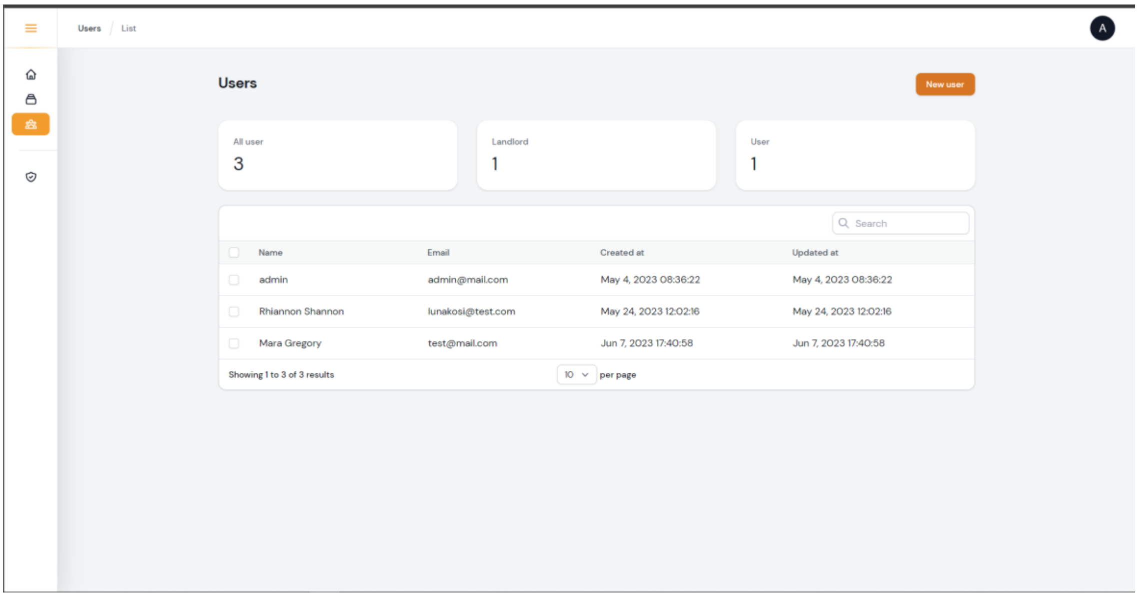
Number of room * 3

House capacity 6

Location Search Box

Figure 4.18 View Post

This figure shows the View page of the post that have been created by the Landlord. They can delete and edit the post if they want.



Users

All user 3

Landlord 1

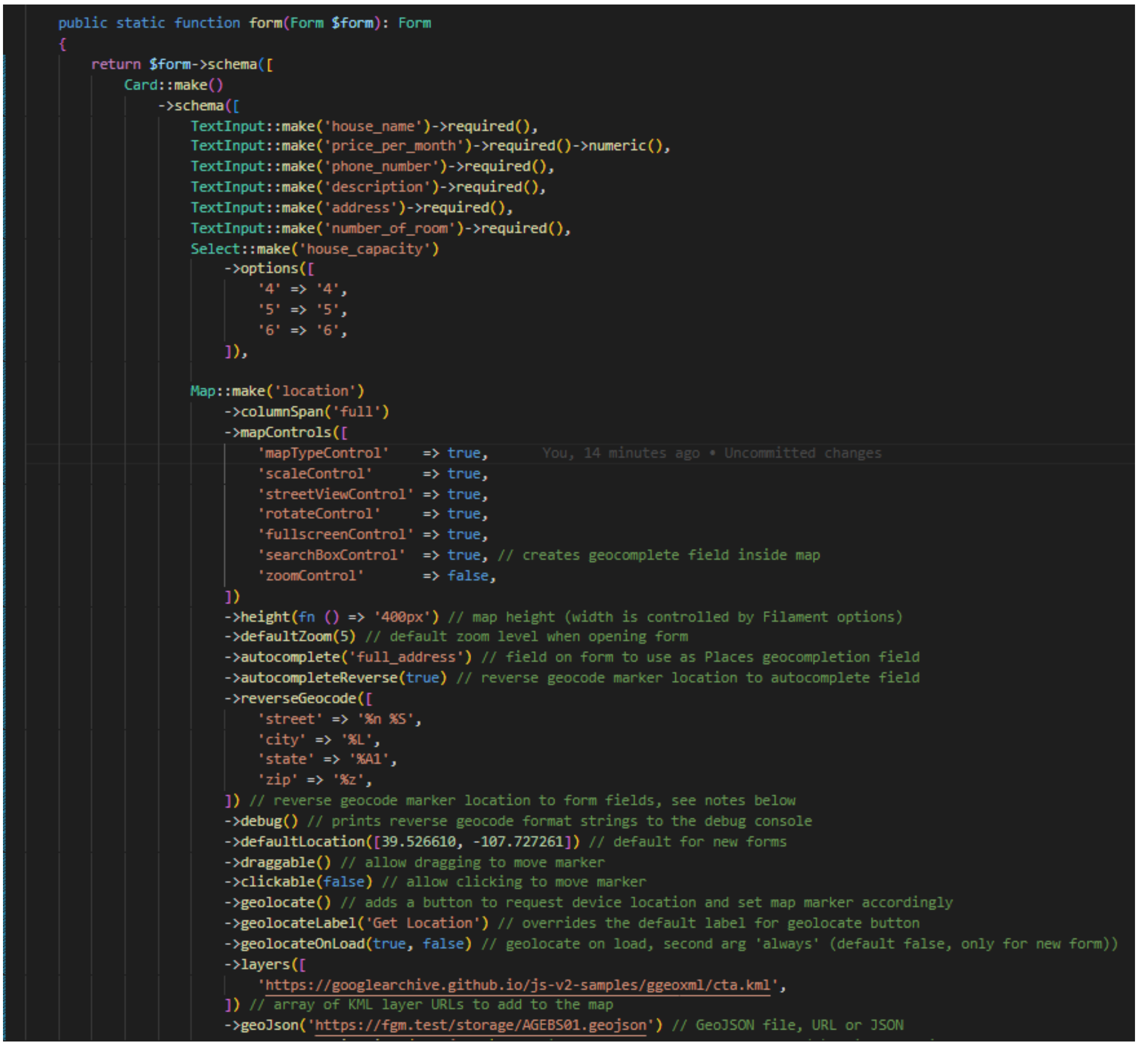
User 1

□	Name	Email	Created at	Updated at
□	admin	admin@mail.com	May 4, 2023 08:36:22	May 4, 2023 08:36:22
□	Rhiannon Shannon	lunakosi@test.com	May 24, 2023 12:02:16	May 24, 2023 12:02:16
□	Mara Gregory	test@mail.com	Jun 7, 2023 17:40:58	Jun 7, 2023 17:40:58

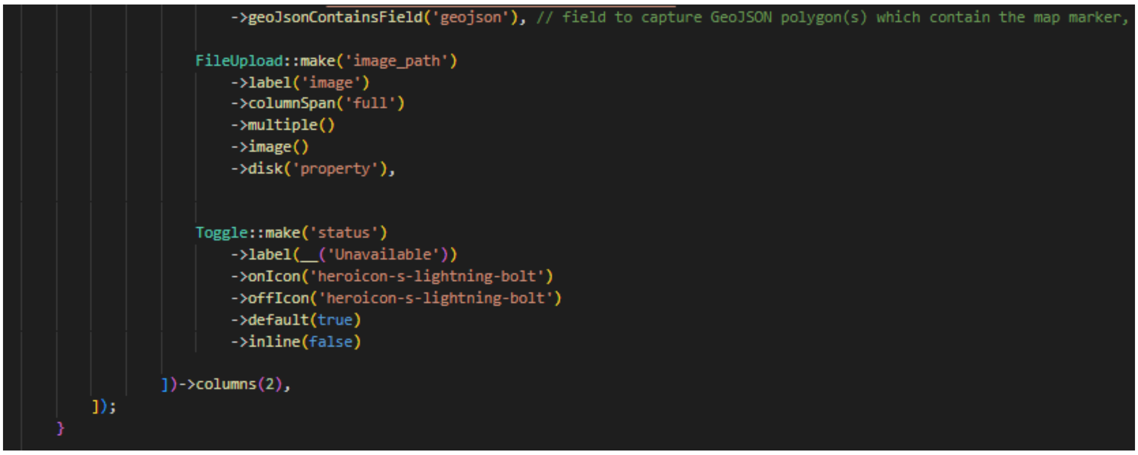
Showing 1 to 3 of 3 results 10 per page

Figure 4.19 Manage user interface

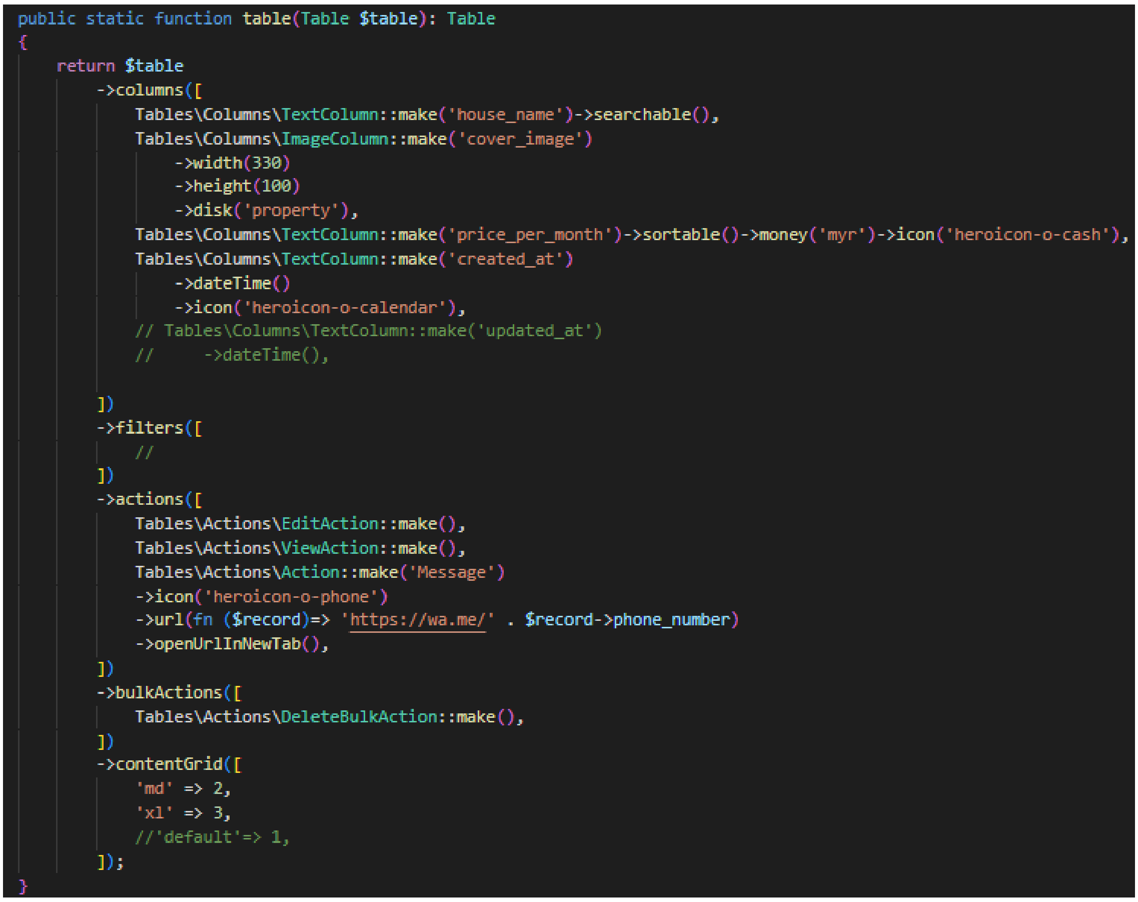
This interface allows the Petakom to see the list of user that is register in the system. the Petakom can delete, view and update any of the user and also disable some of the function for the user in roles.



```php
public static function form(Form $form): Form
{
    return $form->schema([
        Card::make()
            ->schema([
                TextInput::make('house_name')->required(),
                TextInput::make('price_per_month')->required()->numeric(),
                TextInput::make('phone_number')->required(),
                TextInput::make('description')->required(),
                TextInput::make('address')->required(),
                TextInput::make('number_of_room')->required(),
                Select::make('house_capacity')
                    ->options([
                        '4' => '4',
                        '5' => '5',
                        '6' => '6',
                    ]),

                Map::make('location')
                    ->columnSpan('full')
                    ->mapControls([
                        'mapTypeControl'    => true,        You, 14 minutes ago • Uncommitted changes
                        'scaleControl'      => true,
                        'streetViewControl' => true,
                        'rotateControl'     => true,
                        'fullscreenControl' => true,
                        'searchBoxControl'  => true, // creates geocomplete field inside map
                        'zoomControl'       => false,
                    ])
                    ->height(fn () => '400px') // map height (width is controlled by Filament options)
                    ->defaultZoom(5) // default zoom level when opening form
                    ->autocomplete('full_address') // field on form to use as Places geocompletion field
                    ->autocompleteReverse(true) // reverse geocode marker location to autocomplete field
                    ->reverseGeocode([
                        'street' => '%n %S',
                        'city' => '%L',
                        'state' => '%A1',
                        'zip' => '%z',
                    ]) // reverse geocode marker location to form fields, see notes below
                    ->debug() // prints reverse geocode format strings to the debug console
                    ->defaultLocation([39.526610, -107.727261]) // default for new forms
                    ->draggable() // allow dragging to move marker
                    ->clickable(false) // allow clicking to move marker
                    ->geolocate() // adds a button to request device location and set map marker accordingly
                    ->geolocateLabel('Get Location') // overrides the default label for geolocate button
                    ->geolocateOnLoad(true, false) // geolocate on load, second arg 'always' (default false, only for new form))
                    ->layers([
                        'https://googlearchive.github.io/js-v2-samples/ggeoxml/cta.kml',
                    ]) // array of KML layer URLs to add to the map
                    ->geoJson('https://fgm.test/storage/AGEBS01.geojson') // GeoJSON file, URL or JSON
```

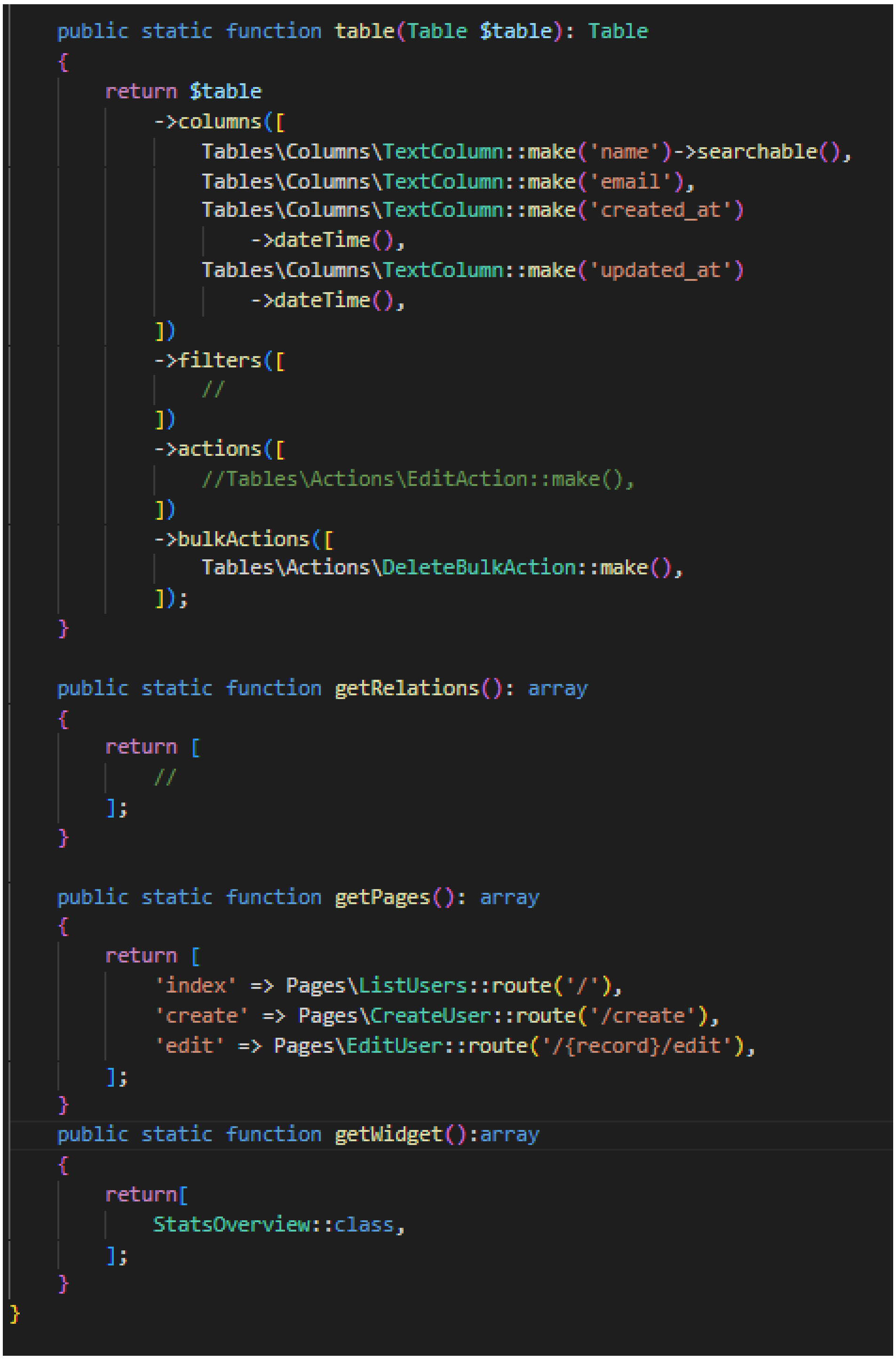


```php
                    ->geoJsonContainsField('geojson'), // field to capture GeoJSON polygon(s) which contain the map marker,

                FileUpload::make('image_path')
                    ->label('image')
                    ->columnSpan('full')
                    ->multiple()
                    ->image()
                    ->disk('property'),


                Toggle::make('status')
                    ->label(__('Unavailable'))
                    ->onIcon('heroicon-s-lightning-bolt')
                    ->offIcon('heroicon-s-lightning-bolt')
                    ->default(true)
                    ->inline(false)

            ])->columns(2),
    ]);
}
```



