



Daffodil
International
University

House Rental system

Submitted By:

Asadur Rahman Didar

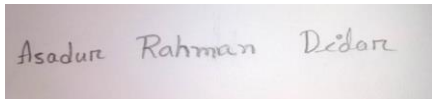
ID: 161-35-1459

This Project report has been submitted in finishing of the
Requirements
for the Course of
Bachelor of Science in Software Engineering.

Department of Software Engineering
Daffodil International University

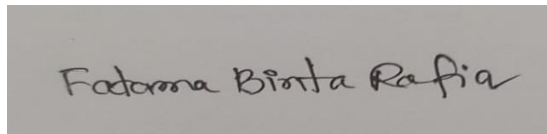
DECLARATION

I, hereby, declare that I have taken this project under the supervision of **Ms. Fatama Binta Rafiq, Lecturer, Department of Software Engineering, Daffodil International University**



.....
Asadur Rahman Didar
ID: 161-35-1459
Batch: 19th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified By:



.....
Ms. Fatama Binta Rafiq
Lecturer
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

APPROVAL

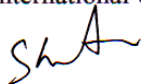
This thesis titled on “**House Rental System**”, submitted by **Asadur Rahman Didar (ID: 161-35-1459)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS



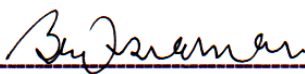
Dr. Imran Mahmud
Head and Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman



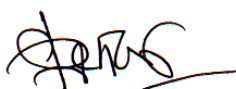
Md. Shohel Arman
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1



Khalid Been Badruzzaman Biplob
Lecturer (Senior)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2



Md. Tanvir Quader
Senior Software Engineer
Technology Team
a2i Programmer

External Examiner

Acknowledgment

At first, I would like to thank almighty Allah. I have taken efforts in this project. It would not have been possible without the kind support and help of many persons. I would like to extend my sincere thanks to all of them.

I also sincerely thank our respected supervisor **Ms. Fatama Binta Rafiq** for the guidance and encouragement for finishing this project. I am thankful for his inspirations to work harder and for motivating me to be better.

Finally, I would like to thank my parents for keeping me in their prayers and supporting me to be better at every step of the way. Without their love and support I would not be successful.

To sum up, without the help of the mentioned supports, the project would not be possible.

Dedication

I dedicate this project to my respectable Father and Mother, my supervisor, my Honorable teachers who are always dear and near to me. Without their patience, understanding, unsparing support, care, affection and love it was not possible to come up to this place.

Executive summary

The following are the services that Best Bet Property Management will provide:

- Pre-rental analysis
- House marketing
- Renter (Tenant) profile view
- Booking house
- Move-in/out inspections
- Maintenance
- House rent collection(ascending to descending)

Table Content

Contents	Page
Declaration	II
Approval	III
Acknowledgement	IV
Dedication	V
Executive summary	VI
List of Figure	VII
Chapter 1: Introduction	1
1.1. Project Overview	2
1.2. Project Purpose	3
1.2.1. Background	3
1.2.2. Benefit	3
1.2.3. Goals	3
1.3. Stakeholders	4
Chapter 2: Software Requirements Specification	
2.1. Functional Requirements	6
2.2. Performants Requirements	7
2.3. Dependability Requirements	8
2.4. Maintainability and Supportability Requirements	9
2.5. Security Requirements	9
2.6. Usability and Human Interaction Requirements	10
Chapter 3: System Analysis	
3.1. Use Case Diagram	12
3.2. Use Case Description	13
3.3. Activity Diagram	20
3.4. System Sequence Diagram	24
Chapter 4: System Design Specification	
4.1. Sequence Diagram	31
4.2. Class Diagram	32
4.3. Development Tools & Technology	33
Chapter 5: System Testing	
5.1. Testing Feature	35
5.2. Testing Strategies	36
5.3. Testing Environment	39
Chapter 6: User Manual	
Chapter 6, Section 6.1 To 6.11	50
Chapter 7: Project Summary	
7.1. Critical Evolution	47
7.2. Limitation	47
Chapter 8: Reference	
Reference	50

List of Figure

Figures	Page
Figure 3.1. Use case	12
Figure 3.2. Activity Diagram for user	20
Figure 3.3. Activity Diagram for admin	21
Figure 3.4. Activity Diagram for booking system	22
Figure 3.5. Activity Diagram for admin (viewing house)	23
Figure 3.6. Sequence Diagram for user	24
Figure 3.7. System Sequence diagram for Landlord	25
Figure 3.8. System Sequence diagram for Landlord 1	26
Figure 3.9. System Sequence diagram for Landlord 2	27
Figure 3.10. System Sequence diagram for add booking (renter)	28
Figure 3.11. System Sequence diagram for delete booking (renter)	29
Figure 4.1. System Sequence diagram for Landlord	31
Figure 4.2. Class Diagram for house rental system	32
Figure 6.1. Home page for house rental system	41
Figure 6.2. Registration page for house rental system	42
Figure 6.3. Login page for house rental system	43
Figure 6.4. Dashboard for landlord	43
Figure 6.5. Landlord profile information	44
Figure 6.6. Dashboard for rental panel	44
Figure 6.7. Dashboard for all house areas	45

Chapter 1

Introduction

Introduction

A web-based application management system called House Rental System allows landlords and renters to communicate more effectively. The goal of this project is to simplify for the homeowner the method for managing their home. Many renters and landlords retain unneeded records of every transaction they have. Furthermore, without a concrete document, they are unable to determine the true profit or cost. Which has caused them both great suffering. Their entire process is simplified by this system.

1.1 Project Overview:

My project was to create a web application platform for booking homes online.

Admins manage the entire project. While the landlord is adding a house. The renter then reserves a house suitable for their needs and manages this booking. Admin will monitor and manage the system. They will have the full access to the website. They can see all the house and booking lists if they want. They also have the access for the User information. But only basic ones such as Name, Email, Phone Number and so on. However, a Landlord can add their house and manage them later. They can see the booking information of their. Along with that, a Renter can book a living space from anywhere with their preferences. They also can manage that information later on. They can see some basic information of the house as well. Above all, all of the users can have access to their own profile. And they can manage that information too. On the other hand, as for the system, Landlord and Renter will receive an email after every successful booking and after the booking has been expired. The old management experienced some irregularities from the beginning. Both landlords and tenants suffer from this outdated management system. House Rent System is a simple web-based account management system that can solve most problems. Users do not need to learn accounting principles to view this web application. Easier than today's social media sites. Owners connect directly with renters using messages using the system and notifications. What floor does the message occupy? Owners can store tenant details in the system. If a tenant commits an incident, this information will help put the tenant into a trance. No need to look through the old files provided to him when renting an apartment. Also, the other side of the tenant receives notifications and messages if provided by the owner. He can see his past and his upcoming due dates. Additionally, he can view total payments. It can cause problems when home repairs are needed. One month before you can give your owner consent to move out before you leave the house.

1.2 Project Purpose:

Reserving a vacant room for yourself is usually difficult as a lot of time is spent actually looking for a property, so doing it over the internet is much easier. Owners can store renter details in the system. If a renter commits an incident, this information will help put the renter into a trance. Our main focus was to find house easily for renters/tenants.

1.2.1. Background:

A house rental system is a computer program that manages the process of renting rooms and properties. This is a software application that helps renter (tenant) to manage their properties by tracking landlords, rent price, location.

Our background is web-application which handles our whole tasks while other tasks are handled by machines.

1.2.2. Benefit:

Benefits of this project is that users can find their suitable living space while sitting in their home. They do not need to go to the field to find living space. This will help them to save their time and efforts.

By successful completion of this project, users will be able to-

1. Automatic creation of renter registration form
2. Landlord can add, edit or delete renter history
3. Landlords have a simple and smart dashboard
4. Renters can issue exit permits
5. Renters can resolve repair issues at home

1.2.3. Goals:

Admin can easily manage the application and monitor the required tasks. Such as, User information, House information, House list, booking information, booking list and so on. On the other hand, a Landlord can easily add house and manage them. They can also have access to the required information about the user who reserves a vacant apartment. While the Renter can easily reserve a house suitable for his/her needs and manage that booking information.

Landlord can store renter details in the system. If a renter commits an incident, this information will help put the renter into a trance. No need to look through the old files provided to him when renting an apartment. Also, the other side of the renter receives notifications and messages if provided by the owner.

Owners can store renter details in the system. If a renter commits an incident, this information will help put the renter into a trance. No need to look through the old files provided to him when renting an apartment. Also, the other side of the renter receives notifications and messages if provided by the landlord.

1.3. Stakeholders:

1. Internal Stakeholder:

Business Analyst: Collect requirement from all point of view as little as possible External Stakeholder System Architect: Design the System, Software Developer: Developing software, Landlord: House details

Tester: Testing software Operational team: Deploy it to client

1. Project manager (Admin)
2. Landlord

2. External Stakeholder:

Admin: Owner of the software

User: Renter of the home or apartment

Renter (Tenant)

Chapter 2

Software Requirement Specification

SRS:

All requirement specification described in this section based on this project.

2.1. Functional Requirement:

1. Admin gets all the access
2. Everyone (Admin, Landlord, Renter) have to login
3. Users gets to decide their role (Landlord or renter or Both)

Open Account	Create account as an admin or user
Description	The user wants to use our system
Originator	Null
Stakeholder	<ul style="list-style-type: none">• Owner of the apartment(Landlord)• Renter of the apartment(Tenant)• Software Developer

Login	Login as an internal user
Description	The user wants to access our system
Originator	Null
Stakeholder	<ul style="list-style-type: none">• Owner of the apartment(Landlord)• Renter of the apartment(Tenant)• Software Developer

Manage Dashboard	Managing Dashboard
Description	The user manages his own dashboard
Originator	Null
Stakeholder	<ul style="list-style-type: none">• Owner of the apartment(Landlord)• Renter of the apartment(Tenant)• Software Developer

Rent Manage	Create renter account
Description	The user can create renter account and manage them
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment(Landlord) • Software Developer

Report	Report Generate
Description	The user can view auto generate report
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment(Landlord) • Software Developer

2.2. Performance Requirement:

2.2.1. Speed and Latency Requirement:

The system should have moderate speed, especially if an admin or super admin can manage activities and dashboards.

PR-01	Page Refresh Rate
Description	<p>While homeowners and renters browse this system</p> <p>The page will be displayed immediately. Depending on where you're from</p> <p>Internet connectivity for owners and renter.</p>
Stakeholders	Landlord, Renter

2.2.2. Precision or Accuracy requirements:

There is nothing specific accuracy requirements.

2.2.3. Capacity Requirement:

The system is able to manage all the information home owner, super admin and renter.

PR-02	At first the system will contain all the registered landlord And renter information
Description	The information of registered home owner and renters will be stored in database.
Stakeholders	Landlord, Renters

2.3. Dependability Requirement:

2.3.1. Reliability Requirement:

This is a project related to housing and renter management, so it is of great importance to both landlords and renters. Our project's main goal is to reduce the complexity of the management process.

2.3.2. Availability Requirement:

Availability is more necessary for this project.

DR-01	Must be available the system at 24x7
Description	<ul style="list-style-type: none">• Must be available the system at 24 hours in a day.• Must be updated the system regularly.
Stakeholders	Landlord, Renters and System Designer

2.3.3. Robustness & Fault Tolerance Requirement:

DR-02	The system manages over access
Description	Multiple users can access the system. this The system can be used by multiple users.
Stakeholders	N/A

2.3.4. Safety-Critical requirements:

There is nothing specific Safety-Critical requirements.