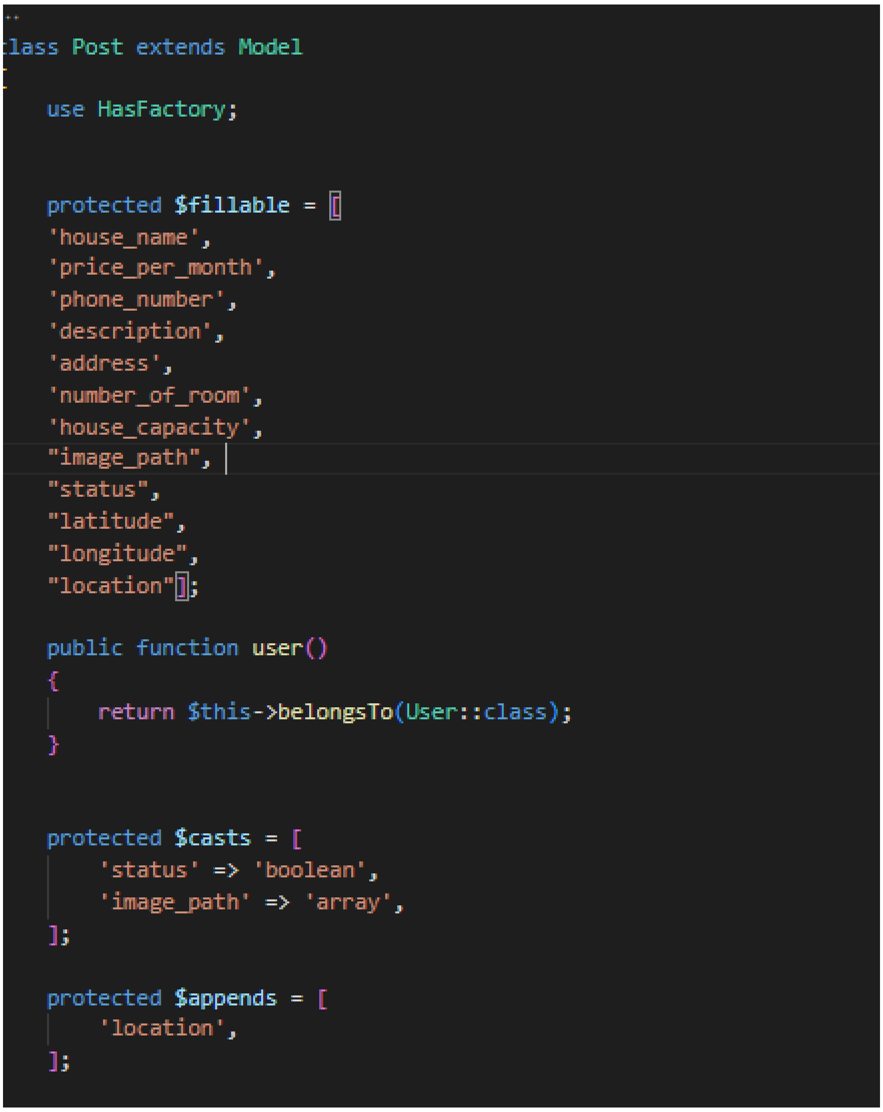
```php
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('house_name')->searchable(),
            Tables\Columns\ImageColumn::make('cover_image')
                ->width(330)
                ->height(100)
                ->disk('property'),
            Tables\Columns\TextColumn::make('price_per_month')->sortable()->money('myr')->icon('heroicon-o-cash'),
            Tables\Columns\TextColumn::make('created_at')
                ->dateTime()
                ->icon('heroicon-o-calendar'),
            // Tables\Columns\TextColumn::make('updated_at')
            //     ->dateTime(),

        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
            Tables\Actions\ViewAction::make(),
            Tables\Actions\Action::make('Message')
            ->icon('heroicon-o-phone')
            ->url(fn ($record)=> 'https://wa.me/' . $record->phone_number)
            ->openUrlInNewTab(),
        ])
        ->bulkActions([
            Tables\Actions\DeleteBulkAction::make(),
        ])
        ->contentGrid([
            'md' => 2,
            'xl' => 3,
            //'default'=> 1,
        ]);
}
```

Figure 4.20 PostResource.php



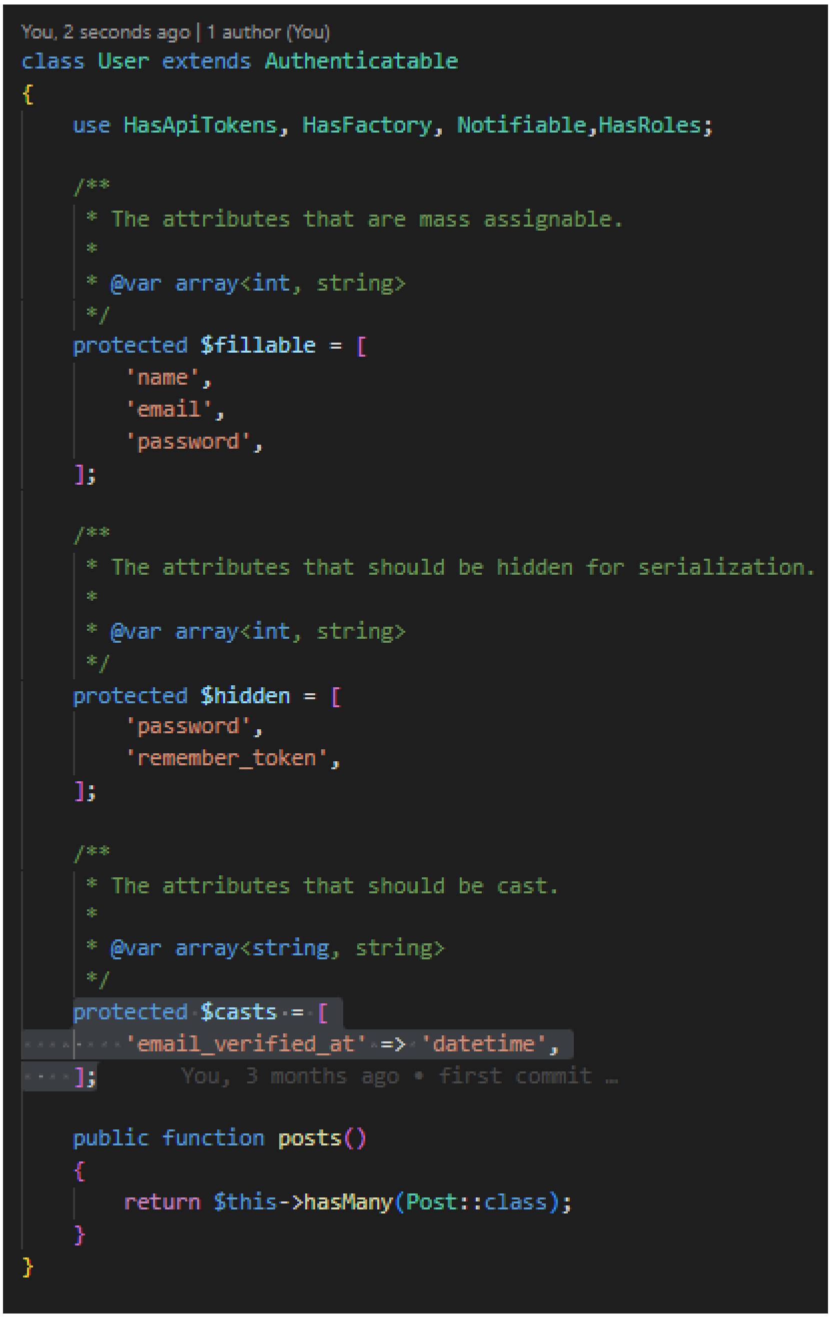
```
    public static function table(Table $table): Table
    {
        return $table
            ->columns([
                Tables\Columns\TextColumn::make('name')->searchable(),
                Tables\Columns\TextColumn::make('email'),
                Tables\Columns\TextColumn::make('created_at')
                    ->dateTime(),
                Tables\Columns\TextColumn::make('updated_at')
                    ->dateTime(),
            ])
            ->filters([
                //
            ])
            ->actions([
                //Tables\Actions\EditAction::make(),
            ])
            ->bulkActions([
                Tables\Actions\DeleteBulkAction::make(),
            ]);
    }

    public static function getRelations(): array
    {
        return [
            //
        ];
    }

    public static function getPages(): array
    {
        return [
            'index' => Pages\ListUsers::route('/'),
            'create' => Pages\CreateUser::route('/create'),
            'edit' => Pages\EditUser::route('/{record}/edit'),
        ];
    }
    public static function getWidget():array
    {
        return[
            StatsOverview::class,
        ];
    }
}
```

Figure 4.21 UserResource.php



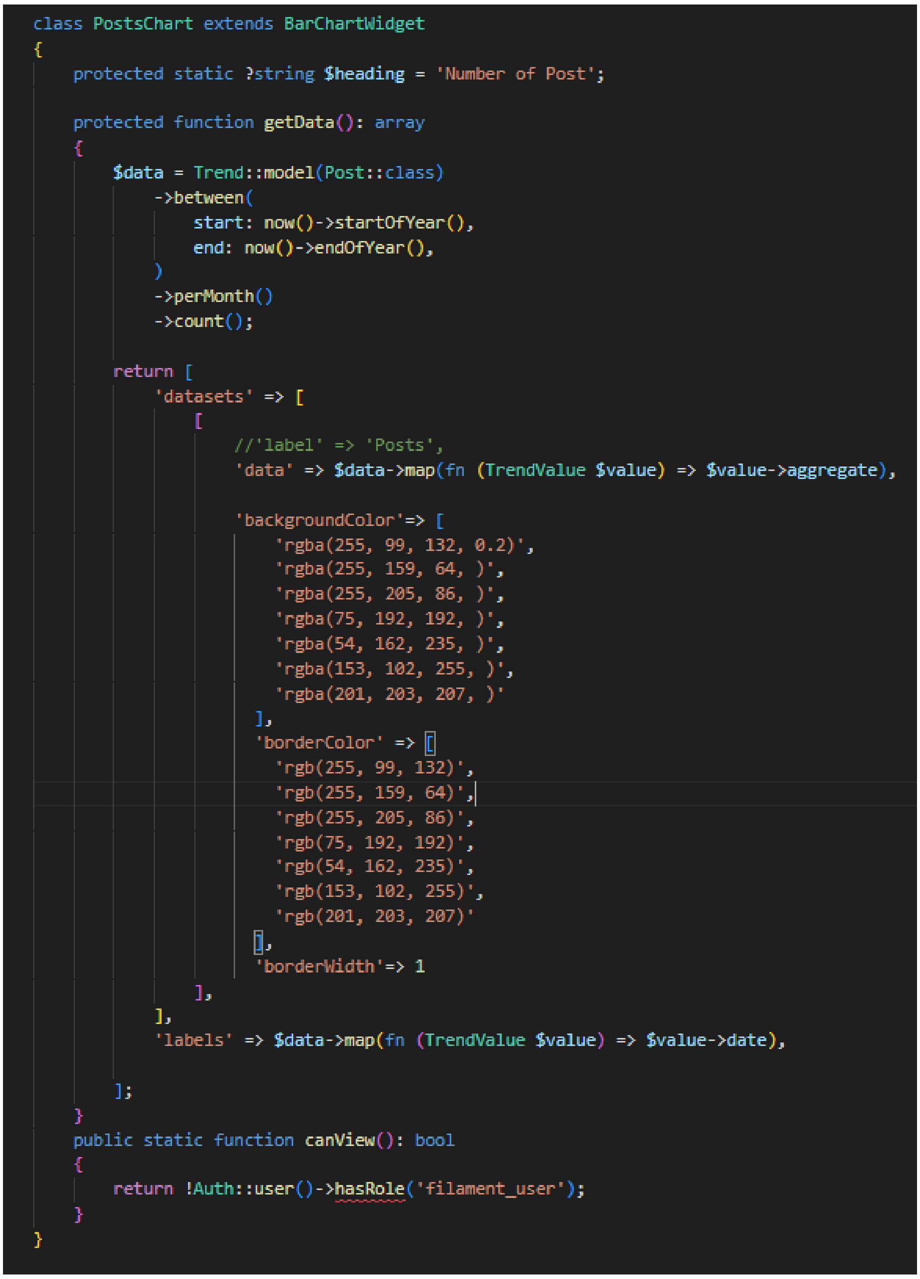
```
class Post extends Model
{
    use HasFactory;


    protected $fillable = [
    'house_name',
    'price_per_month',
    'phone_number',
    'description',
    'address',
    'number_of_room',
    'house_capacity',
    "image_path",
    "status",
    "latitude",
    "longitude",
    "location"];

    public function user()
    {
        return $this->belongsTo(User::class);
    }


    protected $casts = [
        'status' => 'boolean',
        'image_path' => 'array',
    ];

    protected $appends = [
        'location',
    ];
```

Figure 4.22 .post.php



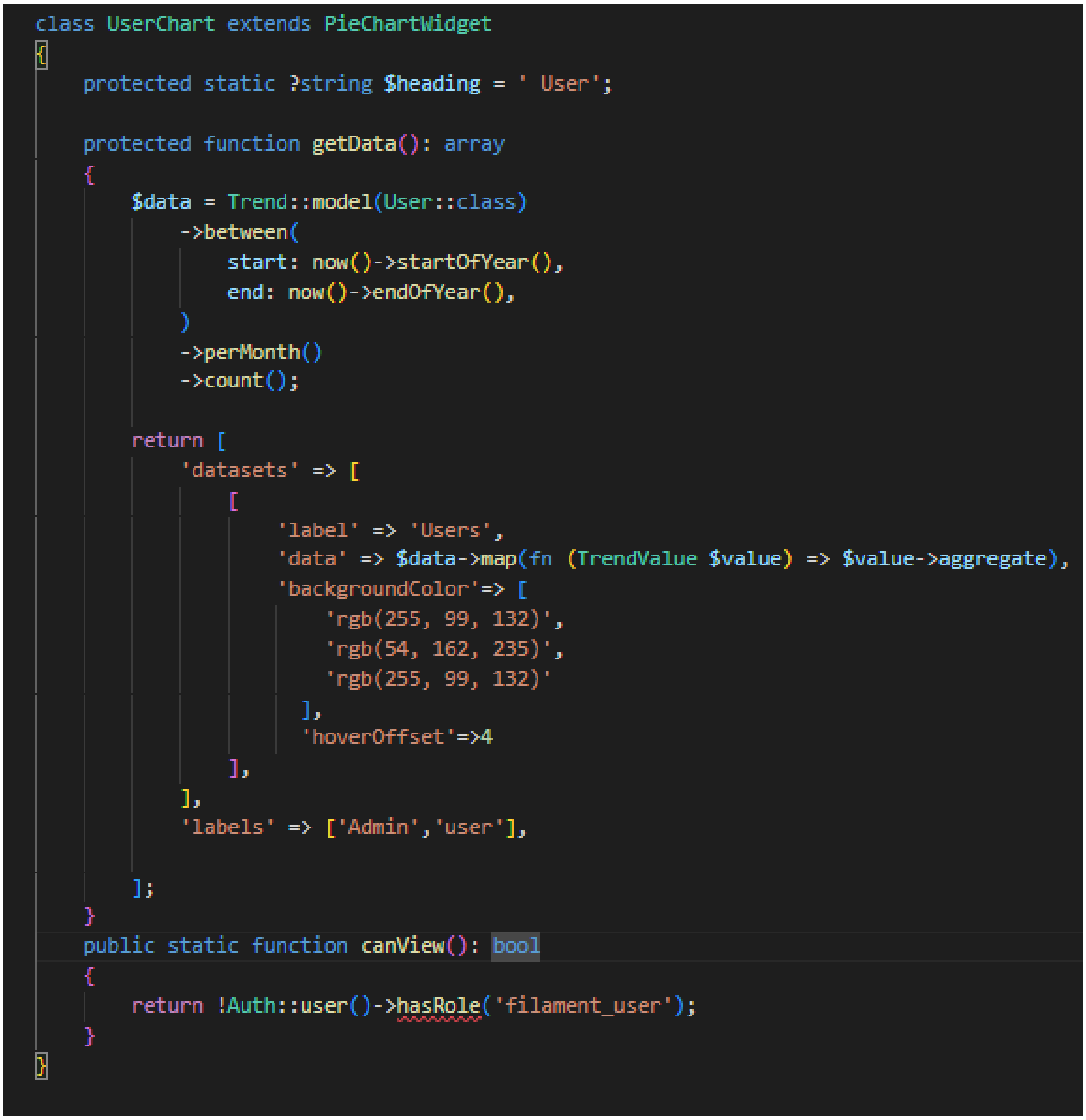
```
You, 2 seconds ago | 1 author (You)
class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable,HasRoles;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];       You, 3 months ago • first commit …

    public function posts()
    {
        return $this->hasMany(Post::class);
    }
}
```