2.4. Maintainability and Supportability:

2.4.1. Maintenance Requirements:

MS-01	The system can support to browse this site in any time.
Description	Landlords can visit this page to manage their homes anytime. You can also manage renter.
Stakeholders	Landlord, Renters

2.4.2. Supportability requirements:

There is nothing specific supportability requirements.

2.4.3. Adaptability requirements:

There is nothing specific adaptability requirements.

2.5. Security requirements:

This system has some security requirements. Like-

- Username/Password
- Validation
- Authentication

2.5.1. Access Requirements:

To get access the whole module the system provides validation and authentication process.

SR-01	This system gives security policy.
Description	Without registering renter and landlords in this system, Unable to access system. mechanism Provided security services.
Stakeholders	Landlord, Renters and Super admin

2.5.2. Integrity Requirements:

All usernames and passwords are saved in encrypted shape to shield your data. Cracking usernames and passwords isn't easy.

2.5.3. Privacy Requirements:

This machine has numerous statistics safety strategies. Like verification and authentication. All statistics is saved in a database. All usernames and passwords are saved encrypted. It's tough to decipher. This mechanism presents privateness to users.

SR-02	All your data is protected
Description	All data is stored in a database. All usernames and passwords Encrypt and save. It's hard to decipher.
Stakeholders	Landlord, Renters

2.6. Usability and Human Integrity requirements:

2.6.1. Ease of Use requirements:

This system is very easy for use and also understandable.

2.6.2. Understand-ability and Politeness requirements:

This system is very easy for understand and also usable.

2.6.3. Accessibility requirements:

There are no access requirements beside those that have been outlined in the below:

AR-1: Log in as a Admin

AR-2: Log in as an Landlord

AR-3: Log in as a Renter

2.6.4. Accessibility requirements:

Trainers of Home Soft will be handled it.

Style CSS	The look & feel must be controllable using a stylesheet
Description:	Styles for web-based user interface elements Defined via Cascading Style Sheets (CSS). For more information See Eligibility Criteria for style requirements.
Originator:	Null
Stakeholders:	• Users
Fit Criterion:	-I have a CSS file that defines all the appearance attributes - Layout stretches to actual screen size (no fixed width) -Style supports 2 column layout with sidebar on the left The style supports dropdown menus - The style supports internationalization (i.e. mandarin)

Chapter 3

System Analysis

Scope of the study

3.1. Use Case Diagram:

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Figure 3.1. Use Case Diagram

3.1. Use Case Description:

Open Account	Create account as an admin or user
Description	The user wants to use our system
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment • Renter of the apartment • Software Developer
Scope	Access our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment • Renter of the apartment • Software Developer
Trigger Type	Event Trigger
Triggered	The user wants to use our system
Input	<ul style="list-style-type: none"> • Email Address • Password
Precondition	Has a clearance from Home Soft Software firm
Main Success Scenario	<ul style="list-style-type: none"> • The user opens the registration page • The user enters the email address • The user enters the password • The user clicks the registration button • The system creates a new session for user
Alternative Scenario	Automatic forward to main dashboard
Success End Scenario	<ul style="list-style-type: none"> ➢ The user has access to the system ➢ The page that the user wanted to access prior to authentication is displayed (automatic forwarding) ➢ the system created a new (http-)session
Failed End Scenario	Registration not complete because of no clearance

Login	Login as an internal user
Description	The user wants to access our system
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment • Renter of the apartment • Software Developer
Scope	Access our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment • Renter of the apartment • Software Developer
Trigger Type	Event Trigger
Triggered	The user wants to access our system
Input	<ul style="list-style-type: none"> • Email Address • Password
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user opens the login page • The user enters the email address • The user enters the password • The user clicks the registration button

	<ul style="list-style-type: none"> • The system creates a new session for user
Alternative Scenario	Automatic forward to main dashboard
Success End Scenario	<ul style="list-style-type: none"> ➤ The user has access to the system ➤ The page that the user wanted to access prior to authentication is displayed (automatic forwarding) ➤ the system created a new (http-)session
Failed End Scenario	<ol style="list-style-type: none"> 1. Wrong username 2. Wrong Password

Create Dashboard	Create dashboard for each apartment
Description	The home owner creates his own dashboard for each apartment
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment
Scope	Create his own dashboard in our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment
Trigger Type	Event Trigger
Triggered	The user wants to create and manages his own dashboard
Input	<ul style="list-style-type: none"> • Apartment name • Total Flat in apartment • Apartment Location • Apartment Image
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user creates dashboard • The user opens dashboard
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> ➤ The user has entered the to the system ➤ The system creates a new database for user
Failed End Scenario	Wrong input

Select Dashboard	Select dashboard for specific apartment
Description	The landlord selects his own dashboard for specific Apartment.
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment
Scope	Manage his own dashboard in our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment • Software Developer
Trigger Type	Event Trigger
Triggered	The user wants to manages his own dashboard
Input	<ul style="list-style-type: none"> • Rent • Miscellaneous
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user creates dashboard • The user opens dashboard

	<ul style="list-style-type: none"> • The user enters the estimate cost • The user enters the miscellaneous • The user clicks the save button • The system creates a new database for user
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> ‣ The user has entered the to the system ‣ The system creates a new database for user
Failed End Scenario	Wrong input

Create Renter	Create Renter for manage them
Description	The landlord can create renter for manage them
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment
Scope	Create renter through our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment
Trigger Type	Event Trigger
Triggered	The user wants to create tenant
Input	<ul style="list-style-type: none"> • Renter Name • Permanent Address • Contact Number • NID Number • Country

	<ul style="list-style-type: none"> • Religion Status • Education • Job Title • Company Name & Location • Father Name • Date of birth
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user opens renter add option • The user enters renter name • The user enters renter permanent address • The user enters renter contact number • The user enters renter NID number • The user enters renter country • The user enters renter religious status • The user enters renter education qualification • The user enters renter job qualification • The user enters renter job company name • The user enters renter father name • The user enters renter martials status • The user enters renter date of birth • The user clicks the save button
Alternative Scenario	Null
Success End Scenario	> The user has added renter member
Failed End Scenario	Wrong input

Renter Dashboard	Every renter has his own dashboard
Description	The renter views his own dashboard
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment • Renter of the apartment
Scope	View dashboard in our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment • Software Developer • Renter of the apartment
Trigger Type	Event Trigger
Triggered	The user wants to view his own dashboard
Input	Null
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user view dashboard
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> ➤ The user has entered the to the system ➤ The system creates a new database for user
Failed End Scenario	Null

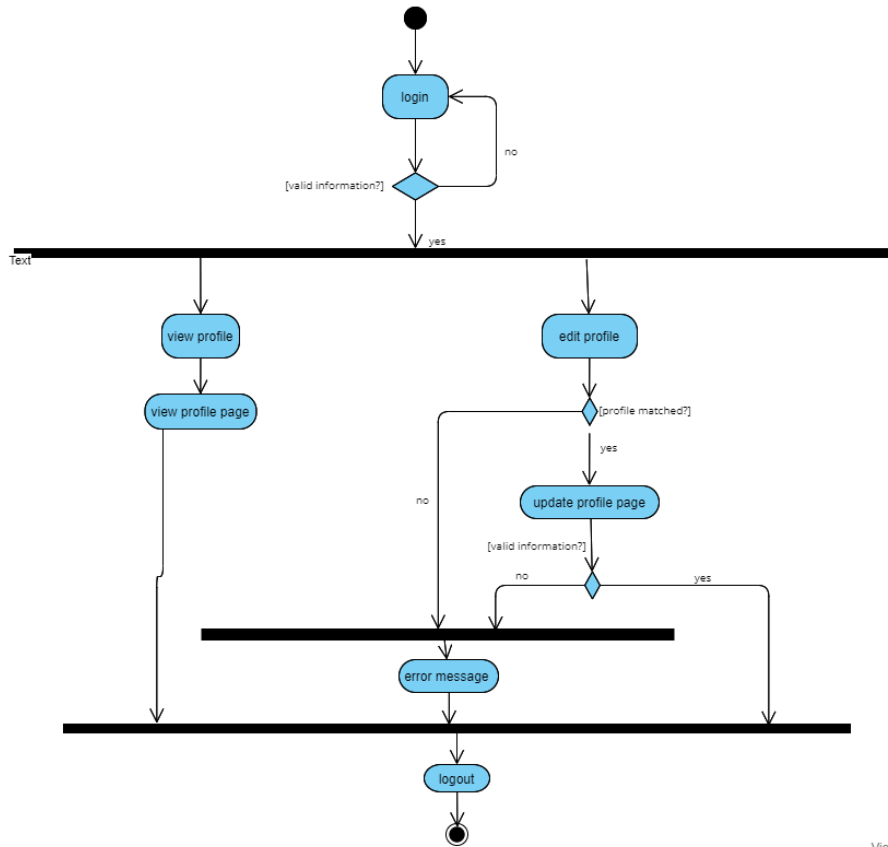
Send Message	Send message to each other
Description	The user can message to each other
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment • Renter of the apartment
Scope	Message send through our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment • Tenant of the apartment
Trigger Type	Event Trigger
Triggered	The user wants send message
Input	<ul style="list-style-type: none"> • Select user • Subject • Message
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user opens Message option • The user selects the particular user • The user enters the subject • The user enters the message • The user clicks the send button
Alternative Scenario	Null
Success End Scenario	➤ The user has sent the message to particular user
Failed End Scenario	Null

Report Generate	Report generates automatically daily basis input
Description	The tenant views monthly and yearly
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment
Scope	View report in our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment
Trigger Type	Event Trigger
Triggered	The user wants to view his report
Input	Null
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user view report
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> ➤ The user has entered the to the system ➤ The system generate report for user
Failed End Scenario	Null

Change Password	Change password of account
Description	Every user can change his account password
Originator	Null
Stakeholder	<ul style="list-style-type: none"> • Owner of the apartment • Tenant of the apartment • Super admin of the system
Scope	Change password through our system
Actor	<ul style="list-style-type: none"> • Owner of the apartment • Tenant of the apartment • Super admin of the system
Trigger Type	Event Trigger
Triggered	The user wants change password
Input	<ul style="list-style-type: none"> • New password two time
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> • The user opens change password option • The user enters new password two time
Alternative Scenario	Null
Success End Scenario	➤ The user has changed the password
Failed End Scenario	Wrong input

3.3. Activity Diagram:

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Figure 3.2. Activity Diagram for User

In this figure, we can see how a user login in to this system. If a user provides valid information, then a he can login to this system. If he failed to give valid information, system won't let him log in. After logging in, a user can check his profile information. If he wants to edit any information, he can also do that. After editing profile information, he can view that updated profile information page. If he didn't submit required information, system will show error message. And a user can also logout from that system.