Figure 4.23 user.php



```php
class PostsChart extends BarChartWidget
{
    protected static ?string $heading = 'Number of Post';

    protected function getData(): array
    {
        $data = Trend::model(Post::class)
            ->between(
                start: now()->startOfYear(),
                end: now()->endOfYear(),
            )
            ->perMonth()
            ->count();

        return [
            'datasets' => [
                [
                    //'label' => 'Posts',
                    'data' => $data->map(fn (TrendValue $value) => $value->aggregate),

                    'backgroundColor'=> [
                        'rgba(255, 99, 132, 0.2)',
                        'rgba(255, 159, 64, )',
                        'rgba(255, 205, 86, )',
                        'rgba(75, 192, 192, )',
                        'rgba(54, 162, 235, )',
                        'rgba(153, 102, 255, )',
                        'rgba(201, 203, 207, )'
                      ],
                      'borderColor' => [
                        'rgb(255, 99, 132)',
                        'rgb(255, 159, 64)',
                        'rgb(255, 205, 86)',
                        'rgb(75, 192, 192)',
                        'rgb(54, 162, 235)',
                        'rgb(153, 102, 255)',
                        'rgb(201, 203, 207)'
                      ],
                      'borderWidth'=> 1
                ],
            ],
            'labels' => $data->map(fn (TrendValue $value) => $value->date),

        ];
    }
    public static function canView(): bool
    {
        return !Auth::user()->hasRole('filament_user');
    }
}
```

Figure 4.24 BarChart widget



```php
class UserChart extends PieChartWidget
{
    protected static ?string $heading = ' User';

    protected function getData(): array
    {
        $data = Trend::model(User::class)
            ->between(
                start: now()->startOfYear(),
                end: now()->endOfYear(),
            )
            ->perMonth()
            ->count();

        return [
            'datasets' => [
                [
                    'label' => 'Users',
                    'data' => $data->map(fn (TrendValue $value) => $value->aggregate),
                    'backgroundColor'=> [
                        'rgb(255, 99, 132)',
                        'rgb(54, 162, 235)',
                        'rgb(255, 99, 132)'
                      ],
                      'hoverOffset'=>4
                ],
            ],
            'labels' => ['Admin','user'],

        ];
    }
    public static function canView(): bool
    {
        return !Auth::user()->hasRole('filament_user');
    }
}
```

Figure 4.25 Pie Chart widget

4.3 Testing and Result Discussion

After the development of UniHomes is completed, testing process is carried out to evaluate the functionality and usability of the web-based system. User Acceptance test (UAT) is done that involved students from Universiti Malaysia Pahang. UAT let the user evaluate the available features in the web-based system and help in finding bugs or errors to be improve.

The objective to develop a prototype system for student house management system is achieve as show in chapter 4. The system is able to be use by the students, landlords and PETAKOM. The requirement for the system is well studied and are translated to a working prototype that is evaluate by the users. Lastly, to validate the proposed prototype system user acceptance test has been done to collect the feedback from the users.

From the User Acceptance Test done, the result shows that 10 out of 10 testers are satisfied with the system. The system has received a 100% satisfaction rate from all 10 testers. During the test, all functions of the system passed successfully, demonstrating that it meets the specified requirements and fulfils the needs of the users. The positive feedback and successful test results lead to the conclusion that the system has achieved its objectives and is ready for deployment to the end-users. The high satisfaction rate and flawless performance during testing indicate that the software development process using the Waterfall model has been effective in delivering a reliable and robust system.

CHAPTER 5

CONCLUSION

5.1 Introduction

Chapter 5 contain the summarization of developing the UniHomes, a web-based system to help the student finds their house and room in order to achieve the objective and overcome the problem that have been stated in the problem statement in chapter 1. This web-based system can be used to become one stop centre for students to find their accommodations. The system is developed using Laravel Framework, Figma and Visual Studio code. The methodology that is used is Waterfall since it is highly structured and organized, making it easier to plan and manage projects effectively.

5.2 Limitation and Constraint

i. Limitation of Time

The limited time for developing the project may result in some function or features are not able to be implemented properly.

ii. Limitation of resources

As the project does not have enough resources to be delegate. It takes more time to complete a feature or function.

iii. Limitation of knowledge

Limited knowledge on developing the project result in more time taken to complete a features as research and study need to be done first.

5.3 Future Work

There are few enhancements that can be applied for future improvement of the UniHomes.

i. Develop the UniHomes with mobile application for IOS and Android platform

ii. Collaborate with any organization to make sure that the system is known and trusted such as Ministry of High Education.

iii. Expand the user scope not only limited UMP student but to other IPTA and IPTS students.

REFERENCES

4 Types of Software Maintenance & What is Software Maintenance? (n.d.). Retrieved November 23, 2022, from https://cpl.thalesgroup.com/software-monetization/four-types-of-software-maintenance

Asia's largest online property portal group | PropertyGuru Group. (n.d.). Retrieved February 12, 2023, from https://www.propertygurugroup.com/

Landlord Reveals Worst Student Housing Problems, And Why It's A Never-Ending Nightmare | TRP. (n.d.). Retrieved October 27, 2022, from https://www.therakyatpost.com/news/malaysia/2022/09/21/landlord-reveals-worst-student-housing-problems-and-why-its-a-never-ending-nightmare/

Malaysia's Largest Marketplace - Buy & Sell Your New and Preloved Items - Mudah.my. (n.d.). Retrieved February 12, 2023, from https://www.mudah.my/

Rumah Siswa. (n.d.). Retrieved February 12, 2023, from https://rumahsiswa.mohe.gov.my/about-us

"Scam" rumah sewa, penuntut IPT jadi sasaran. (n.d.). Retrieved October 27, 2022, from https://www.bharian.com.my/berita/kes/2022/07/982122/scam-rumah-sewa-penuntut-ipt-jadi-sasaran

SDLC - Waterfall Model. (n.d.). Retrieved November 13, 2022, from https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

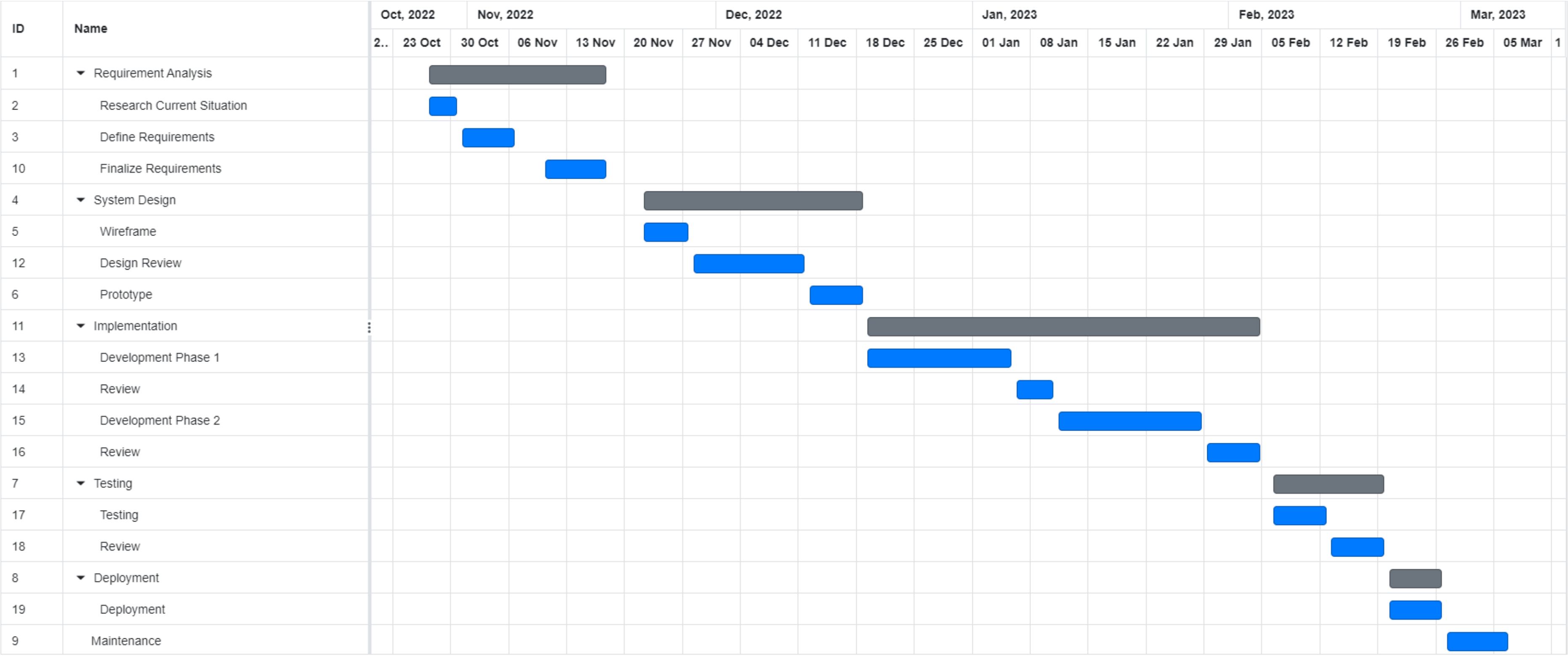
Seven out of 10 landlords won't rent out their property to students | The Independent | The Independent. (n.d.). Retrieved February 12, 2023, from https://www.independent.co.uk/news/business/news/student-housing-properties-landlords-wont-rent-out-a8100056.html

What is User Acceptance Testing (UAT)? - Definition from Techopedia. (n.d.). Retrieved February 12, 2023, from https://www.techopedia.com/definition/3887/user-acceptance-testing-uat-software-testing

What is WaterFall Model in Software Developement Life Cycle | SDLC. (n.d.). Retrieved November 23, 2022, from https://www.toolsqa.com/software-testing/waterfall-model/

APPENDICES

APPENDIX A
GANTT CHART



ID	Name
1	Requirement Analysis
2	Research Current Situation
3	Define Requirements
10	Finalize Requirements
4	System Design
5	Wireframe
12	Design Review
6	Prototype
11	Implementation
13	Development Phase 1
14	Review
15	Development Phase 2
16	Review
7	Testing
17	Testing
18	Review
8	Deployment
19	Deployment
9	Maintenance

Figure 3.22 Gantt Chart

APPENDIX B
SRS DOCUMENT

Version

1

UNIHOMES

Faculty of Computing

Software Requirement Specification (SRS)

Table of Contents

1.0 PRODUCT DESCRIPTION

1.1 User Characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product, including educational level, experience, and technical expertise.

Table 1: User Characteristics

User	Education level	Background experience
Student	• Diploma and above • Possess basic knowledge in Bahasa Malaysia and English • IT literate	• Age level (18 – 45) • Suitable for all user that have experience in using web application.
PETAKOM	• Diploma and above • Possess basic knowledge in Bahasa Malaysia and English • IT literate	• Age level (18 – 45) • Suitable for all user that have experience in using web application.
Landlord	• Possess basic knowledge in Bahasa Malaysia and English	• Have basic knowledge using computer • Local citizen

1.2 Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer’s options. These include

- Landlord can only have one account that been approved by PETAKOM.
- The system might take 0.5 seconds to save the user information after the user log out.
- The system must use sufficient security to avoid sensitive data being leak.
- The system must be use with stable internet connection to avoid the system fail to load
- The system should be available for the users 24 hours a day.
- The system would reject any registration that contains false information based on Communication and Multimedia Act 709 under section 11 (Diraja, 2010).
- The system allows the users to update their registration information based on Communication and Multimedia Act 709 under section 12 (Diraja, 2010).
- UniHomes only can access the user location when the user allows it.
- The system should not contain any sensitive words, symbols or picture because of the diverse races.
- The system should be able to be use in all browsers such as Google Chrome, Mozilla Firefox, Safari, Opera, Microsoft Edge.

2.0 INTERFACE REQUIREMENTS

This should be a detailed description of all inputs into and outputs from the software system.

2.1 User Interface

Describe the logical characteristics of each interface between the software product and the users. Define the softw are components for which a user interface is needed.

INTERFACE NAME	DESCRIPTION	
Landlord		
Register	Landlord can register their account by entering their information	
Login	Landlord can login to the system by entering their username and password	
View profile	The Landlord can see their profile details	
Edit profile	The Landlord can edit their profile details	
Add house	Landlord can add their house to be rent to the system	
Edit post	Landlord can edit the post details	
Delete post	Landlord can delete their post	
Generate report	The Landlord can view the analytic in graphic	
Student		
Register	student can register their account by entering their information	
Login	student can login to the system by entering their username and password	
View profile	The student can see their profile details	
Edit profile	The student can edit their profile details	
Search	Student can search for house in the interface	
PETAKOM		
Register	PETAKOM can register	

	their account by entering their information	
Login	PETAKOM can login to the system by entering their username and password	
View profile	PETAKOM can see their profile details	
Edit profile	PETAKOM can edit their profile details	
View review	PETAKOM can view the review write by the student	
Delete review	PETAKOM can delete the review	
View post	PETAKOM can view post make by Landlord	

2.2 Hardware Interface

Not Applicable

2.3 Software Interface

The system is implemented in windows, Linux or MacO

3.0 SOFTWARE PRODUCT FEATURES

3.1 Register and Login

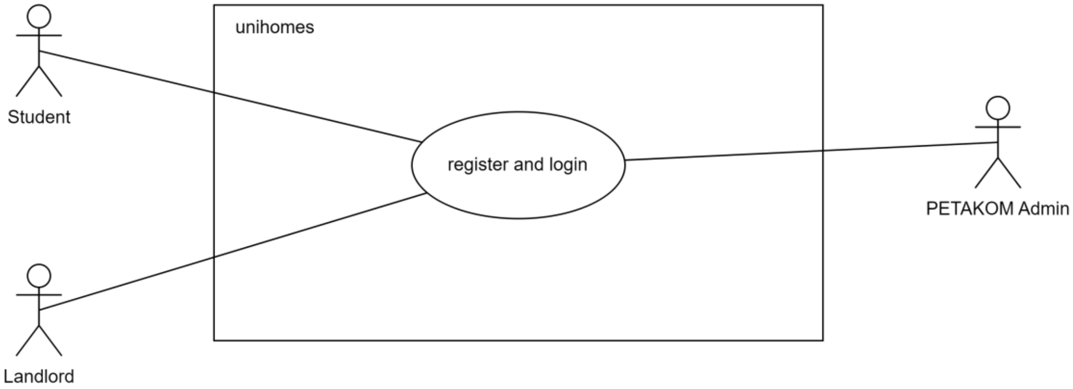


Figure 1: Register and login

Table 2: Use Case Name

Use Case ID	UH-UCD-01
Brief Description	This use case allows the users to register and login to the system
Actor	Student, landlord, PETAKOM
Preconditions	
Basic Flow	Student and landlord 1. The use case starts when the user at the login page. 2. The system the login interface. 3. The user can • [A1] Login to the system. • [A2] Register. 4. The use case end.
Alternative Flow	[A1] login to the system 1. The user enters their username. 2. The system validate the username 3. The user enters their password. 4. The system validate the password 5. The user clicks <<login>> button to log in. [E1: invalid

	username and password]. 6. The system shows the main page. 7. The use case continues. [A2] Register 1. The user clicks <<Register>> button. 2. The system shows the registration form. 3. The user fills in the information needed. 4. The system validate the information 5. The user clicks the <<submit>> button 6. The system create the user 7. Use case continues .
Exception Flow	E1: invalid username and password 1. The users login with invalid username or password. 2. The error message showed. 3. Continue with step 1.
Post-Conditions	The user able to login to the system.
Rules	Must have an account to register.
Constraints	Not applicable
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.2 Manage Student Profle

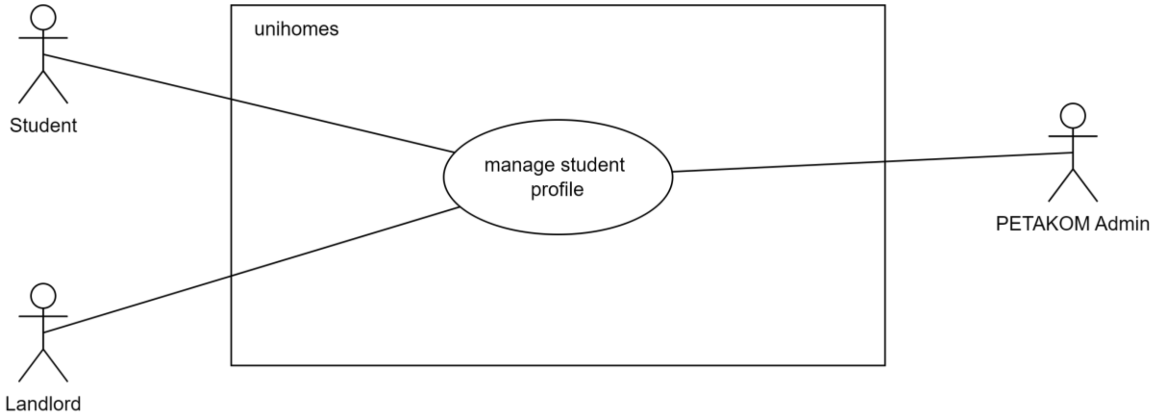


Figure 2: Manage student profile

Table 3: Manage student profile

Use Case ID	UH-UCD-02
Brief Description	This use case allows the student to view, update and delete the student profile. Landlord and PETAKOM can view the student profile.
Actor	Student, landlord, PETAKOM.
Preconditions	Must be log in to the system.
Basic Flow	Student 1. The use case starts when the student login into the system [E1-invalid username and password]. 2. Student click on the <<Profile >> button. 3. The student can: • [A1] view their profile • [A2] update their profile • [A3] delete their profile

	4. The use case end. Landlord and PETAKOM 1. The use case starts when the landlord and PETAKOM login to the system. 2. Landlord and PETAKOM click the student profile. 3. Use case end.
Alternative Flow	[A1] view their profile 1. Student click <<profile>> button. 2. System show the user profile. 3. Use case continues [A2] update their profile 1. Student click <<profile>> button. 2. Student view their profile. 3. Student change the information. 4. System process the change 5. Student click <<Save>> button. 6. System saved the change. 7. Use case continues [A3] delete their profile 1. Student click <<profile>> button. 2. Student click <<Delete>> button. 3. System process the request. 4. Use case continues
Exception Flow	E1-invalid username and password 1. The users login with invalid username or password. 2. The error message showed. 3. Continue with step 1.
Post-Conditions	Student able to view, update and delete their profile. Landlord and PETAKOM can view the student profile.
Rules	The student , landlord and PETAKOM must login to the system

Constraints	Not applicable
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.3 Manage Landlord Profile

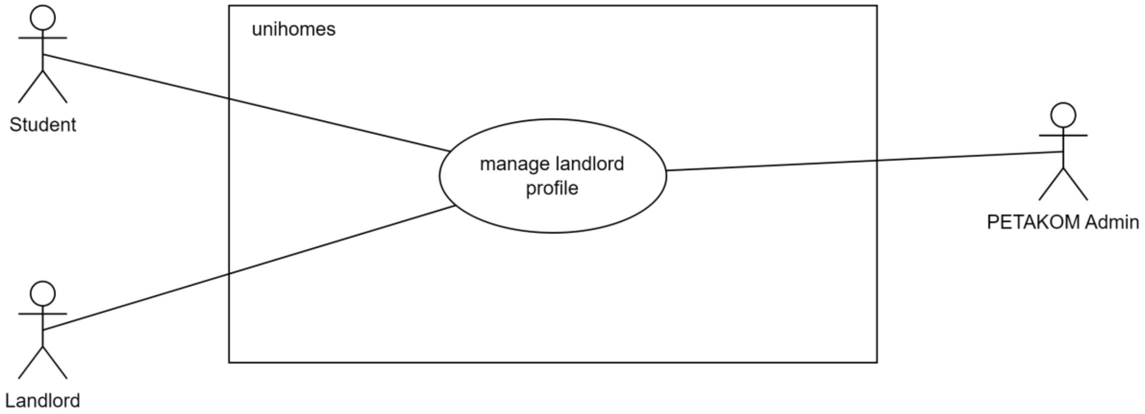


Figure 2: Manage landlord profile

Table 3: Manage landlord profile

Use Case ID	UH-UCD-03
Brief Description	This use case allows the landlord to view, update and delete the student profile. Student and PETAKOM can view the student profile.
Actor	Student, landlord, PETAKOM.
Preconditions	Must be log in to the system.
Basic Flow	Landlord 1. The use case starts when the landlord login into the system [E1-invalid username and password]. 2. landlord click on the <<Profile >> button. 3. The landlord can: • [A1] view their profile • [A2] update their profile • [A3] delete their profile 4. The use case end. Student and PETAKOM 1. The use case starts when the student and PETAKOM login to the system.

	2. Student and PETAKOM click the landlord profile. 3. Use case end.
Alternative Flow	[A1] view their profile 1. Landlord click <<profile>> button. 2. System show their profile. 3. Use case continues [A2] update their profile 1. Landlord click <<profile>> button. 2. Landlord view their profile. 3. Landlord change the information. 4. System process the change 5. Landlord click <<Save>> button. 6. System saved the change. 7. Use case continues [A3] delete their profile 1. Landlord click <<profile>> button. 2. Landlord click <<Delete>> button. 3. System process the request. 4. Use case continues
Exception Flow	E1-invalid username and password 1. The users login with invalid username or password. 2. The error message showed. Continue with step 1.
Post-Conditions	Landlord able to view, update and delete their profile. Student and PETAKOM can view the student profile.
Rules	The student, landlord and PETAKOM must login to the system
Constraints	
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.4 Manage House Detail

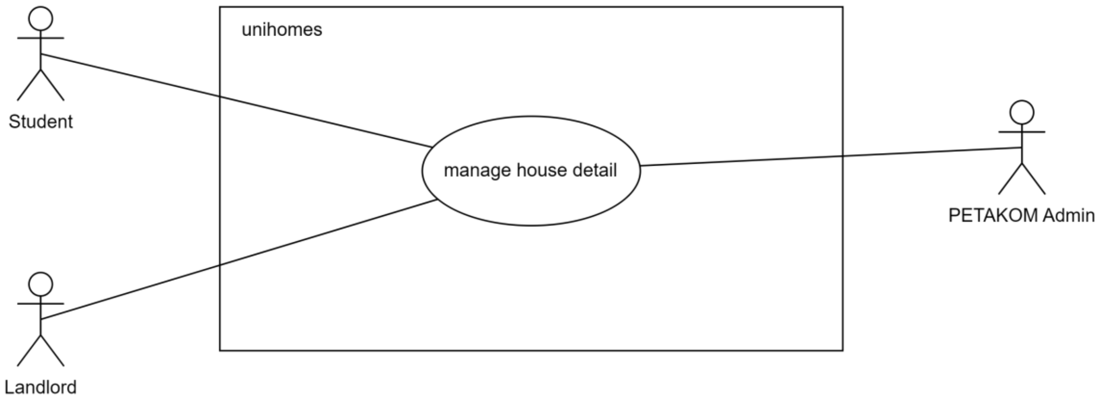


Figure 2: manage house details

Table 3: Manage house details

Use Case ID	UH-UCD-04
Brief Description	This use case allows the landlord to view, create, update and delete their house details. Student and PETAKOM able to view the house details
Actor	Student, landlord, PETAKOM.
Preconditions	The user must login to the system.
Basic Flow	Landlord 1. The use case starts when landlord login to the system 2. The landlord clicks <<post?>> button. 3. the landlord can: • [A1] add house details. • [A2] view house details. • [A3] update house details. • [A4] delete house details. 4. Use case end. Student and PETAKOM 1. The use case starts when the student and PETAKOM login to the system 2. Student and PETAKOM click landlord post

	3. The use case end.
Alternative Flow	[A1] add house details. 1. Landlord click <<add>> button. 2. System shows the form 3. Landlord fill in the house information. 4. System store the information 5. Landlord click save. 6. Post created in the system 7. Use case continues. [A2] view house details. 1. Landlord click <<Post>> button. 2. System shows list of post 3. Choose which post to view 4. System show the details. 5. Use case continues [A3] update house details. 1. Landlord click <<Post>> button. 2. System shows list of post 3. Choose which house to update 4. System shows the infor 5. Landlord change the details. 6. Landlord click <<Save>> button. 7. System saved the change 8. Use case continues [A4] delete house details. 1. Landlord click <<Post>> button. 2. System shows list of post 3. Choose which house to delete 4. Landlord click <<Delete>> button. 5. System delete the post.

	6. Use case continues
Exception Flow	E1-invalid username and password 1. The users login with invalid username or password. 2. The error message showed. 3. Continue with step 1.
Post-Conditions	The landlord can view, create, update and delete house information. The student and PETAKOM can view the house details.
Rules	All user must login to the system
Constraints	
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.5 Search House

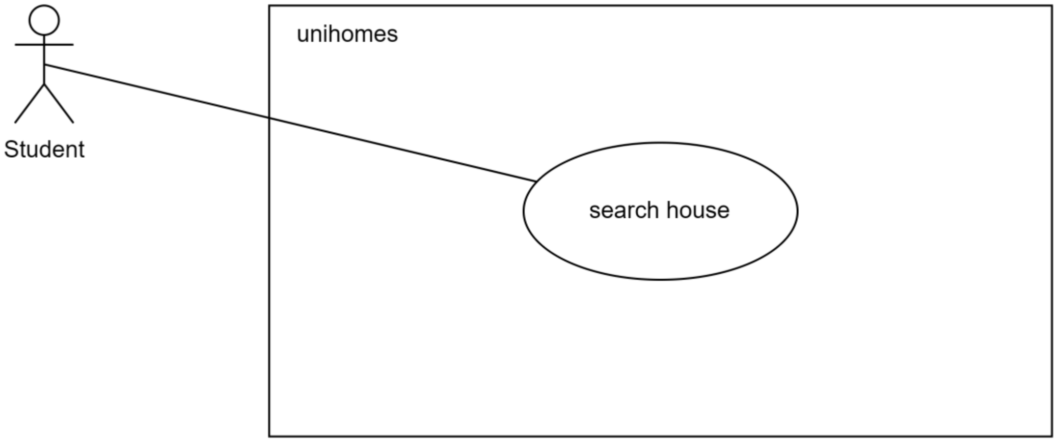


Figure 2: Search house

Table 3: search house

Use Case ID	UH-UCD-05
Brief Description	This use case allows the student to search for house or room to rent.
Actor	Student
Preconditions	Must have an account.
Basic Flow	1. The use case starts when the student login into the system [E1-invalid username and password]. 2. The system show homepage 3. Student fill in keyword in the search fields [E2: no result found] 4. The system shows the result 5. Student choose which house to contact 6. Use case end.
Alternative Flow	
Exception Flow	E1-invalid username and password 1. The users login with invalid username or password. 2. The error message showed.

	3. Continue with step 1. E2- no result found 1. There is no result found. 2. System show no result. 3. Continue with step 1.
Post-Conditions	Student can make search in the system
Rules	Must login to the system.
Constraints	
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.6 Manage Review

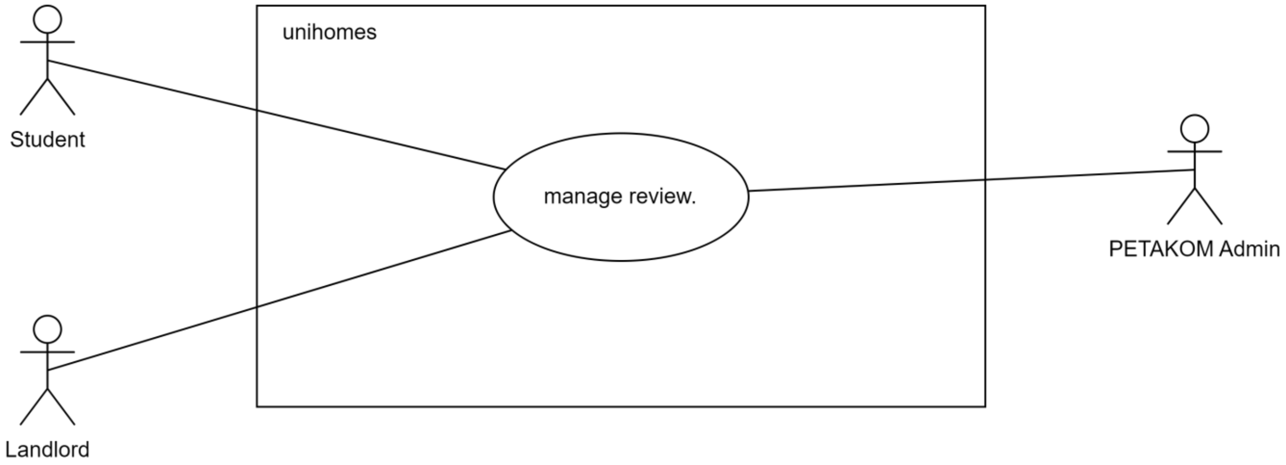


Figure 2: Manage review

Table 3: Manage review

Use Case ID	UH-UCD-06
Brief Description	This use allows the student to write a review about the house and can be view by the landlord and delete PETAKOM.
Actor	Student, landlord, PETAKOM.
Preconditions	Must have an account.
Basic Flow	Student 1. The use case start when the student login to the system. 2. Student click the house 3. Student click <add review>> button. 4. Student write a review 5. Student click <<Post>> button. 6. Use case end. PETAKOM and Landlord. 1. Users login to the system 2. Click house to view the review 3. Read the review write by student 4. Petakom can [A1-delete the review]

	5. Use case end.
Alternative Flow	A1-delete the review 1. PETAKOM choose the post 2. Select review to delete 3. Click<<delete>> button. 4. Use case end.
Exception Flow	
Post-Conditions	Student able to write a review. Landlord can view the review. PETAKOM can delete inappropriate review.
Rules	Must login to the system
Constraints	
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.7 Proof of Payment

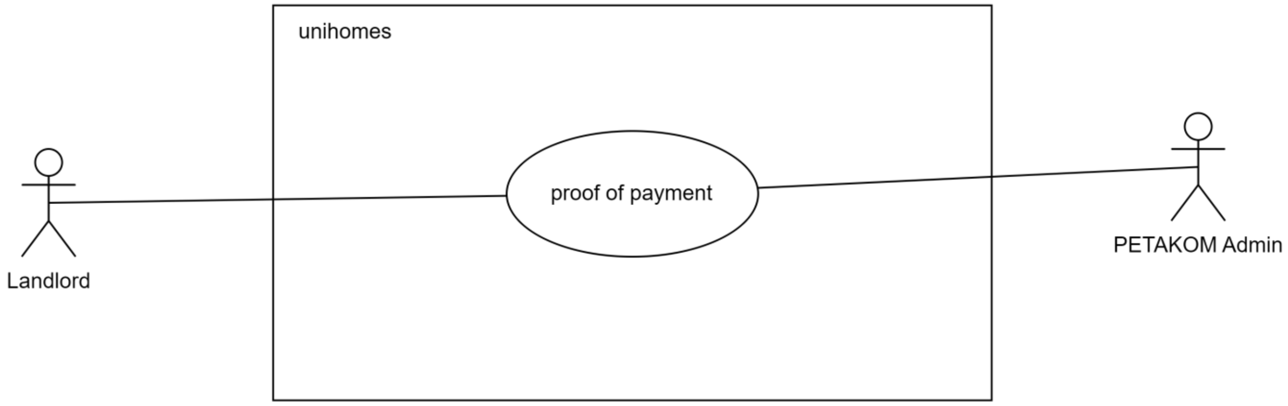


Figure 2: Proof of payment

Table 3: proof of payment

Use Case ID	UH-UCD-07
Brief Description	This use case allows the landlord to upload proof of payment. PETAKOM can view the receipt uploaded by the landlord
Actor	Landlord and PETAKOM
Preconditions	must fill in the information and upload document
Basic Flow	Landlord 1. Click register account 2. Fill in information needed 3. Click upload file 4. Upload payment receipt 5. Use case end PETAKOM 1. Login to the system 2. Click <<dashboard>> 3. View list of application 4. Click an application 5. View application information 6. Click <<approve>>