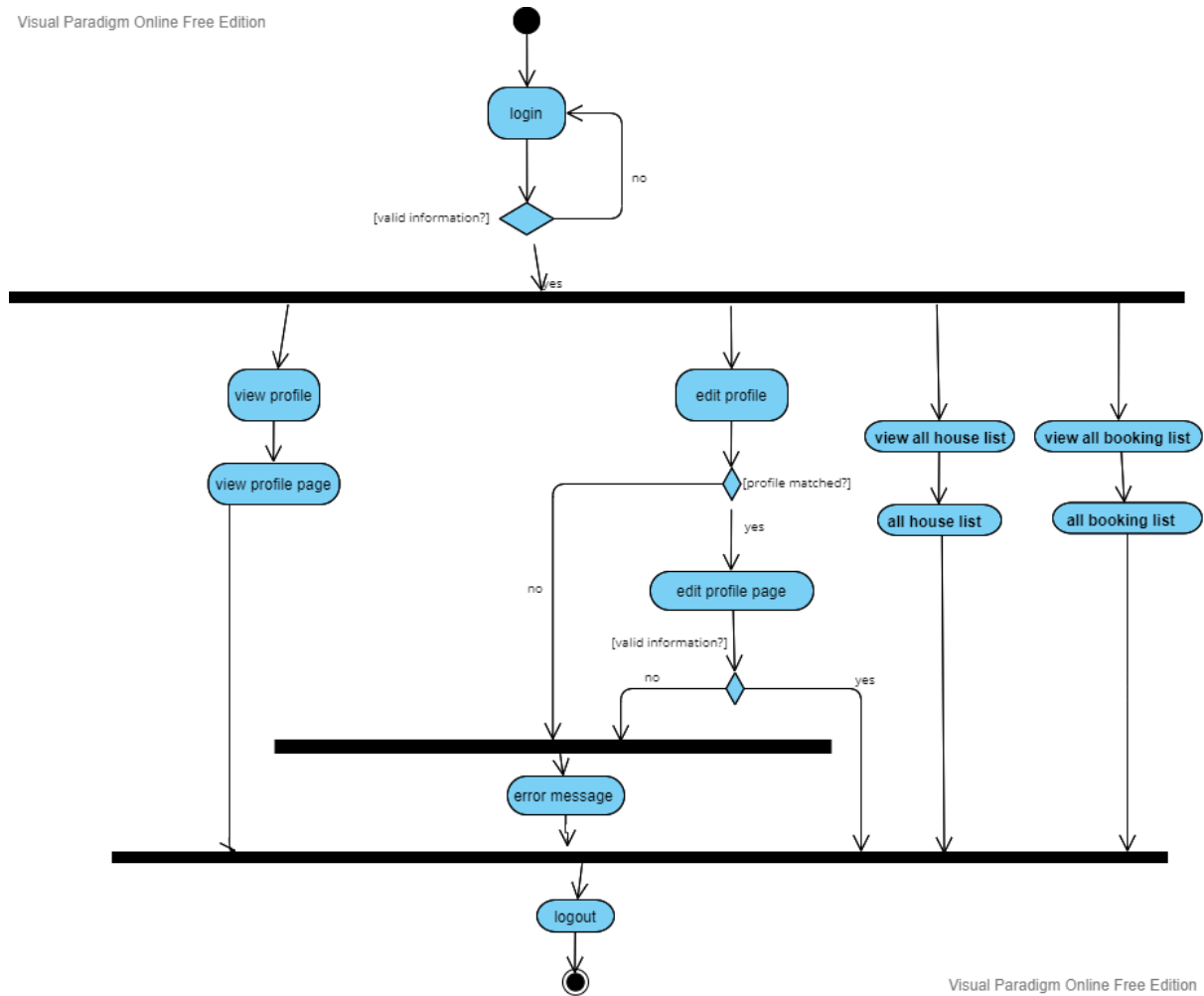


Figure 3.3. Activity Diagram for Admin

In this figure, we can see how a project manager (admin) manage this system. At first, admin log in to this system with valid information. He can also view and edit his own profile. He can also view all house list, which are added by landlords. Admin can also add houses. He can view all booking list which are added by renters. And he can also add booking in the booking list. He can check all information those are added by landlord and renters (tenants).

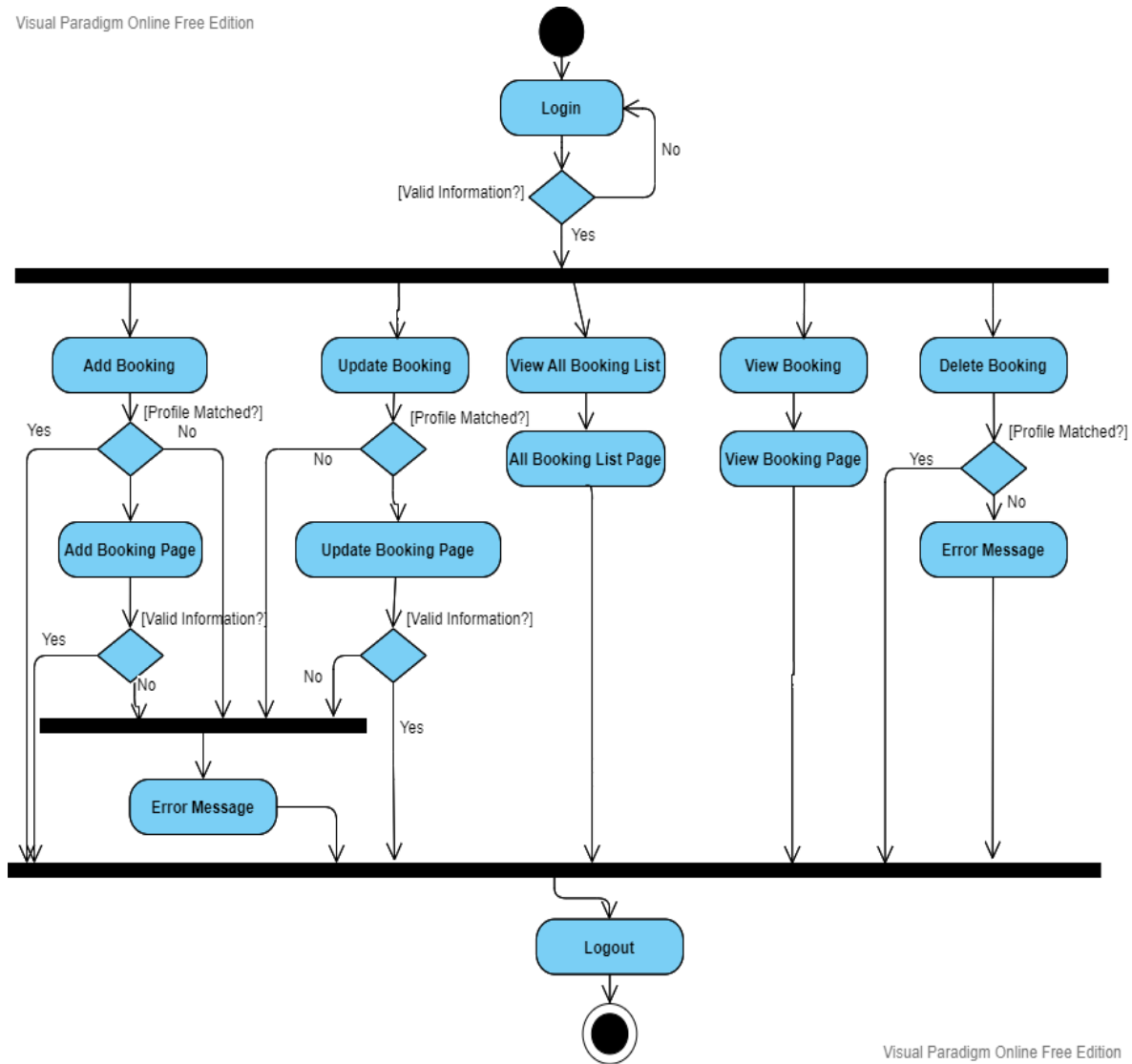


Figure 3.4. Activity Diagram for booking system (renter)

This diagram is for booking system. Renters (tenants) can add booking here. They check all house information and they can book houses after checking all provided information. They can view their booking. Renters can also delete their booking if needed. If they failed to give any valid information, system will show error message. Then the system will record house, renter information and the system will manage this booking system. This information can be viewed by admin and landlord, if renter confirm booking.

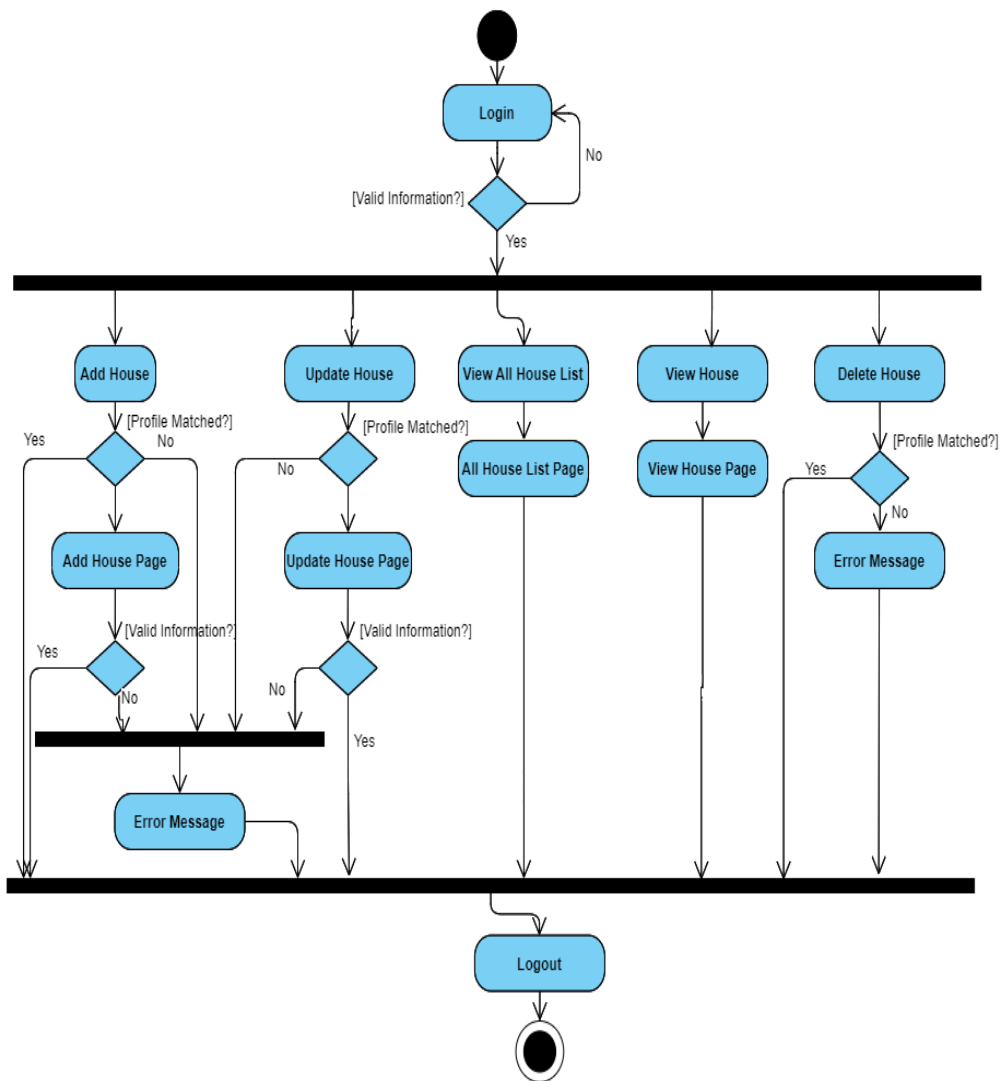


Figure 3.5. Activity Diagram for admin (view house)

This diagram is for admin. After login to this system, admin can add house. Then he can add house page. He can check all houses which are added by landlords. After that admin can update house information (if needed).he can check all house list after updating house information. Admin can also delete house, if landlord provides any unusual information.

3.4. System Sequence Diagram:

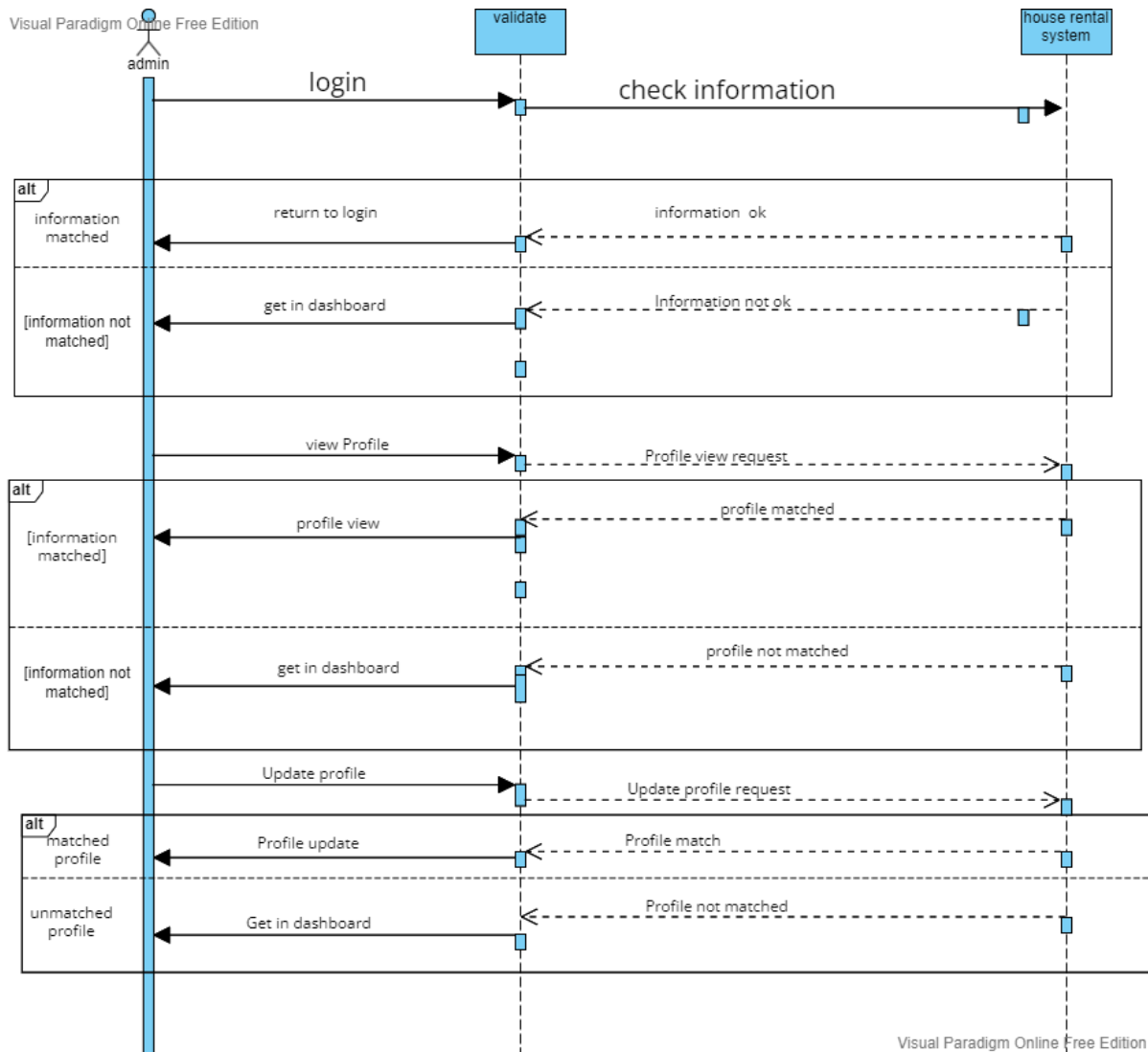


Figure 3.6. System Sequence diagram for user

This figure showing the sequence diagram for user (renter). At first he needs to login to the system. Then system will check information. If the information is valid, then system will get into dashboard. If the information isn't valid, then system will return to login page. Then user can request for viewing profile. System will check information. Then get into profile page. If user request for updating profile, then system will get in update profile page. System will check information of the profile matched or not.

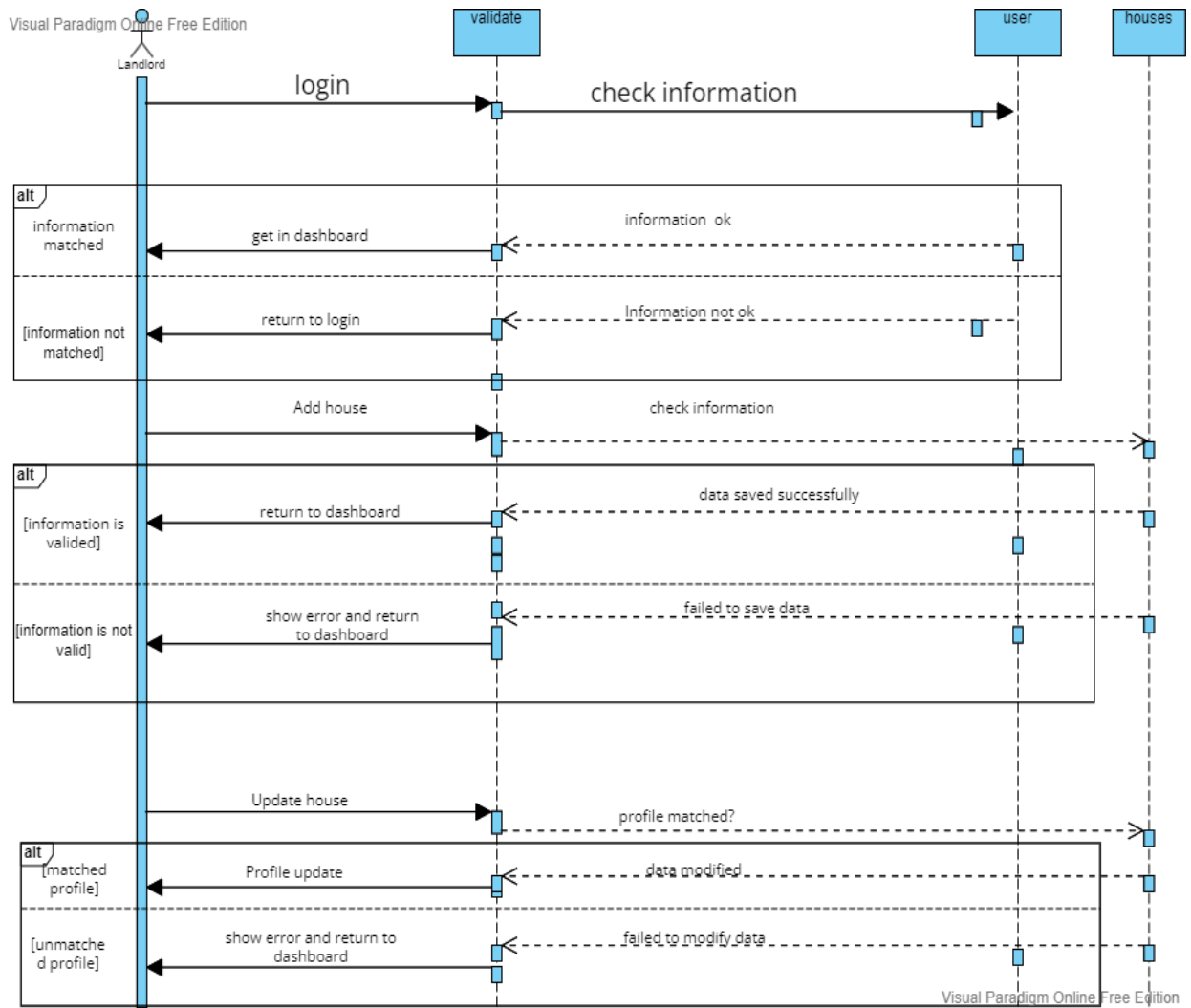


Figure 3.7. System Sequence diagram for Landlord

This figure shows, sequence diagram for landlord. At first a landlord needs to log in to this system. System will validate all information. Then a landlord will get into add house page. He needs to add all type of information of house here. System will check all those information and store those data in database. If a landlord wants to update house information, he can change house information here. System will modify all those information. Then system will return to dashboard.

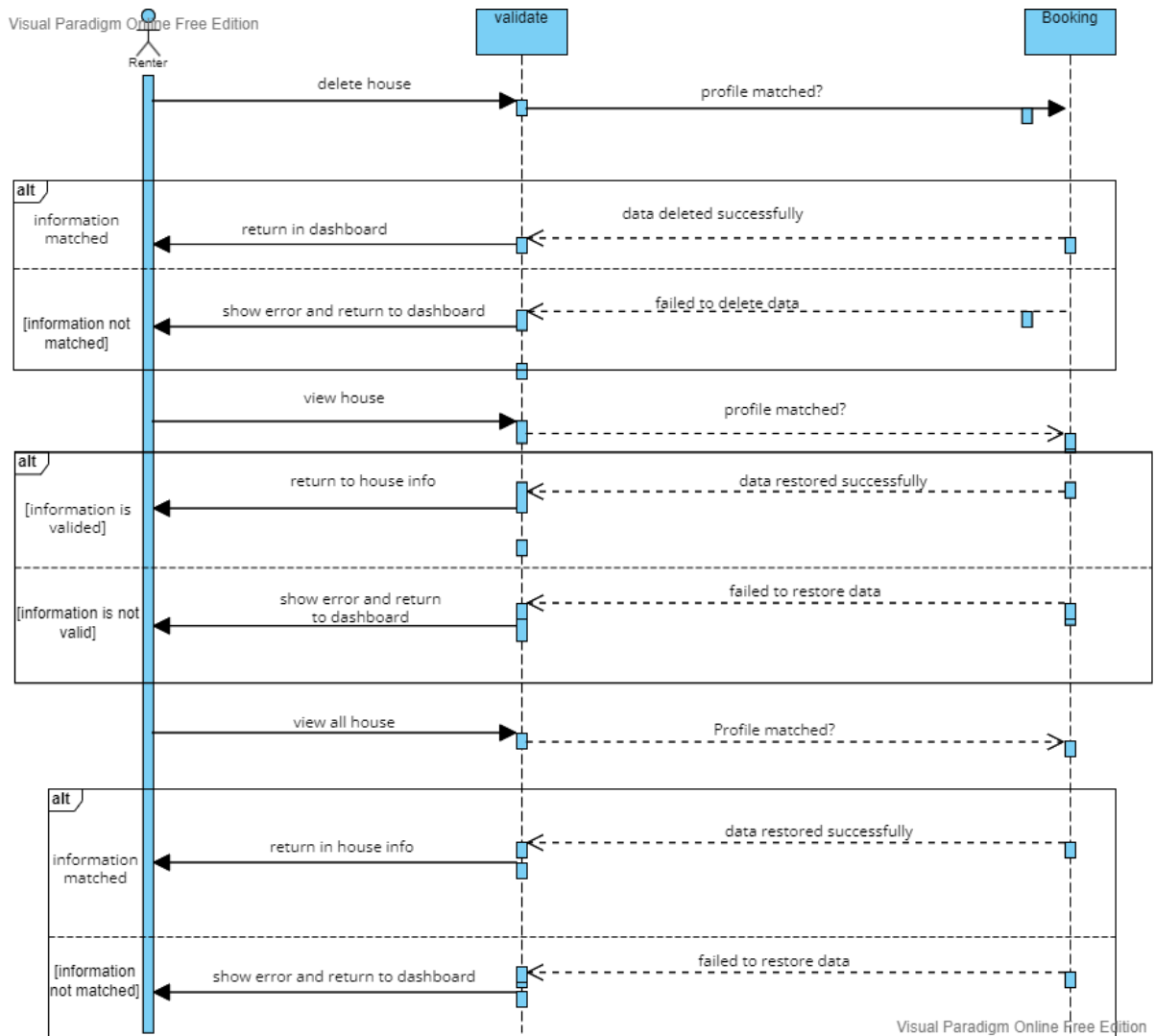


Figure 3.8. System Sequence diagram for Landlord 1

In this section landlord can delete house information, if the house is booked or have any problem with it. If he delete house information, then system will delete all data from database and return to landlord dashboard. System will restore all data. Then landlord can only view all house, which are added by other landlords.