	7. Use case end.
Alternative Flow	
Exception Flow	
Post-Conditions	PETAKOM can approve or reject landlord application
Rules	PETAKOM must login to the system.
Constraints	
Sequence Diagram	Refer Appendix
Interface	Refer Appendix

3.8 Generate Report

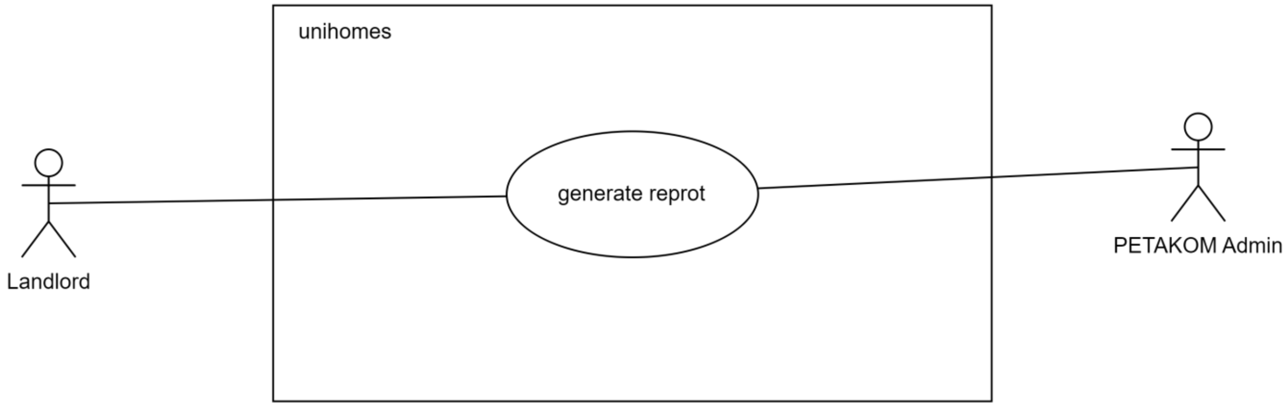


Figure 2: generate report

Table 3: generate report

Use Case ID	UH-UCD-08
Brief Description	This use case allows the landlord and PETAKOM to get graphical analytic in the system
Actor	Landlord and PETAKOM
Preconditions	Must login to the system
Basic Flow	Landlord and PETAKOM 1. Landlord PETAKOM login to the system 2. Click report tab 3. Landlord and PETAKOM can view the graphical analytic of their account. 4. Use case end.
Alternative Flow	
Exception Flow	
Post-Conditions	Landlord and PETAKOM can view graphical analytic in the system
Rules	
Constraints	
Sequence Diagram	Refer Appendix

Interface	Refer Appendix

4.0 REQUIREMENT TRACEABILITY

This section shall contain:

a. Traceability from each software unit identified in this SRS of the system requirements allocated to it.

b. Traceability from each system requirement for the software units to which it is allocated.

Table 4: Requirement Traceability

Requirement	Description
UH-UCD-01	Register and login The user should be able to register and login to the system
UH-UCD-02	Manage student profile The student able to manage their profile
UH-UCD-03	Manage landlord profile The landlord able to manage their profile
UH-UCD-04	Manage house detail The student able to manage their house detail
UH-UCD-05	Search Student able to do search
UH-UCD-06	Manage review Student and PETAKOM able to manage review
UH-UCD-07	Proof of payment Landlord able to submit proof of payment
UH-UCD-08	Generate report Landlord and PETAKOM able to view analytics

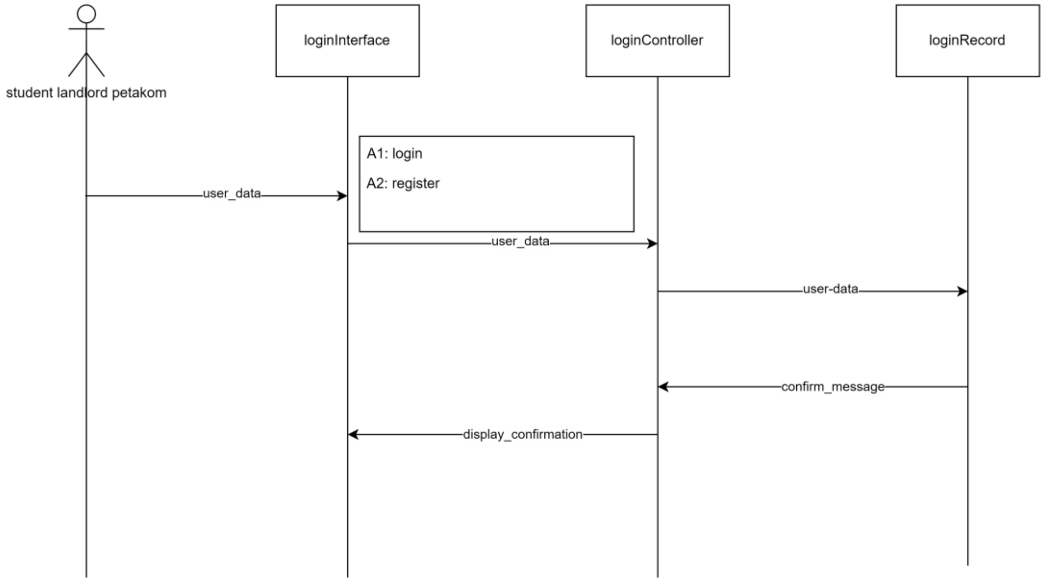
5.0 SYSTEM REQUIREMENTS APPROVAL

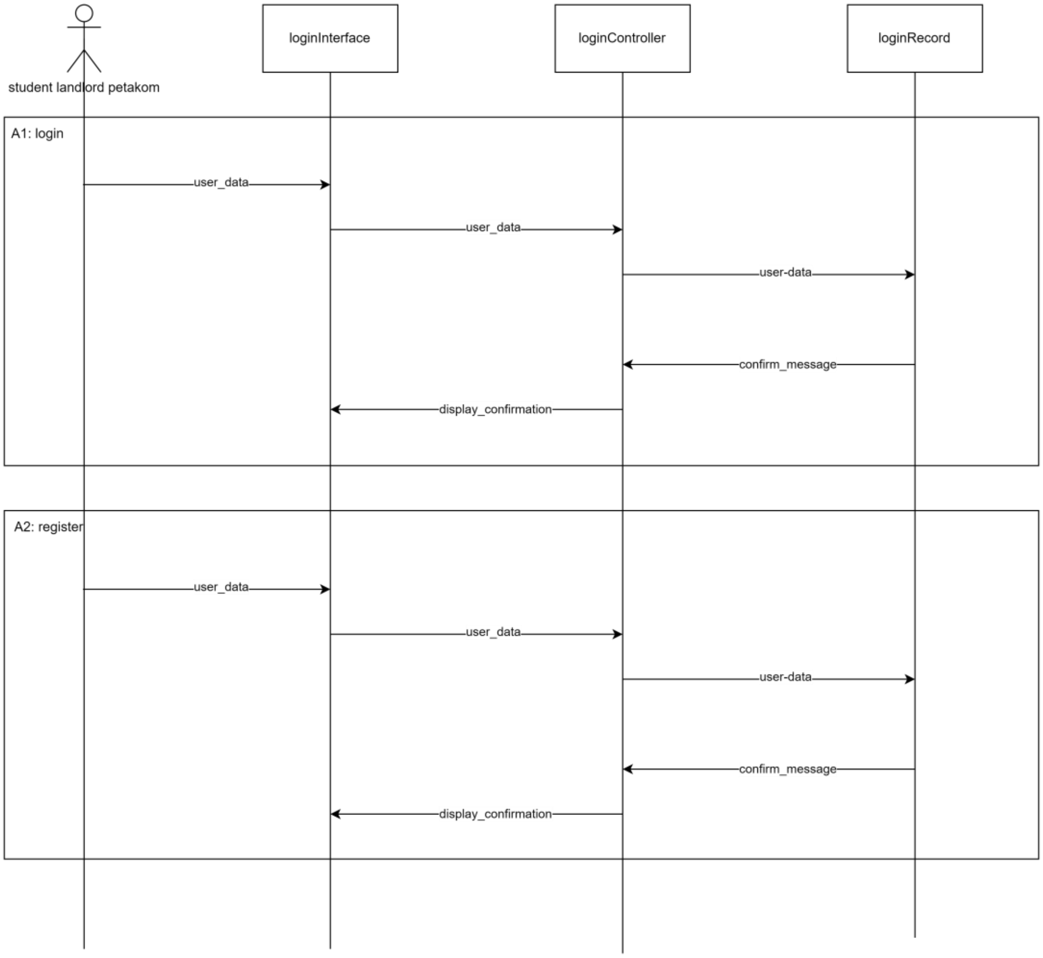
	Name	Date
Verified by: ________________________ Developer		
Approved by: ________________________ Client		