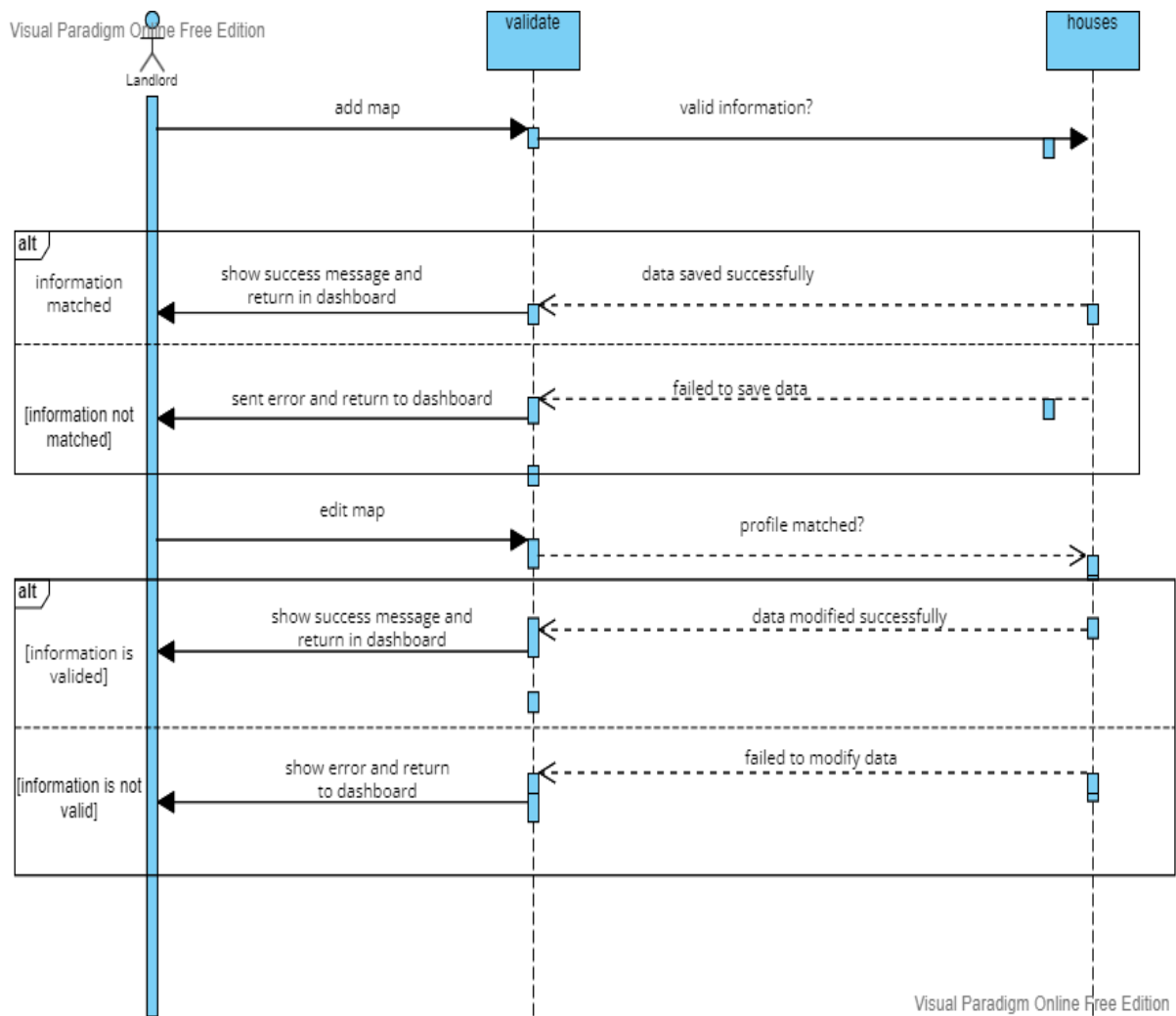


Figure 3.9. System Sequence diagram for Landlord 2

In this figure, we can see how landlord can add map details of his house. First he needs to get in house information page. Then he can add map of that house. System will validate all information here and save those data. System will show success message and return to the dashboard. Landlord can also edit house map location if he wants. System will modify map data and sent success message. It will return to dashboard. If landlord fail to give valid information, then system will fail to modify data and return to dashboard.

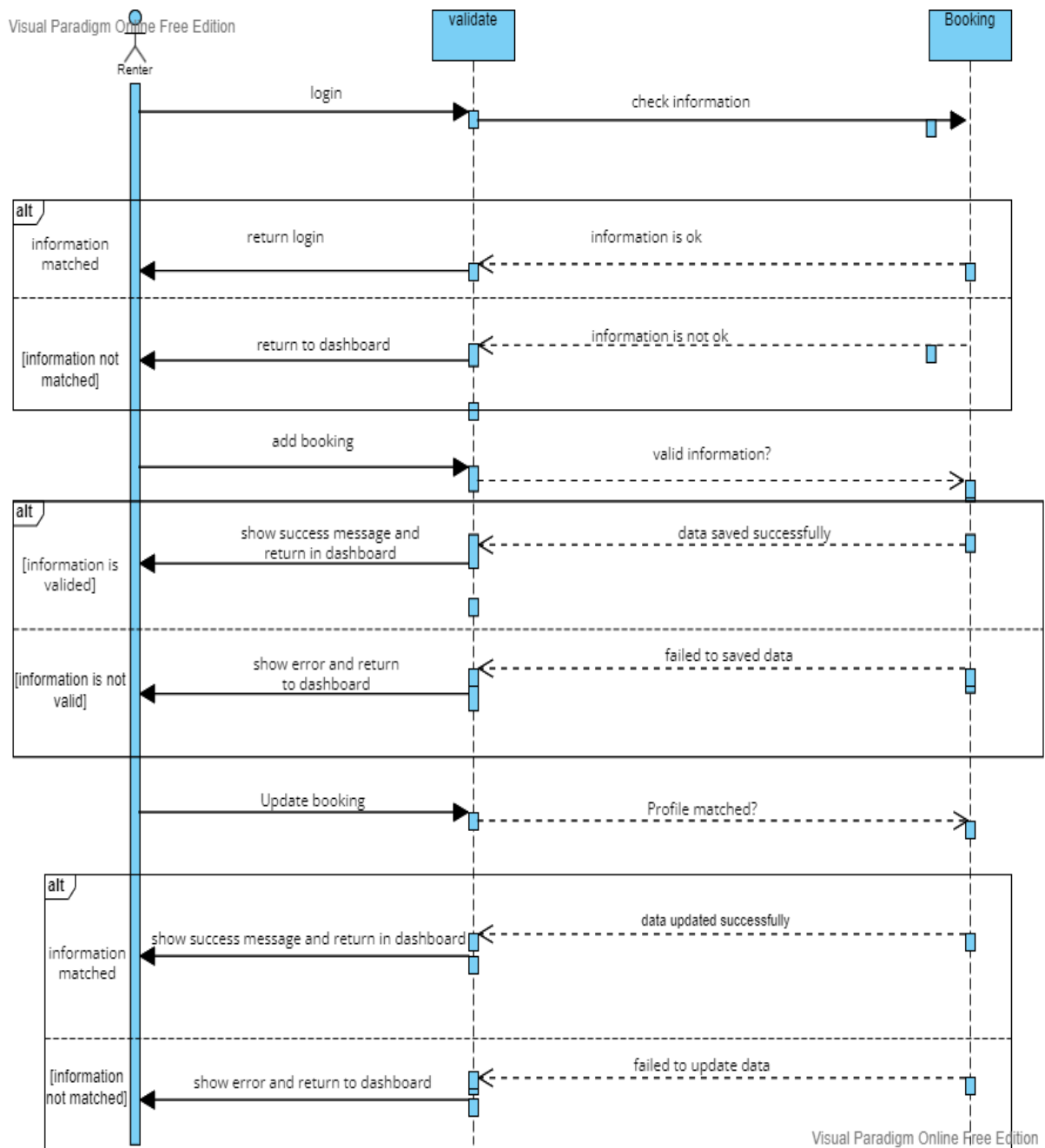


Figure 3.10. System Sequence diagram for add booking (renter)

This figure is for booking system. Renter (tenant) will login to this system first. System will validate renter's all information. Then he can check all house information. After that, renter can add booking of a house. System will save those data and return to dashboard. If renter provides all valid information. If he fails to give valid information, then system will sent error message and return to dashboard. Renter can also update their booking. If he update a booking with valid information then system will update data and show success message. If renter failed to provide valid information, then system will fail to update data. It will show error and return to dashboard.