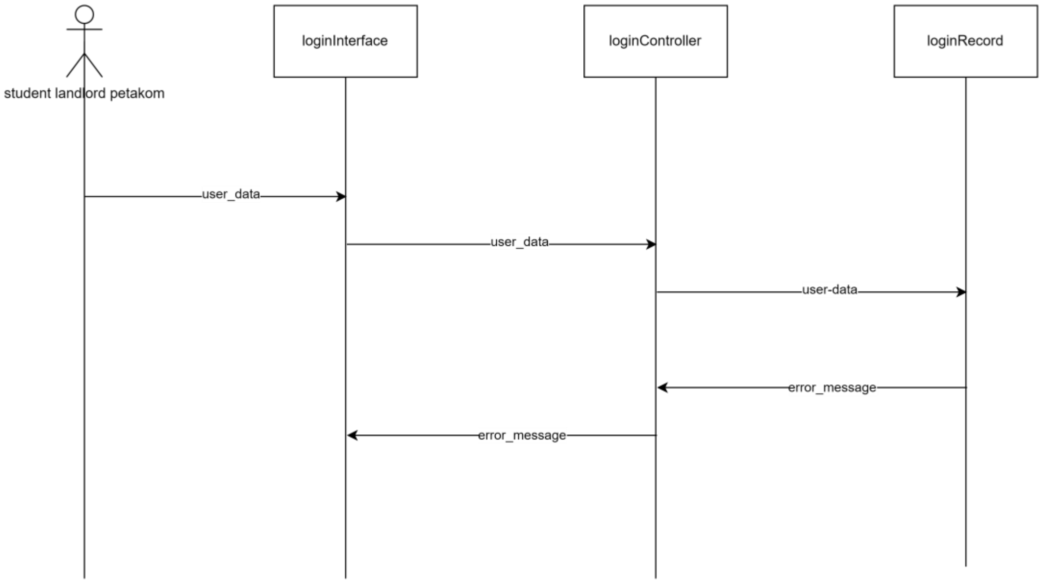
APPENDIX B -1

SEQUENCE DIAGRAM

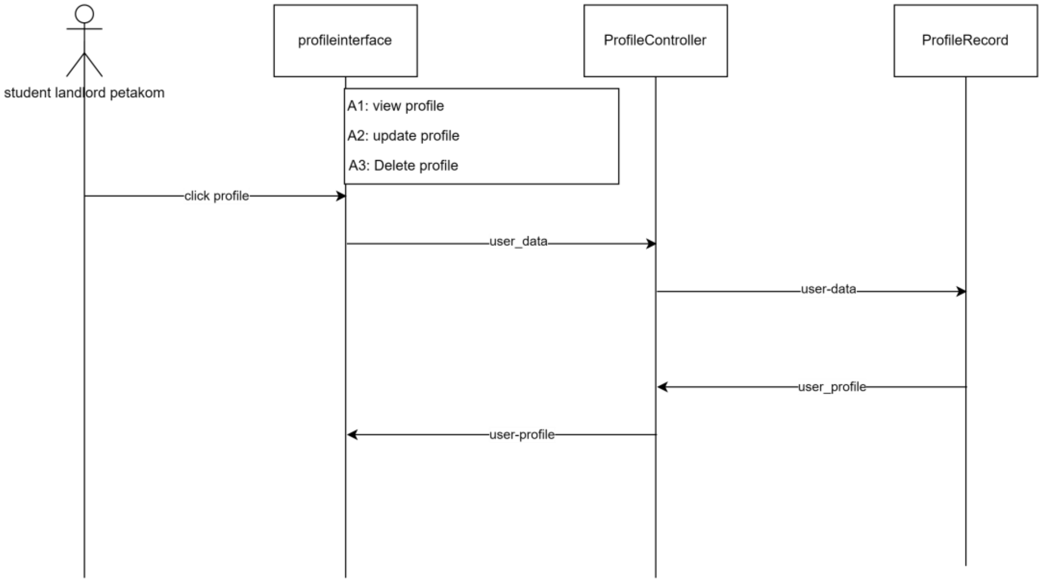
Login and Register

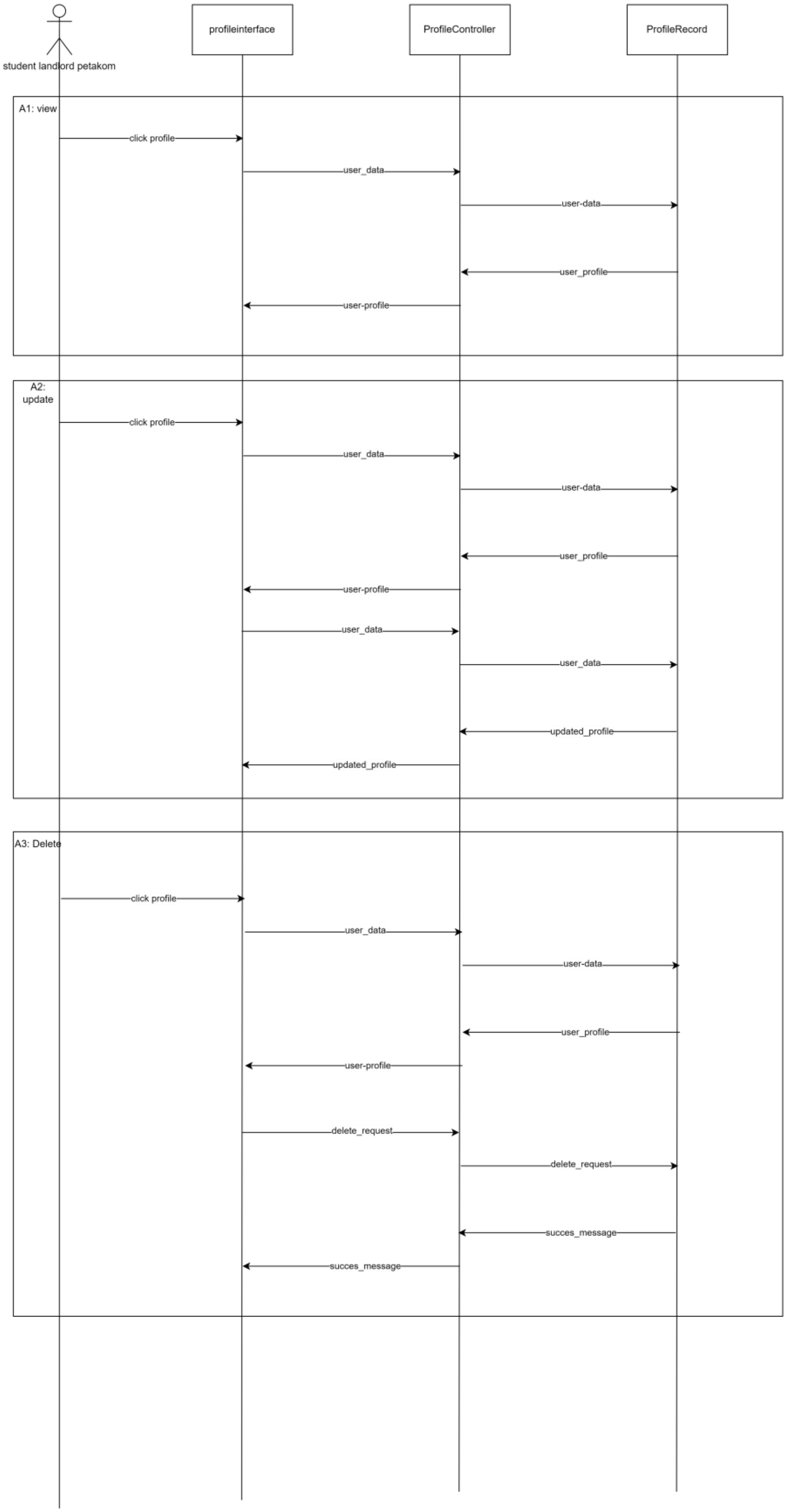


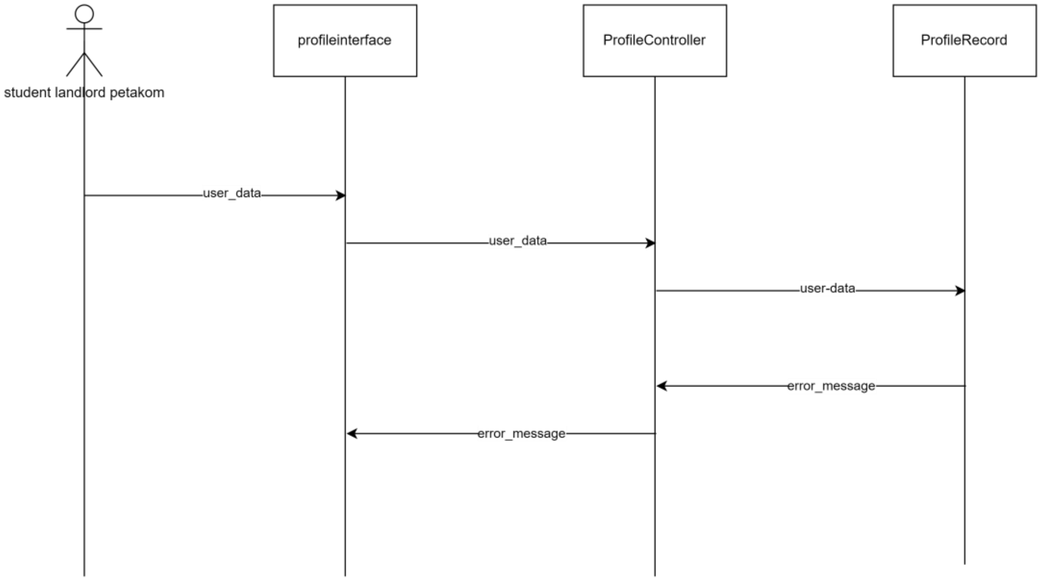



student landlord petakom
loginInterface
loginController
loginRecord
user_data
user_data
user-data
error_message
error_message

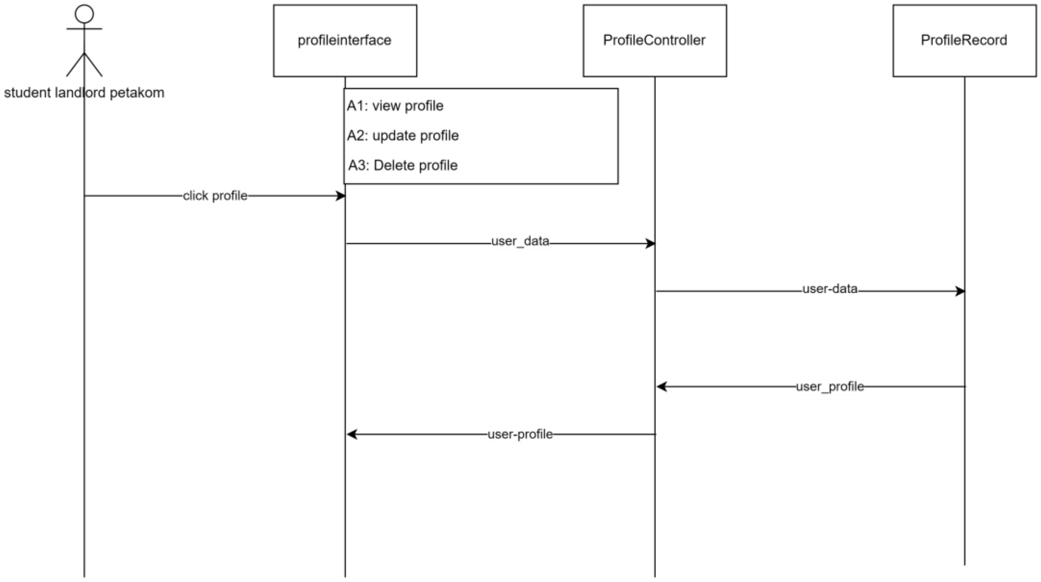
Manage Student Profile

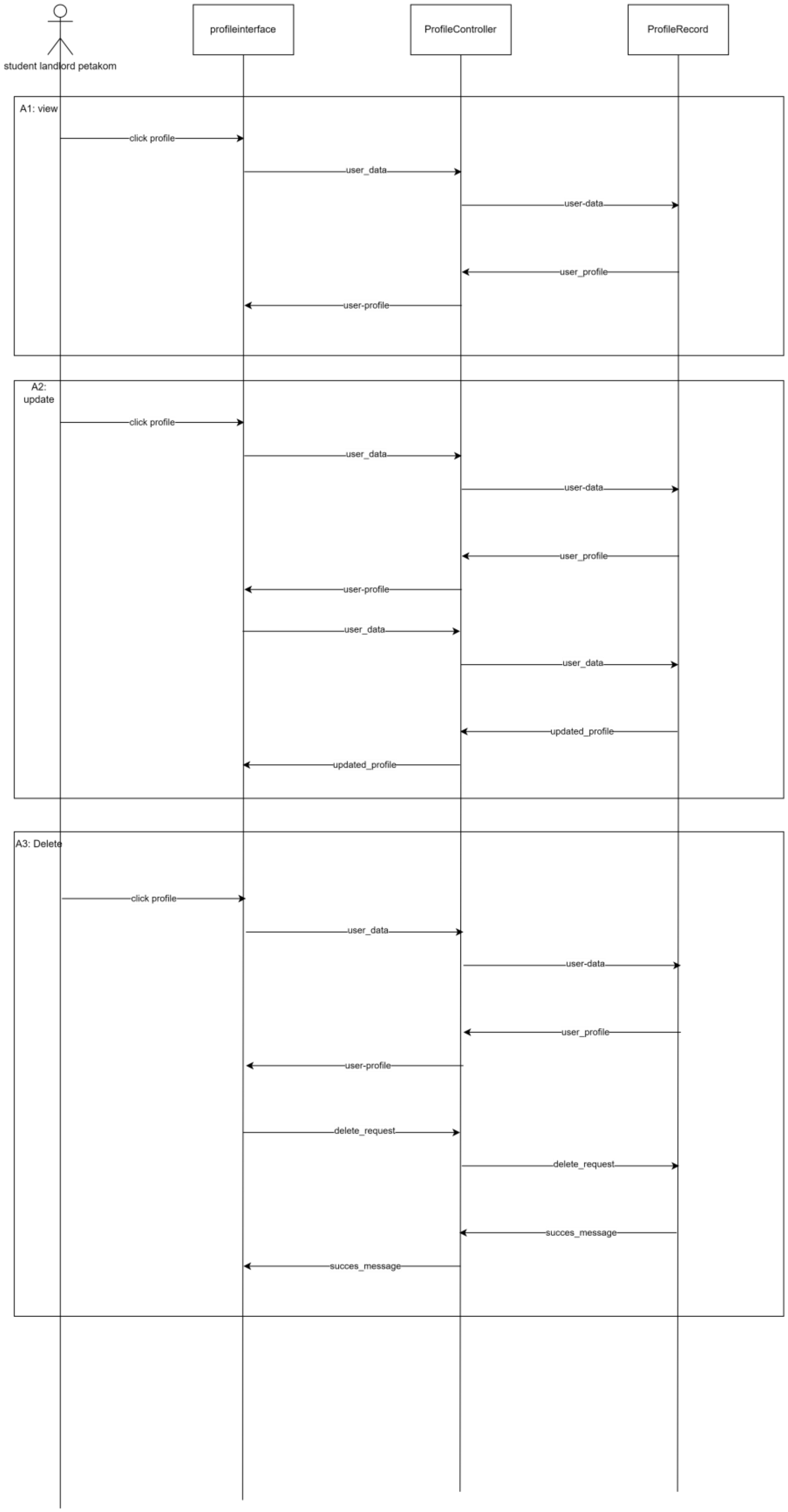


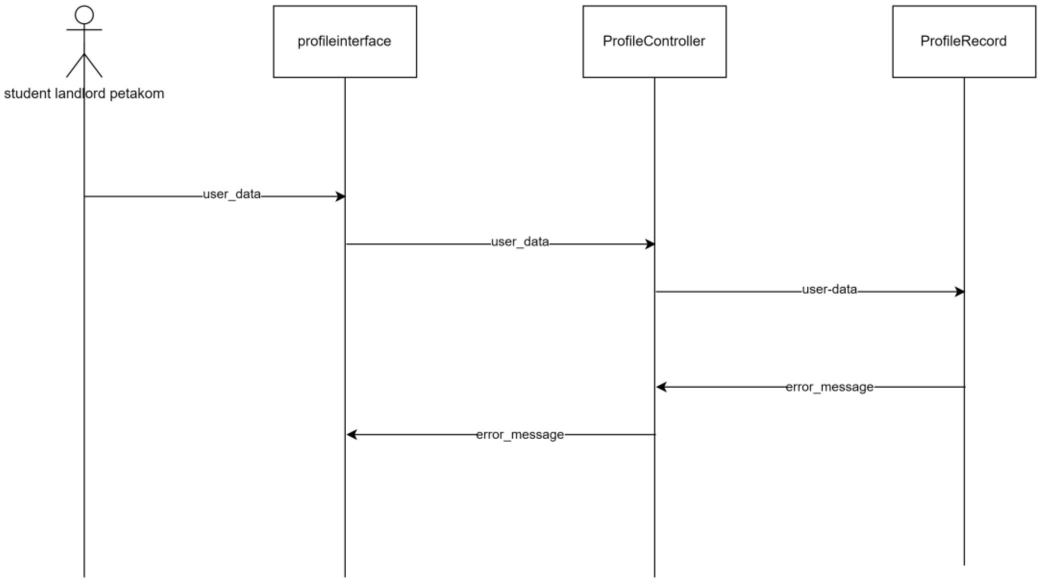

student landlord petakom
profileinterface
ProfileController
ProfileRecord
A1: view
click profile
user_data
user-data
user_profile
user-profile
A2: update
click profile
user_data
user-data
user_profile
user-profile
user_data
user_data
updated_profile
updated_profile
A3: Delete
click profile
user_data
user-data
user_profile
user-profile
delete_request
delete_request
succes_message
succes_message


student landlord petakom
profileinterface
ProfileController
ProfileRecord
user_data
user_data
user-data
error_message
error_message

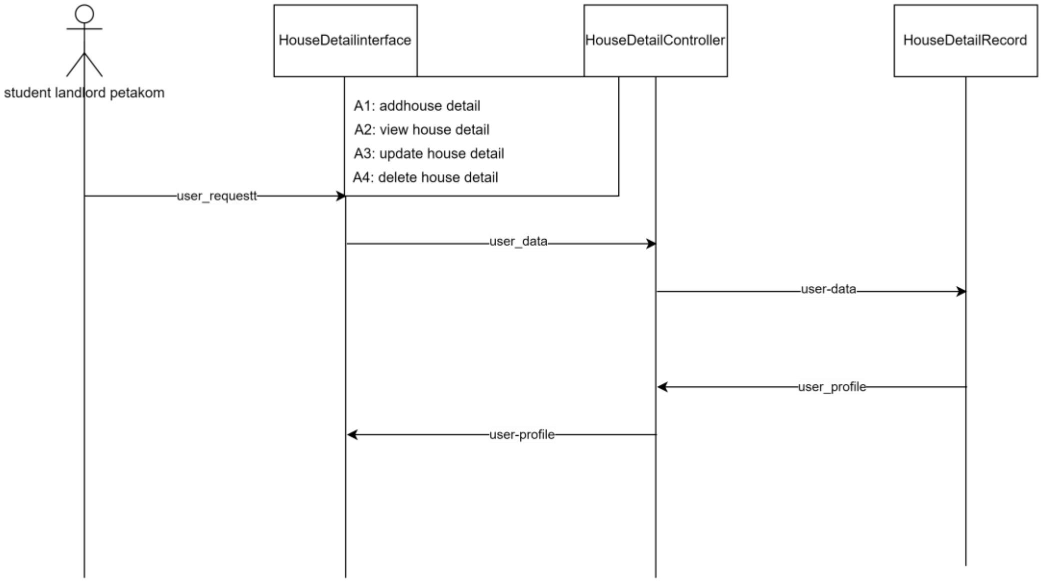
Manage Landlord Profile

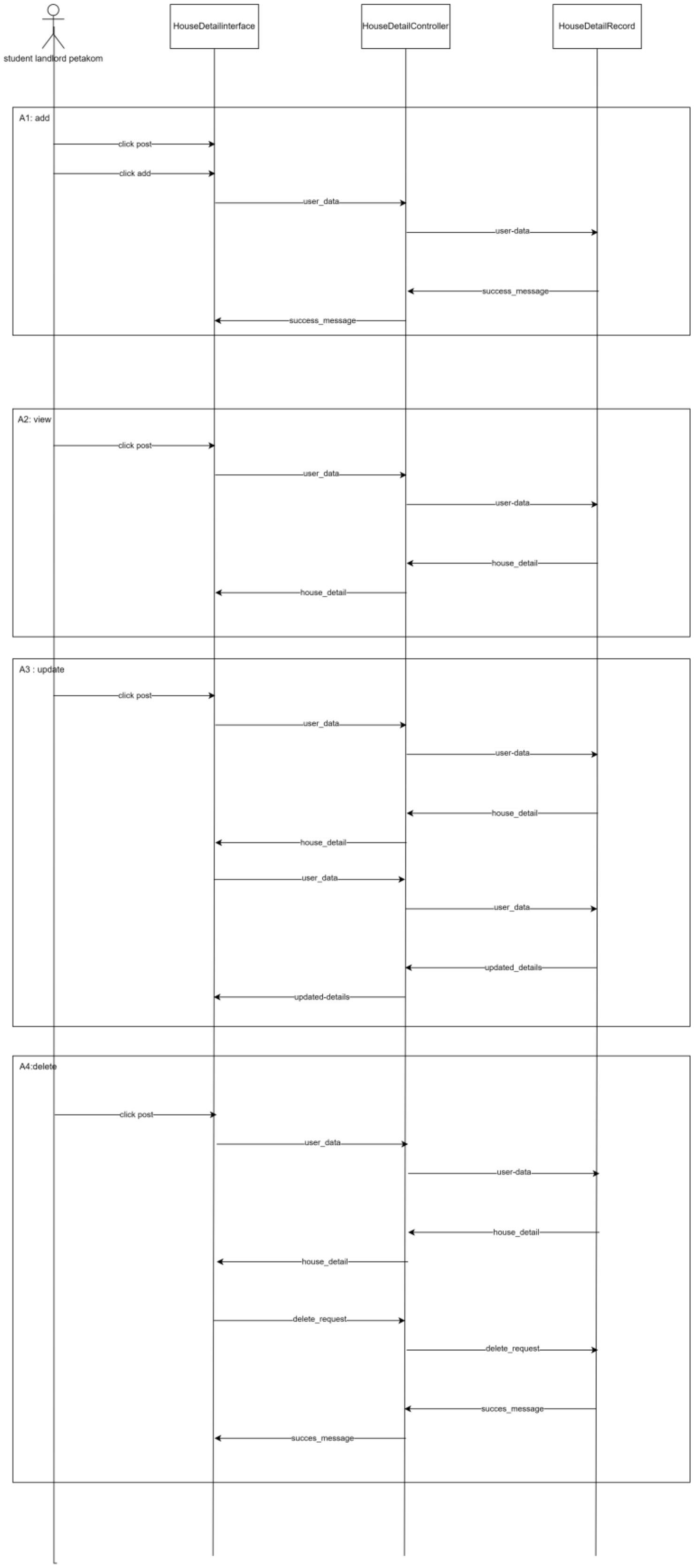


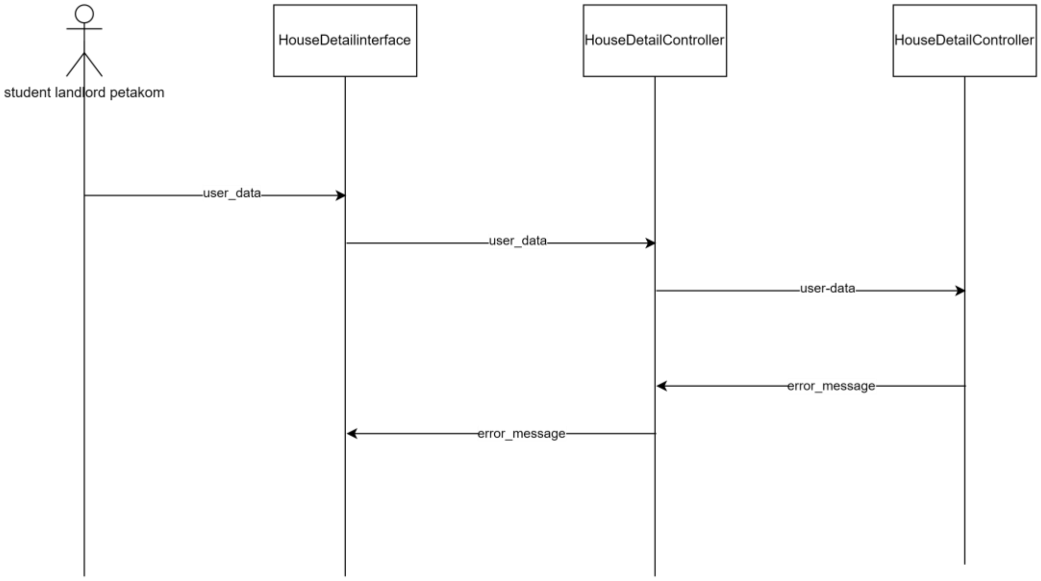

student landlord petakom
profileinterface
ProfileController
ProfileRecord
A1: view
click profile
user_data
user-data
user_profile
user-profile
A2: update
click profile
user_data
user-data
user_profile
user-profile
user_data
user_data
updated_profile
updated_profile
A3: Delete
click profile
user_data
user-data
user_profile
user-profile
delete_request
delete_request
succes_message
succes_message


student landlord petakom
profileinterface
ProfileController
ProfileRecord
user_data
user_data
user-data
error_message
error_message

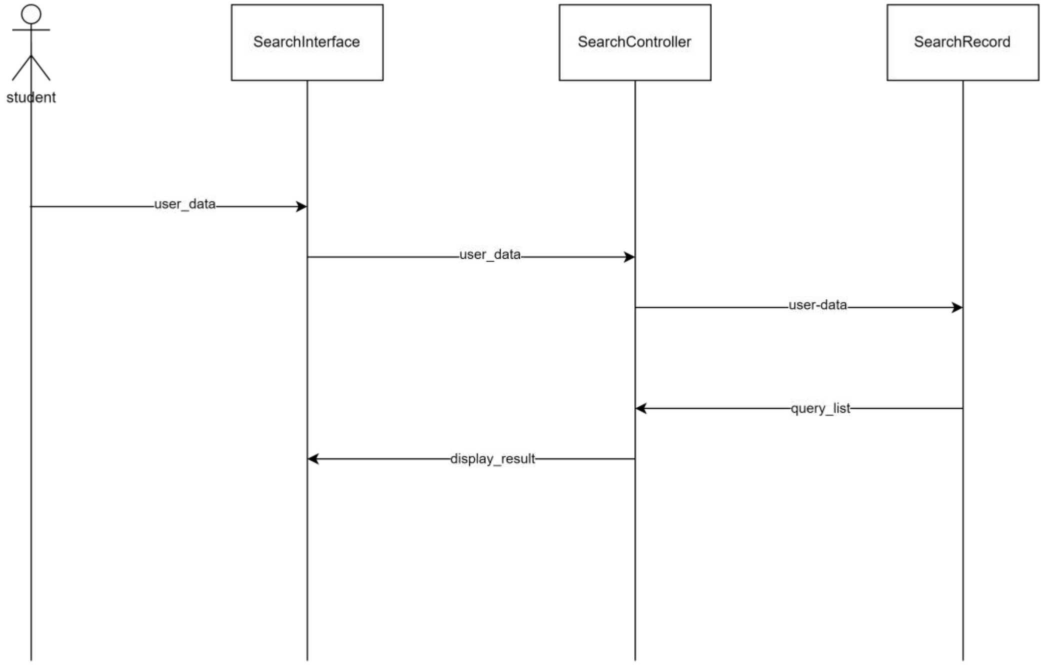
Manage House Detail



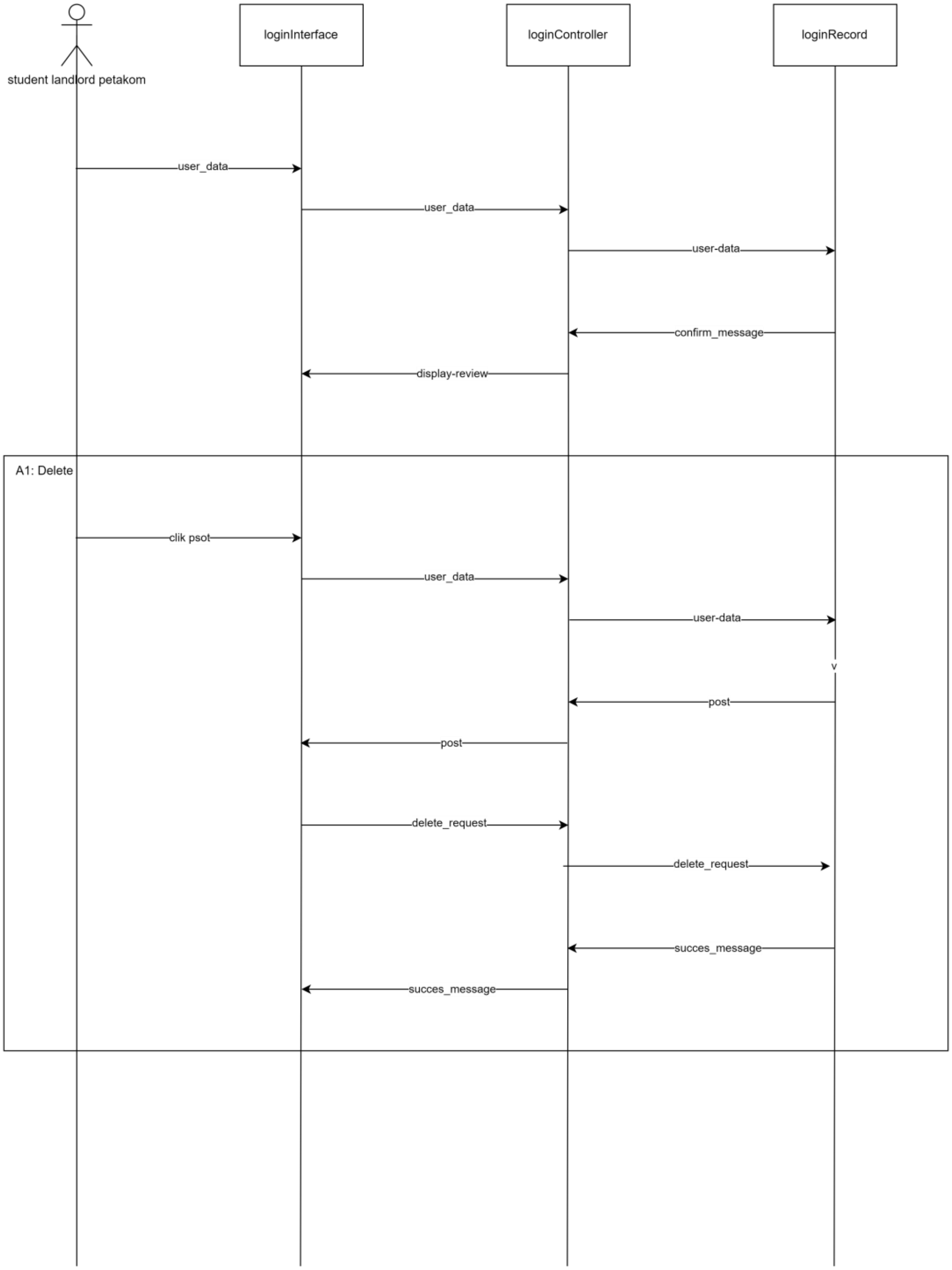

student landlord petakom
HouseDetailinterface
HouseDetailController
HouseDetailRecord
A1: add
click post
click add
user_data
user-data
success_message
success_message
A2: view
click post
user_data
user-data
house_detail
house_detail
A3 : update
click post
user_data
user-data
house_detail
house_detail
user_data
user_data
updated_details
updated-details
A4:delete
click post
user_data
user-data
house_detail
house_detail
delete_request
delete_request
succes_message
succes_message


student landlord petakom
HouseDetailinterface
HouseDetailController
HouseDetailController
user_data
user_data
user-data
error_message
error_message

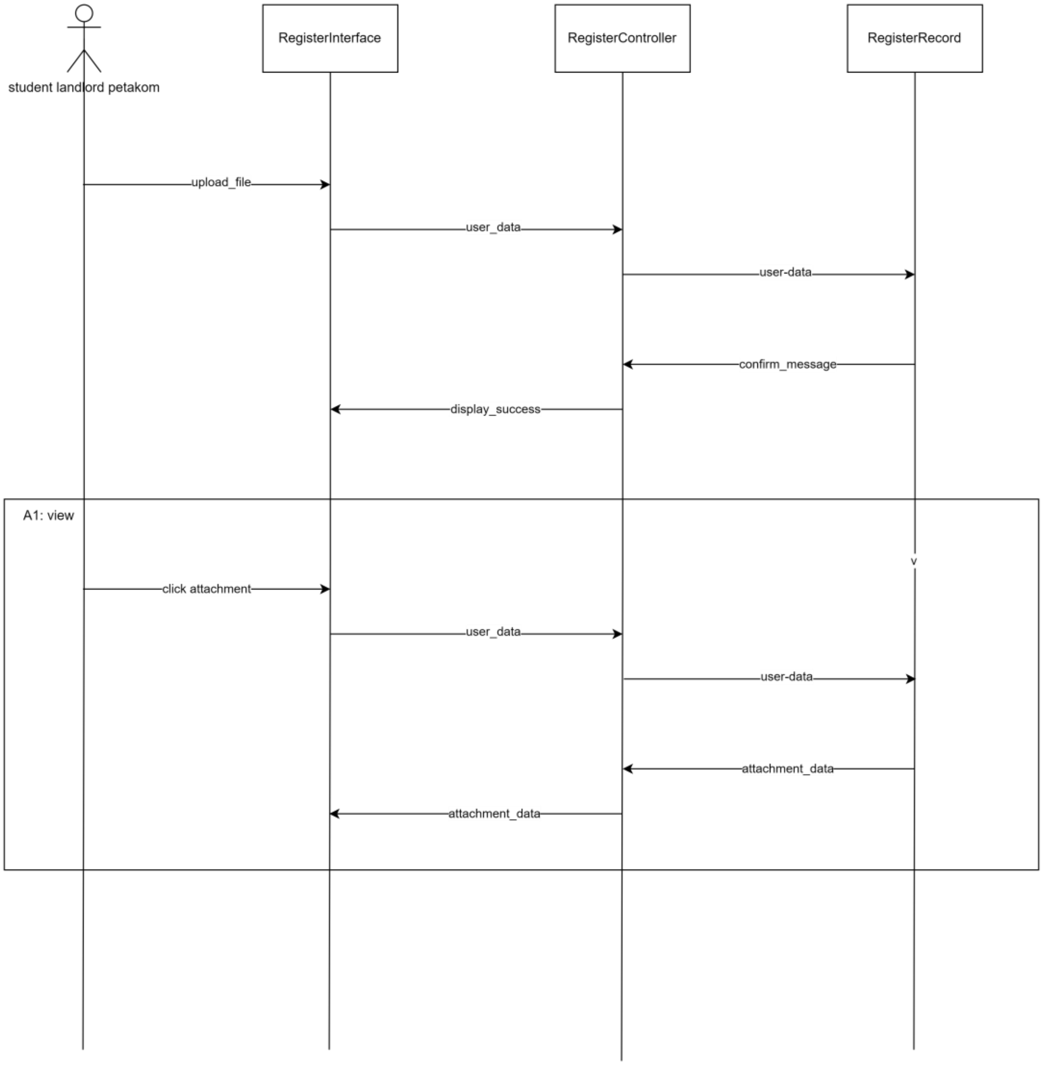
Search


student
SearchInterface
SearchController
SearchRecord
user_data
user_data
user-data
query_list
display_result

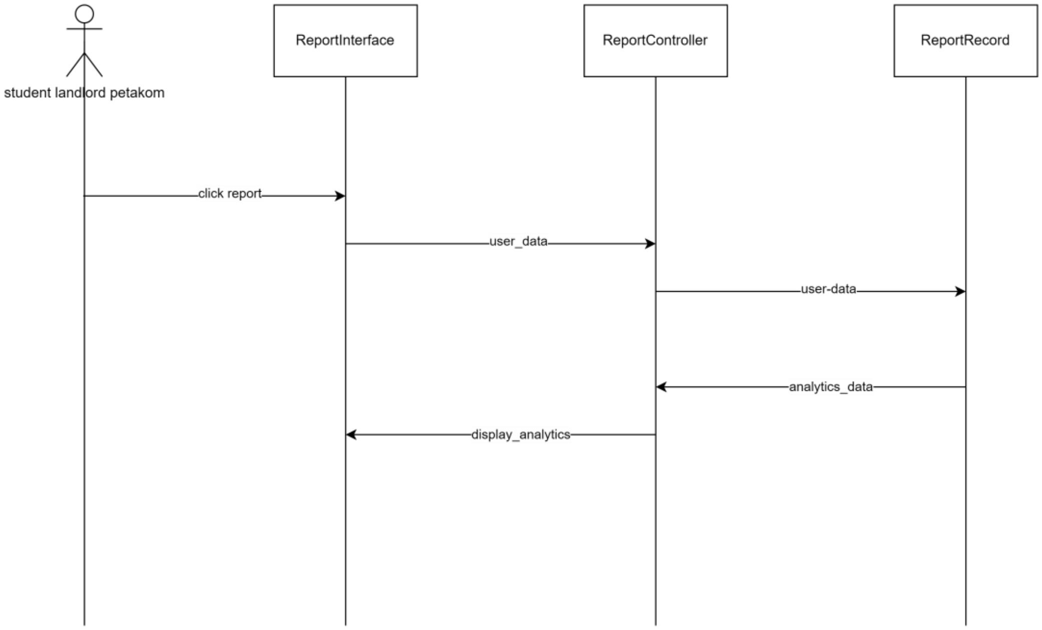
Manage Review



Proof of Payment



Proof of Payment



APPENDIX C
SDD DOCUMENT

Version

1

UNIHOMES

Faculty of Computing

Software Design Document (SDD)

Table of Contents

1.0 DATA DICTIONARY

Available in chapter 3 in thesis

2.0 PRELIMANARY DESIGN

2.1 System Architecture

The UniHomes will be develop using Laravel framework. Model-View-Controller (MVC) architecture is used by the PHP web application framework Laravel. According to the MVC pattern, an application is divided into three primary parts: the model, which represents the data and business logic; the view, which manages the user interface; and the controller, which takes input and organises the model and view.

In Laravel, a class that connects with a database often implements the model, which represents the data. The view serves as a template for creating the HTML that is sent to the user's browser. By updating the model and selecting the right view to present, the controller manages input and coordinates the model and view.

2.1.1 Static Organization

Describe the static organization or your subsystem or package available in the system, and include the external package or library/component that system used.