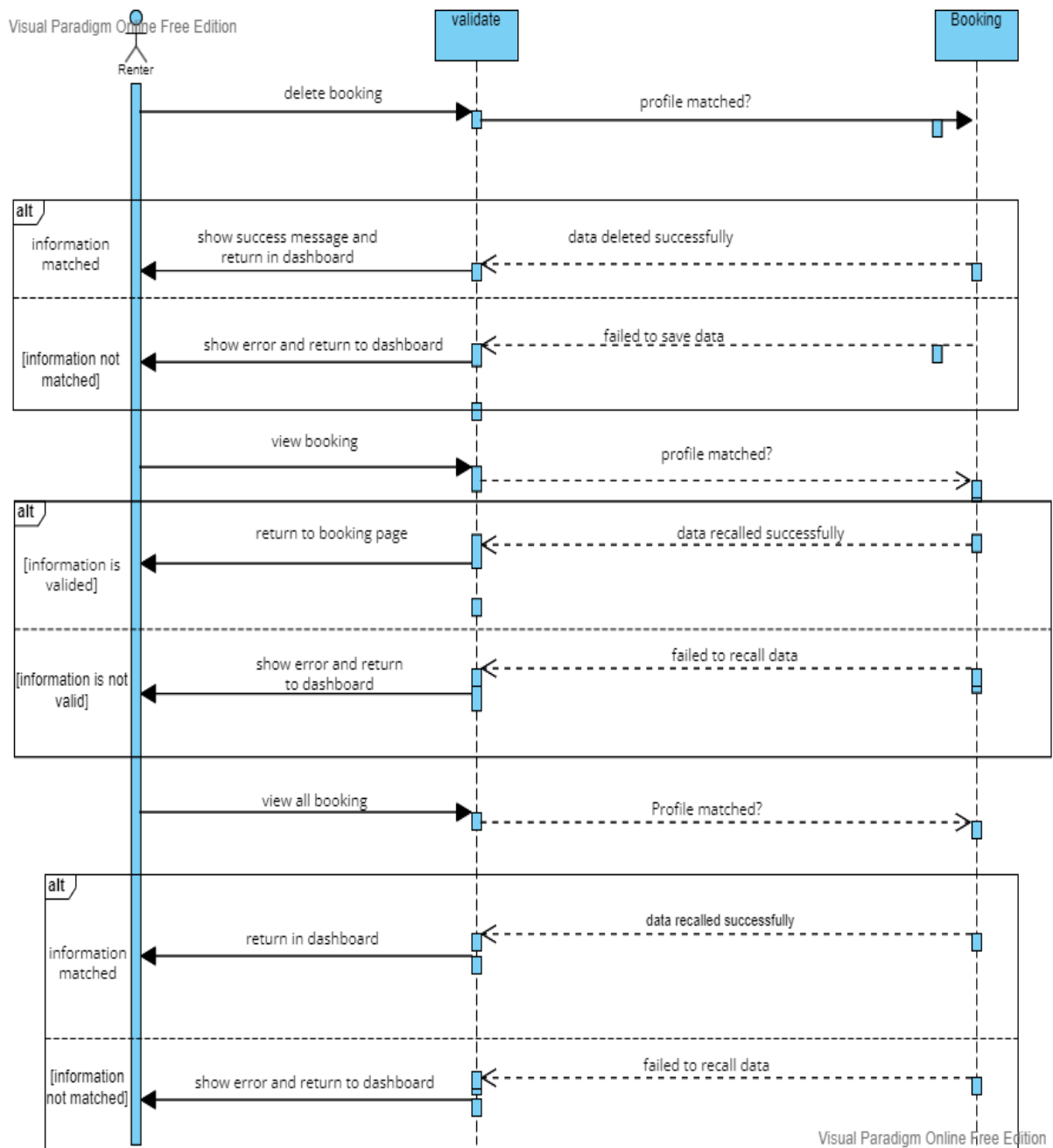


Figure 3.11. System Sequence diagram for delete booking (renter)

In this diagram, renter can delete their booking if wanted. If a renter delete booking, system will delete all booking data. It will show success message and return to renter dashboard. Then renter can view all houses and bookings. System will recalled all data and return to booking page. If he failed to provide valid information, system will show error and return to dashboard. Renter can view all booking. System will recall data and show all booking information. Again, he needs to provide valid information. If information didn't match, then system will fail to recall data. System will show error and return to dashboard.

Chapter 4

System Design Specification

System Design Specification:

4.1. Sequence Diagram:

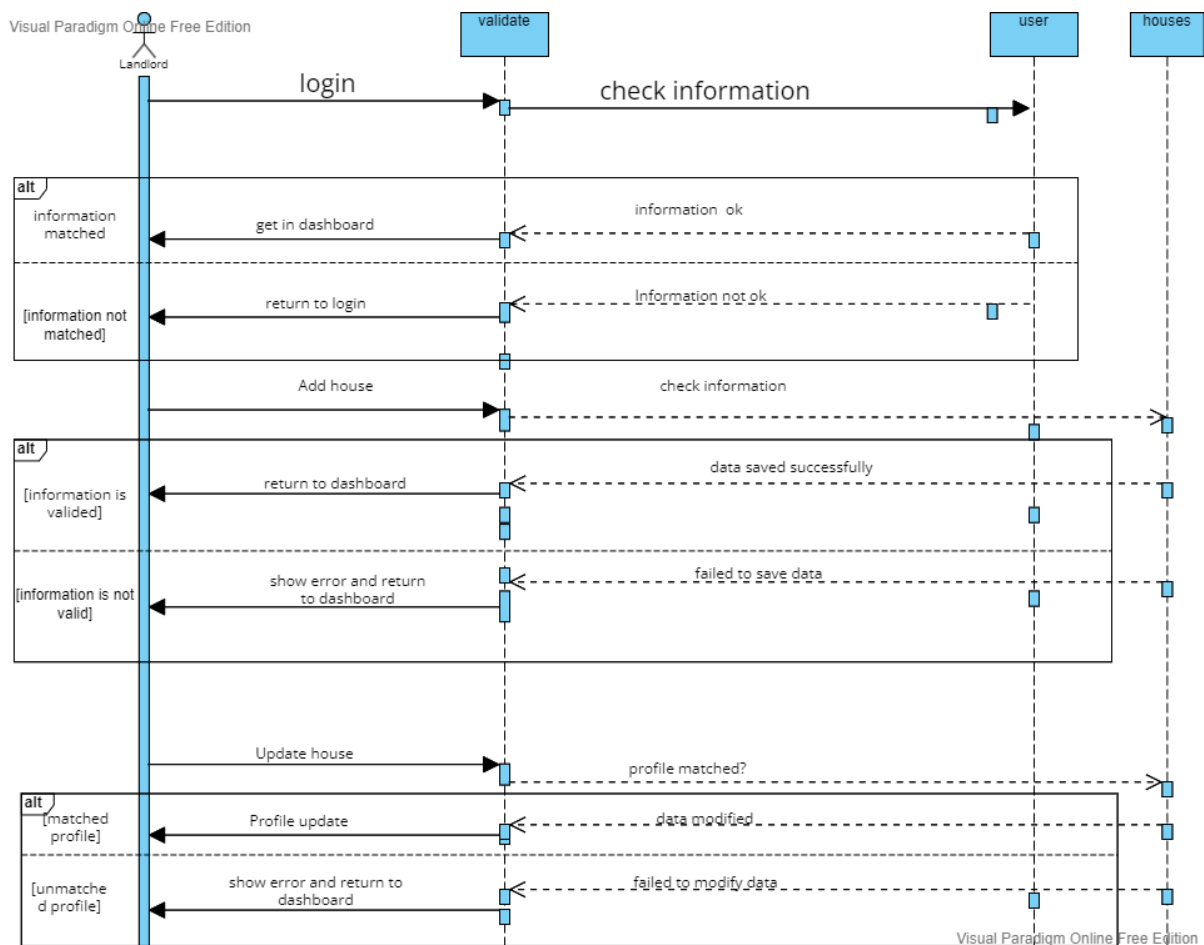


Figure 4.1. System Sequence diagram for Landlord

Login	Landlord will login to this system
Description	User wants to access to the system
Stakeholder	<ul style="list-style-type: none"> Landlord Renter
Precondition	Has an account
Main success scenario	<ul style="list-style-type: none"> The user opens registration page User add house information User update house information System creates new session for user(Landlord)
Alternative scenario	Automatic forward to main dashboard
Success end scenario	<ul style="list-style-type: none"> User has access to the system User can provide house information System created new session
Failed end scenario	Sent error message and return to dashboard.

4.2. Class Diagram:

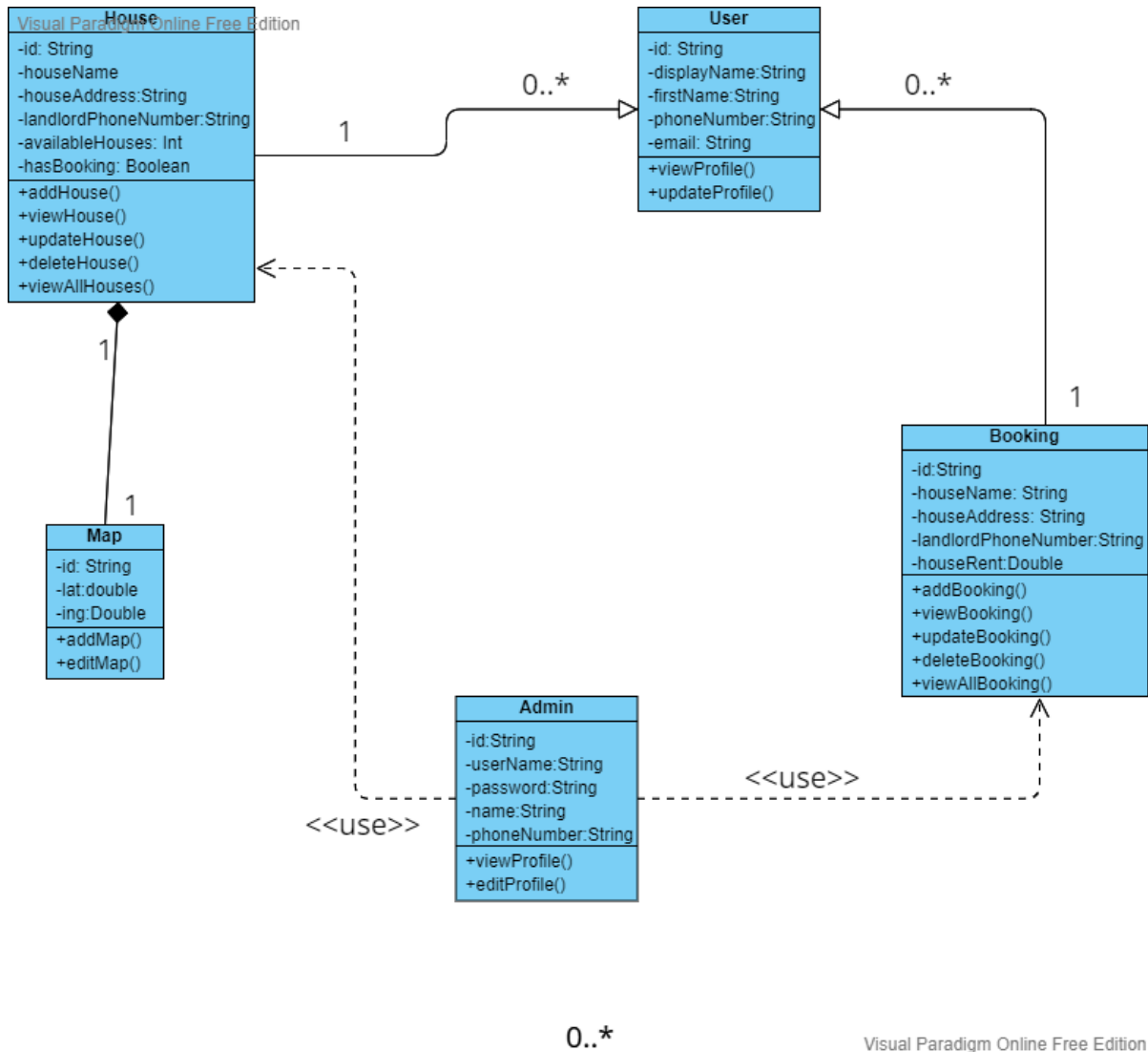


Figure 4.2. Class Diagram for house rental system

4.3. Development Tools & Technology:

4.3.1. User Interface Technology:

- Laravel Framework
- Java Script
- Bootstrap ,html and CSS

4.3.2. Implementation Tools & Platforms:

- Microsoft Visual Studio
- PHP
- MySQL Server used as Database
- Apache HTTP Server as local server.

Chapter 5

System Testing

System Testing

5.1. Testing Feature:

Unit testing is a software engineering technique. Now test the individual code units. I actually used this method after completing a project. A bug was found in the system. Then I try to solve it.

Acceptance testing is used before system release. I actually found a small bug using this method. Then I try to solve it. After the system error is resolved, it will be released and used.

5.1.1. Features To Be Tested:

- Request For Account Open
- Login
- Log Out
- Create Dashboard
- Add Renter
- Manage Renter
- Manage Dashboard
- Add Notice
- Profile

5.1.2. Features Not to Be Tested:

- Report
- Renter Dashboard
- Renter Login

5.2. Testing Strategies:

5.2.1. Test Approach:

Test Case 01:

5.2.1.1. Request for account, Login & Logout:

Test case id	01
Purpose	To login to this system for all type of user
Test data	<ul style="list-style-type: none">• Name• Email• Password• Confirm password• Phone Number• Add map
Steps	<ul style="list-style-type: none">• Browse the website• Enter required information for the register• Click sign up button
Status	Pass

Test Case 02:

5.2.1.2. Add Dashboard:

Test case id	02
Purpose	Add dashboard for each user
Action	Collect required information and create dashboard for users
Status	pass

Test Case 03:

5.2.1.3. Add & Manage Renter:

Test case id	03
Purpose	Adding and managing renter
Test data	<ul style="list-style-type: none">• Renter information• Update profile information• Delete profile information
Steps	<ul style="list-style-type: none">• Browse website• Enter valid information• Click submit button
Status	Pass

Test Case 04:

5.2.1.6. Manage booking:

Test case	04
Purpose	Manage all booking by renters
Test data	<ul style="list-style-type: none">• Check house details• Check house rent price• Check house location• Check landlord information
Steps	<ul style="list-style-type: none">• Browse website• Book a house• Click submit button for booking
Status	pass

5.2.4. Traceability Matrix:

S n o.	Re q. Id	Req. Description	Scenario	TC Id	TC Descriptio n	Test Resu lt	Defe ct Id	Defe ct Stat Us
01	BR 1.0	1.Login	1. Account Open	TC_house_rental_account_open_01	Request to super admin for create account	Pass	N/A	N/A
			2. Login	TC_house_rental_account_open_01	Login into system	Pass	N/A	N/A
			3. Logout	TC_house_rental_account_open_01	Logout from system	Pass	N/A	N/A
02	BR 2.0	2.Dashboard	1. Add Dashboard	TC_house_rental_add_dashboard_02	Create Dashboard	Pass	N/A	N/A
			2. Go To Dashboard	TC_house_rental_go_to_dashboard_02	Go To Dashboard	Pass	N/A	N/A
03	BR 4.0	4. Renter	1. Add Renter	TC_house_rental_add_renter_04	Add Renter to system	Pass	N/A	N/A
			2. View Renter Details	TC_house_rental_view_renter_04	View Added Renter	Pass	N/A	N/A
			3. Edit Renter	TC_house_rental_edit_renter_04	Edit Renter Details	Pass	N/A	N/A
			4. Delete Renter	TC_house_rental_delete_renter_04	Delete Specific Renter	Pass	N/A	N/A
04	BR 6.0	6.Manage Dashboard	1. Add Rent	TC_house_rental_add_rent_renter_06	Add Renter Rent	Pass	N/A	N/A
05	BR 7.0	7.Notice	1. Notice	TC_house_rental_add_notice_07	Add Notice to system	Pass	N/A	N/A
			2. Leave Notice	TC_house_rental_add_leave_notice_07	Add Leave Notice To system	Pass	N/A	N/A
06	BR 8.0	8. Profile	1. View	TC_house_rental_view_08	View Profile	Pass	N/A	N/A

5.3. Testing Environment:

5.3.1. Web

5.3.1.1. Windows 10

- Chrome
- Firefox

5.3.1. Mobile Browser

5.3.1.2. Android 13

- Chrome
- Google
- Firefox

5.3.1. Hardware Requirements:

Hardware Requirements		
Processor	RAM	Hard Disk Space
Dual-Core or higher	4gb or higher	100gb or higher
Software Requirements		
Operating System		
For user no specific OS is required.		

Chapter 6

User Manual

6. User Manual:

6.1. Landing Page:

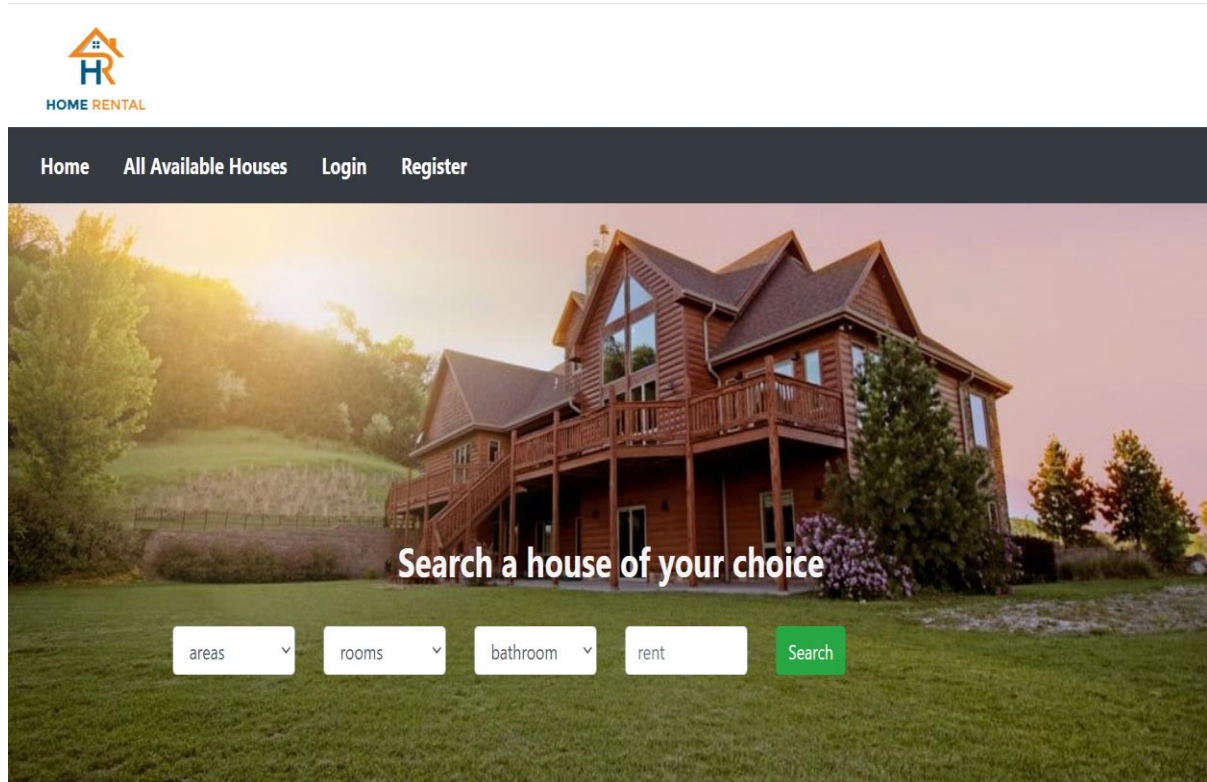


Figure 6.1. home page for house rental system

This landing pages. Where registered user can search house by area, rooms, bathrooms, rent. Here they also see tutorial and about our company.

6.2. Register Request:

The image shows a web browser window displaying the registration page for a house rental system. The page has a dark blue header with navigation links: Home, All Available Houses, Login, and Register. The main content area features a light blue background with a mountain landscape. A dark grey registration form is centered on the page, titled 'Register'. The form contains the following fields: 'Enter your name' (text input), 'asad' (text input), 'select a role' (dropdown menu), 'Nid number' (text input), 'contact (please add 88 before number)' (text input), 'email' (text input), a password field (text input with dots), and 'confirm password' (text input). A blue 'Register' button is at the bottom of the form. The footer is dark blue and contains three columns: 'About Us' with placeholder text, 'Contact Us' with location, phone, and email, and 'Find Us On Social Media' with icons for Facebook, Twitter, Instagram, and LinkedIn. The copyright notice '©House Rental Website' is at the bottom center.

Figure 6.2. Registration page for house rental system

Unregistered user request to here to registered them.

6.3. Login page:

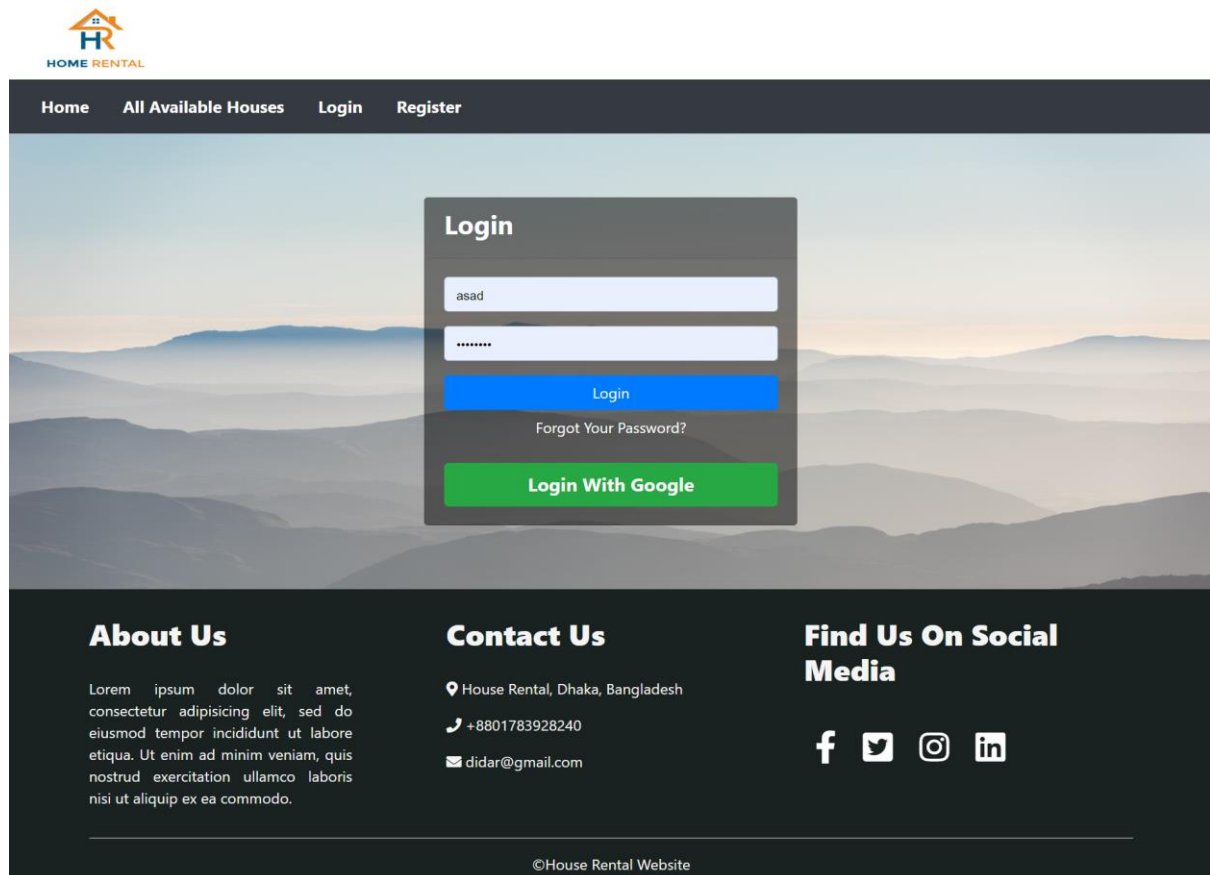


Figure 6.3. Login page for house rental system

Admin get request who has want to use our software. He / She can approve or decline them.