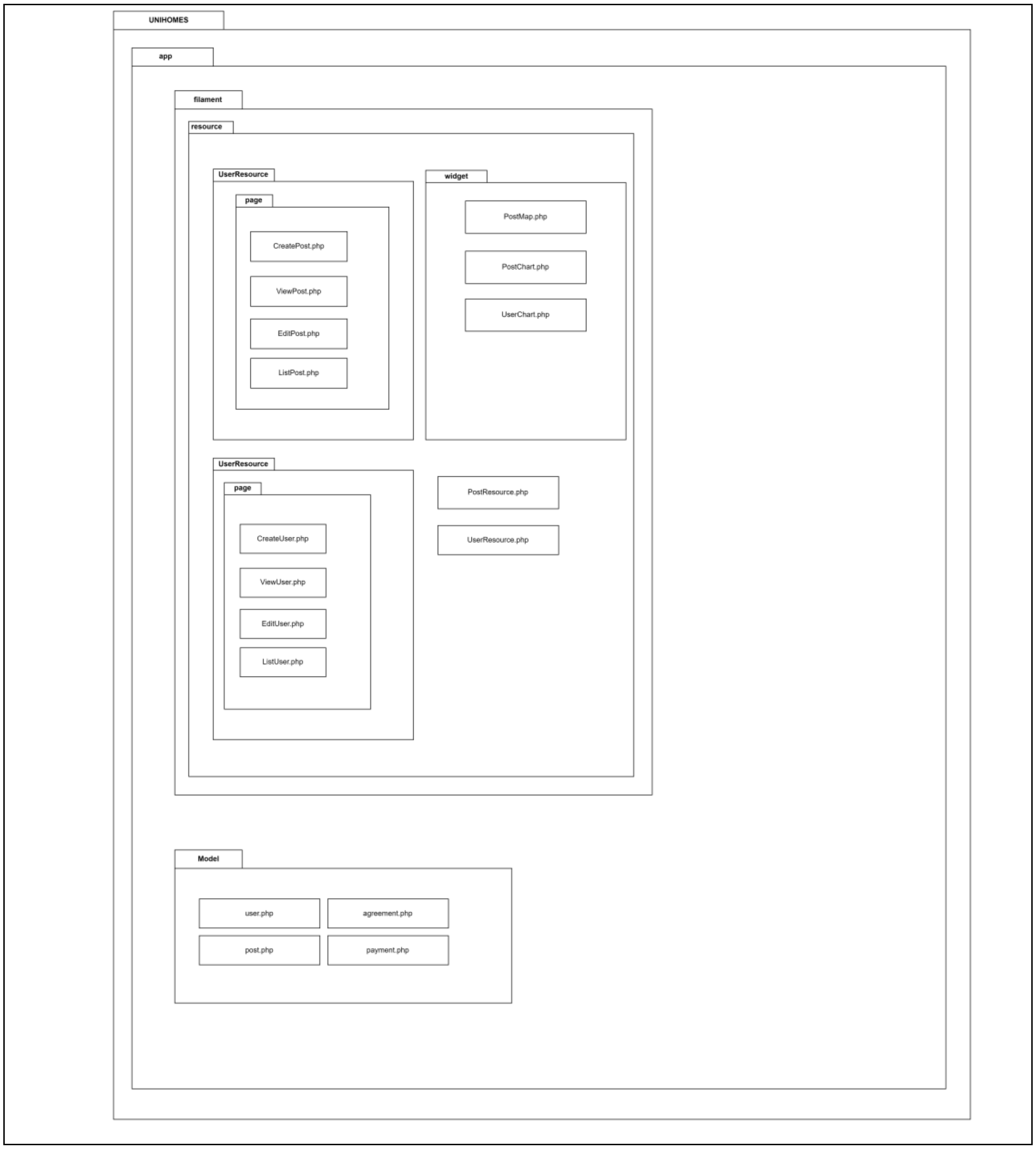


Figure 2.1: Static Organization of UniHomes

This section describes the detail for each subsystem/package .

1. **Model**

This package responsible to represent the data that is being sent to the Controller, data that is being worked on View. This package consists of the following classes or unit

a) User.php
b) post.php
c) Agreement.php
d) Payment.php

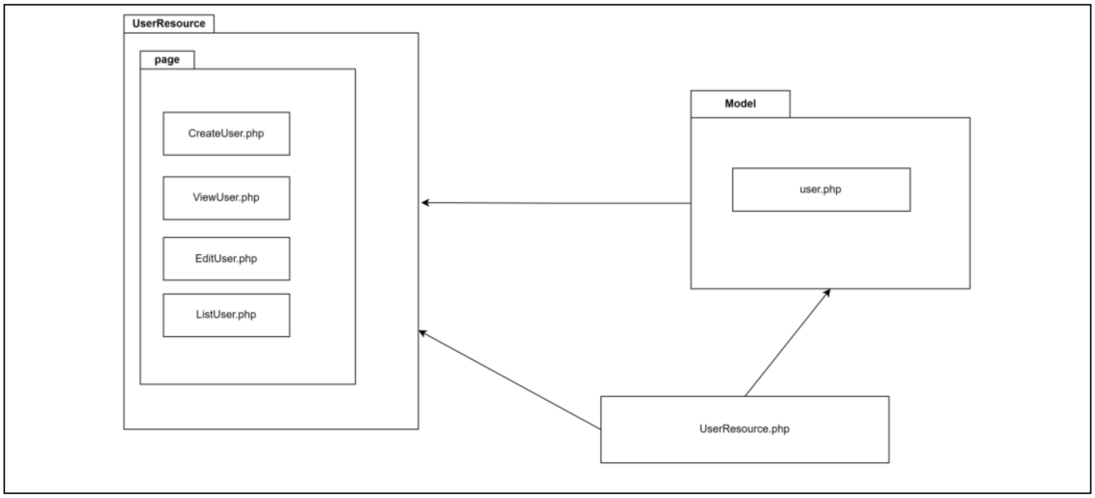
2. **Resource**

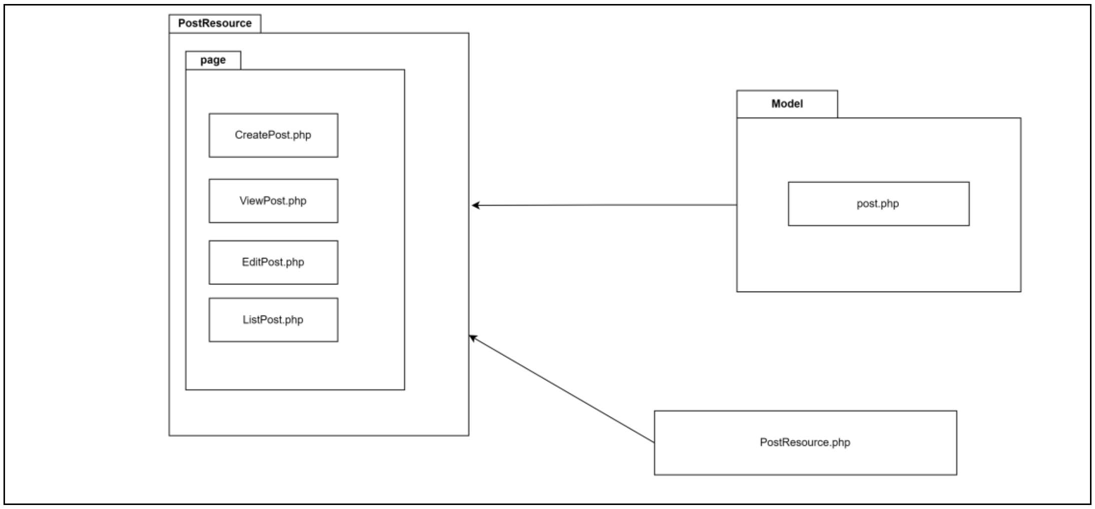
This package contain the interface of the project. This package consists of the following classes or unit

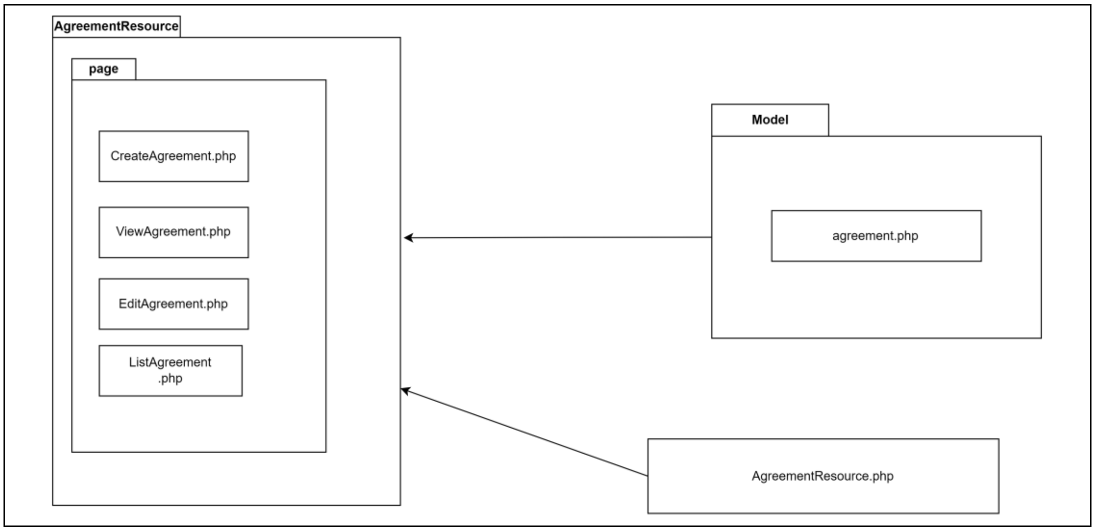
a) UserResource.php
b) PostResource.php
c) AgreementResource.php
d) PaymentResource.php

2.1.2 Dynamic Organization

Describe the components and their relationships between each other in system






AgreementResource
page
CreateAgreement.php
ViewAgreement.php
EditAgreement.php
ListAgreement
.php
Model
agreement.php
AgreementResource.php

3.0 DETAILED DESIGN

This section shall be divided into the following paragraphs to describe each software unit of the systems. If part of all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced.

3.1 User module

3.1.1 user.php

Class Type : model

Responsibility : Responsible for managing the data, performing calculations or operations on the data, and enforcing the business rules of the application.

Attributes : HasRole

Methods : protected $fillable [] : specifies an array of attributes that are mass assignable.
Protected $cast: specify the data types of specific attributes.
Public function post(): to declare relationship

3.1.2 UserResource.php

Class Type : Resource

Responsibility : defines various aspects of the resource, such as its fields, actions, validation rules, and relationships with other resources.

Attributes : $model : string
$favicon : string

Methods : form(): display form
table(): to display the data in table
getPages(): to return pages

3.1.3 ListUser.php

Class Type : Resource

Responsibility : to list the user of the system

Attributes : $resource : string

Methods : getActions: returns an array of actions available for the resource
getHeaderWidget(): to call the widget

3.2. Post module

3.2.1 post.php

Class Type : model

Responsibility : Responsible for managing the data, performing calculations or operations on the data, and enforcing the business rules of the application.

Attributes :

Methods : protected $fillable [] : specifies an array of attributes that are mass assignable.
Protected $cast: specify the data types of specific attributes.
User(): declare relationship with user model
getCoverImageAttribute() : display image in array

3.2.2 PostResource.php

Class Type : resource

Responsibility : defines various aspects of the resource, such as its fields, actions, validation rules, and relationships with other resources.

Attributes : $model: string

Methods : form(): display form
table(): to display the data in table
getPages(): to return pages
getWidget(): call widget

3.2.3 ListPost.php

Class Type : resource

Responsibility : to list the post of the system

Attributes : $model: string

Methods : getHeaderWidget(): to call the widget
getTableQuery() : custome query
getActions: returns an array of actions available for the resource

3.3. Agreement module

3.3.1 Agreement.php

Class Type : model

Responsibility : Responsible for managing the data, performing calculations or operations on the data, and enforcing the business rules of the application.

Attributes :

Methods : protected $fillable [] : specifies an array of attributes that are mass assignable.

Protected $cast: specify the data types of specific attributes.

User(): declare relationship with user model

getCoverImageAttribute() : display image in array

3.3.2 AgreementResource.php

Class Type : resource

Responsibility : defines various aspects of the resource, such as its fields, actions, validation rules, and relationships with other resources.

Attributes : $model: string

Methods : form(): display form

table(): to display the data in table

getPages(): to return pages

getWidget(): call widget

3.3.3 ListAgreement.php

Class Type : resource

Responsibility : to list the post of the system

Attributes : $model: string

Methods : getHeaderWidget(): to call the widget

getTableQuery() : custome query

getActions: returns an array of actions available for the resource

APPENDIX D
UAT

Student

Event	Test Data	Expected Result	Actual Result	pass	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Make a search	Peramu permai	Show post with 'peramu permai'	Show post with 'peramu permai'	pass	
Filter	Filter based on price: ascending	Show result in ascending order	Show result in ascending order	pass	
Contact landlord	Open whatsapp	Open whatsapp	Open whatsapp	pass	

This test perform by:

Name: MUHAMMAD HAZRIQ AKMAL BIN ZAIROL

Date: 8/6/2023

Signature:

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Make a search	Peramu permai	Show post with ‘peramu permai’	Show post with ‘peramu permai’	pass	
Filter	Show result in ascending order	Show result in ascending order	Show result in ascending order	pass	
Contact landlord	Open whatsapp	Open whatsapp	Open whatsapp	pass	

This test perform by:

Name: MUHAMMAD FYRUZ ISMAT BIN AZMI

Date: 8/6/2023

Signature:

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Make a search	Peramu permai	Show post with ‘peramu permai’	Show post with ‘peramu permai’	pass	
Filter	Show result in ascending order	Show result in ascending order	Show result in ascending order	pass	
Contact landlord	Open whatsapp	Open whatsapp	Open whatsapp	pass	

This test perform by:

Name: SALMAN BIN KHAIRUL ANUAR

Date: 8/6/2023

Signature:

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Make a search	Peramu permai	Show post with ‘peramu permai’	Show post with ‘peramu permai’	pass	
Filter	Show result in ascending order	Show result in ascending order	Show result in ascending order	pass	
Contact landlord	Open whatsapp	Open whatsapp	Open whatsapp	pass	

This test perform by:

Name: MUHAMMAD HARITH AIZAT BIN SUHAILI

Date: 8/6/2023

Signature:

Landlord

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Create post	House name: Rumah peramu permai Price per month:180 Phone number:0245679800 Description:fully furnisher Address: Lorong 8 rumah peramu permai Number of room: 3 House capacity: 10	Post created	Post created	pass	
Edit post	Price per month:200	Price per month change to 200	Price per month change to 200	pass	
Delete post	Click delete	Post delete	Post deleted	pass	

Display graph	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	pass	
Upload payment proof	Upload pdf	Pdf uploaded	pdf uploaded	pass	
Create agreement	Create agreement	Agreement created	Agreement created	pass	

This test perform by:

Name: FIRDAUS BIN MOHD SIDEK

Date: 8/6/2023

Signature

Landlord

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Create post	House name: Rumah peramu permai Price per month:180	Post created	Post created	pass	

	Phone number:0245679800 Description:fully furnisher Address: Lorong 8 rumah peramu permai Number of room: 3 House capacity: 10				
Edit post	Price per month:200	Price per month change to 200	Price per month change to 200	pass	
Delete post	Click delete	Post delete	Post deleted	pass	
Display graph	Graph is display in dashboard	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	pass	
Upload payment proof	Upload pdf	Pdf uploaded	pdf uploaded	pass	
Create agreement	Create agreement	Agreement created	Agreement created	pass	

This test perform by:

Name: AHMAD KHAIRUL IMAN BIN AHMAD SAYUTI

Date:8/6/23

Signature

Landlord

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Create post	House name: Rumah peramu permai Price per month:180 Phone number:0245679800 Description:fully furnisher Address: Lorong 8 rumah peramu permai Number of room: 3 House capacity: 10	Post created	Post created	pass	Map does not correctly shown location
Edit post	Price per month:200	Price per month change to 200	Price per month change to 200	pass	
Delete post	Click delete	Post delete	Post deleted	pass	

Display graph	Graph is display in dashboard	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	pass	
Upload payment proof	Upload pdf	Pdf uploaded	Pdf uploaded	pass	
Create agreement	Create agreement	Agreement created	Agreement created	pass	

This test perform by:

Name: MUHAMMAD IQMAL HAKIM BIN AMERUDDIN

Date: 8/6/23

Signature

Petakom

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Delete post	Click delete	Post deleted	Post deleted	pass	
Display graph	Graph is display in dashboard	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	pass	
Manage user	Can view, and delete user	Can view, and delete user	Can view, and delete user	pass	

This test perform by:

Name: AHMAD HISYAM BIN SURYANTO SUGIAN

Date: 8/6/23

Signature

Petakom

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Delete post	Click delete	Post deleted	Post deleted	pass	
Display graph	Graph is display in dashboard	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	pass	
Manage user	Can view, and delete user	Can view, and delete user	Can view, and delete user	pass	

This test perform by:

Name:KHAIRIN CHAN BIN IBRAHIM CHAN

Date: 8/6/23

Signature

Petakom

Event	Test Data	Expected Result	Actual Result	Pass/Fail	Comment
Login	Email:test@mail.com Password:12345678	User logged in	User logged in	pass	
Register	Name:test1 Email:test@mail.com Password:12345678 Confirm password: 12345678	User registered	User registered	pass	
Manage profile	Can view profile	Can view profile	Can view profile	pass	
Delete post	Click delete	Post delete	Post deleted	pass	
Display graph	Graph is display in dashboard	Bar chart and pie chart shown in dashboard	Bar chart and pie chart shown in dashboard	pass	
Manage user	Can view, and delete user	Can view, and delete user	Can view, and delete user	pass	

This test perform by:

Name:NIK AHMAD FARIHIN BIN ZULKIFLI

Date: 8/6/23

Signature