6.4. Dashboard for Landlord:

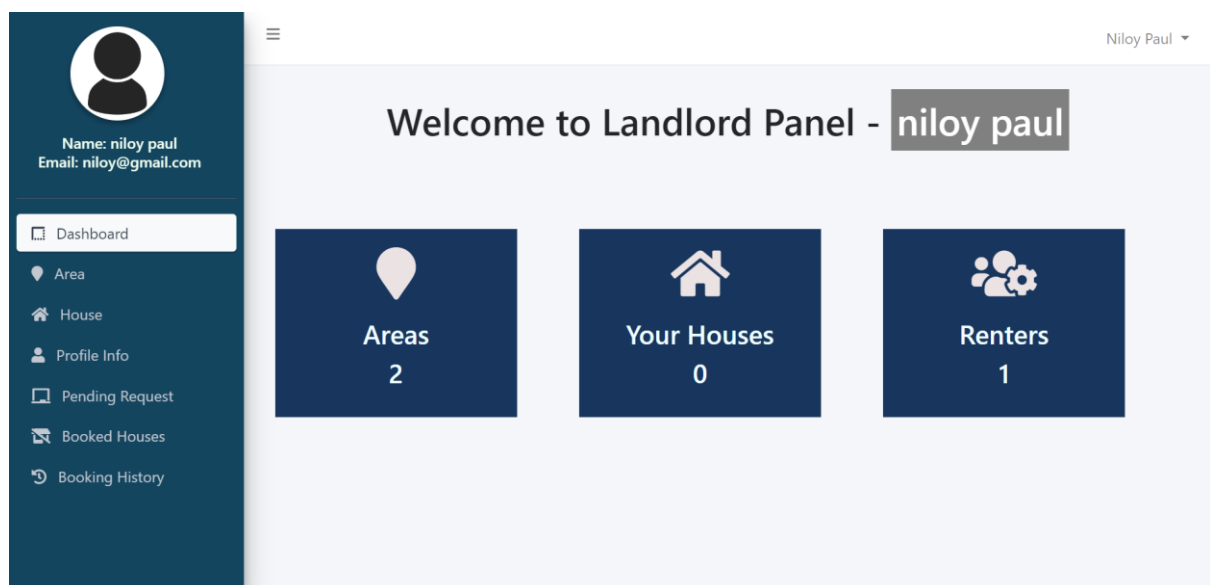


Figure 6.4. Dashboard for landlord

After login as landlord, they can view this page. They can add/edit/update/delete houses.

6.5. Landlord profile information:

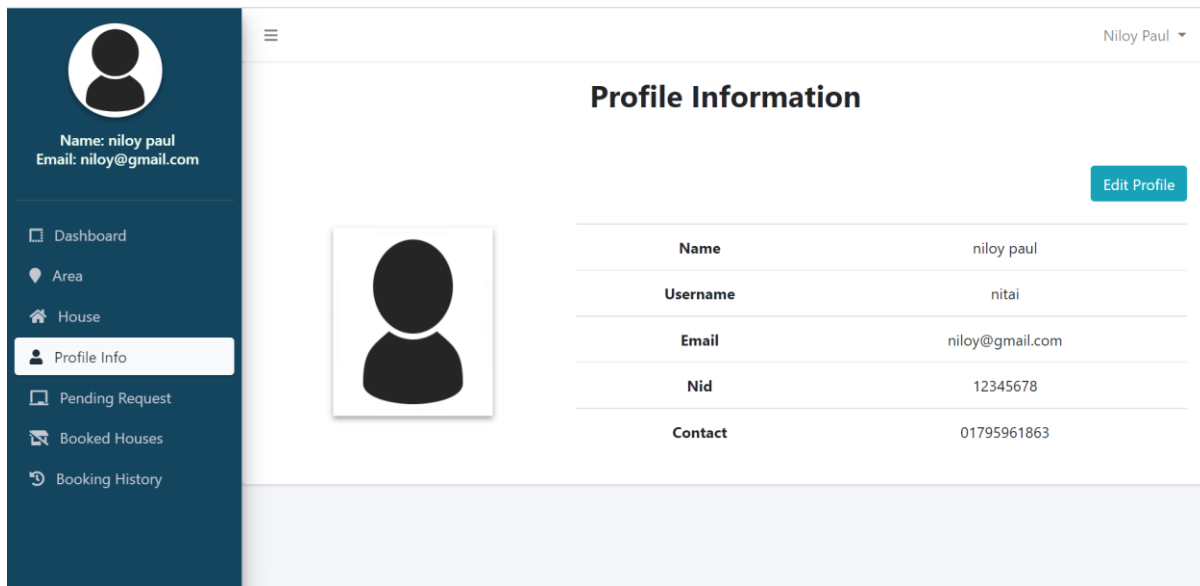


Figure 6.5. Landlord profile information

They can view and edit their own information here.

6.6. Dashboard for Rental panel:

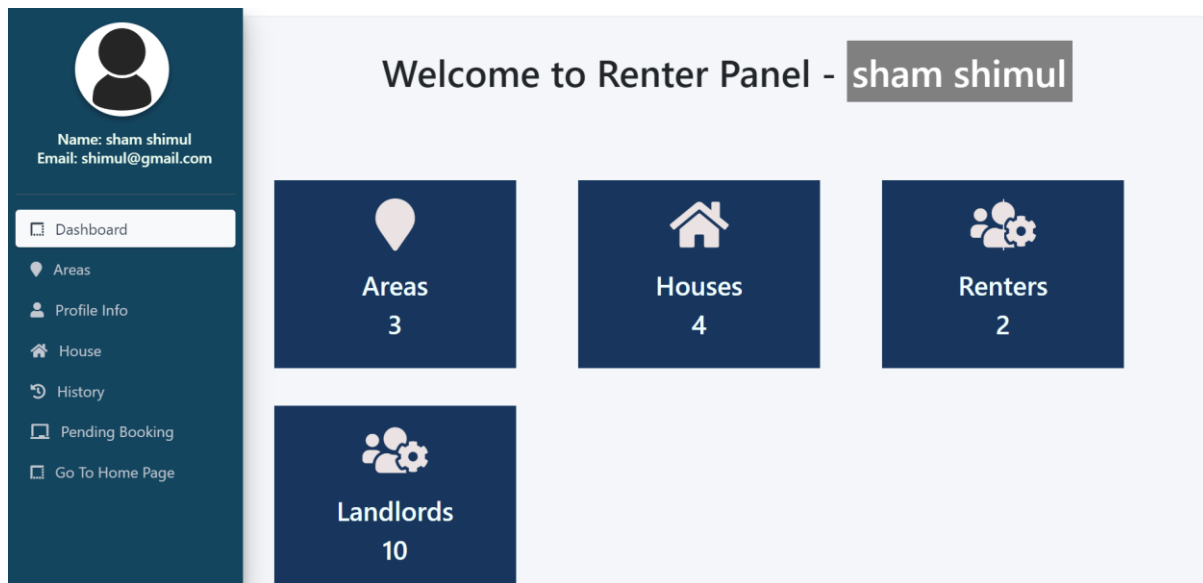
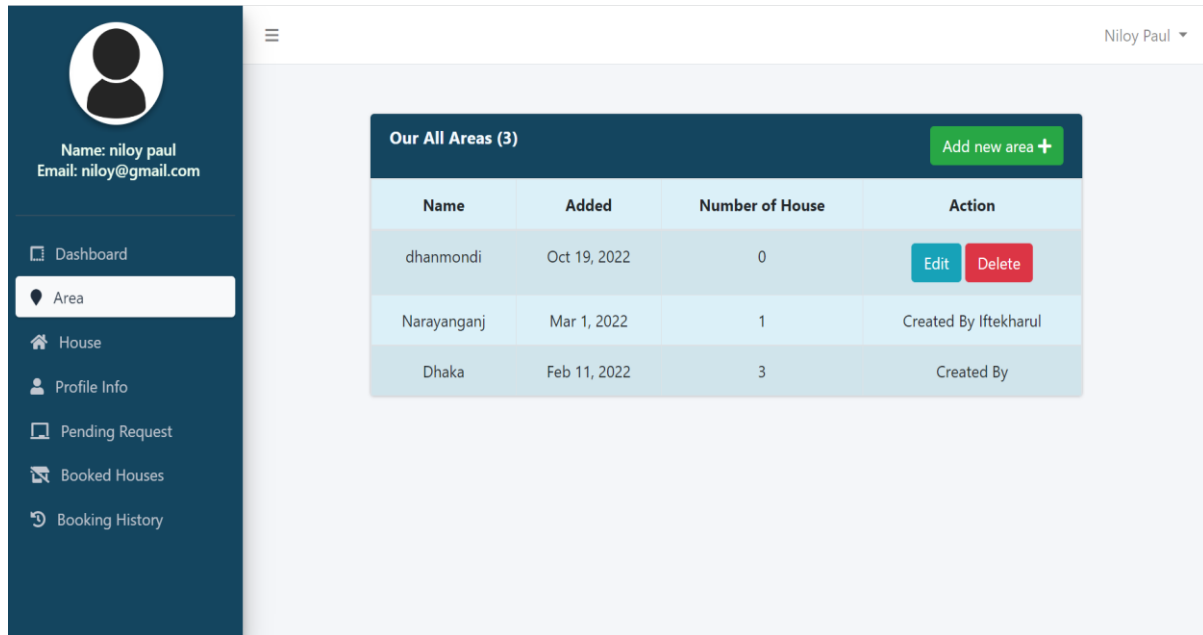


Figure 6.6. Dashboard for rental panel

After login as renter, user can view this page. Then he can view all areas, houses and prices from here.

6.7. View areas:



The screenshot shows a user dashboard for 'Niloy Paul'. The sidebar on the left contains navigation links: Dashboard, Area (selected), House, Profile Info, Pending Request, Booked Houses, and Booking History. The main content area displays 'Our All Areas (3)' with an 'Add new area +' button. Below this is a table with the following data:

Name	Added	Number of House	Action
dhanmondi	Oct 19, 2022	0	Edit Delete
Narayanganj	Mar 1, 2022	1	Created By Iftekharul
Dhaka	Feb 11, 2022	3	Created By

Figure 6.7. Dashboard for all house areas

Here, user (renter) can view all house detail information. When user search a house by area, system will show all houses by location. They can also see when that house was added in this system. Then, they can check all those details like how much houses are available, how much rooms and washrooms for a house. They can check landlord's information too. And system will also show, that house is booked or not. They can also check booking history of each houses.

Chapter 7

Project Summary

7.1. Critical Evolution:

➤ Results and initial goals

Our tasks meet their meant desires and meet suitable first-class standards

➤ Goal achieved

Our venture efficaciously tactics all records from customers and generates user-pleasant dashboards.

➤ Human resources used in the project

We have a well-developed training system. If you need training in the future, we will arrange this for you.

➤ Identifying problems from ongoing projects

We will continuously monitor it and develop further if necessary.

➤ Further project work

We will introduce some more features in the future.

7.3. Limitation:

➤ Overall Aim

Our goal was to develop management software that would improve communication between landlords and renter. Your time and money are precious. This system reduces communication costs between renter and owners. Some earn income by renting apartments or houses. Our system saves time and increases sales. Owners receive up-to-date reports on costs, duties and income. Our system builds a better communication system between owners and renters. Owners can send messages or attach notifications to all renter at once. No face-to-face communication required. A renter can issue an eviction permit to vacate the apartment. Renter can cause housing-related issues. Finally, our system is a management system that can bridge all communication gaps between renter and owners.

7.4. Obstacles and Achievements:

- User Interface
We develop all modules to be user friendly and mobile friendly. Almost all modules are responsive with the exception of a few modules. In the future, we will use React for interactive user interfaces.
- Scalability
We make full use of our computing power and bandwidth to create our projects. As the load increases, you can add servers to balance it. In the future, we will use Vue JS for better performance.
- Performance
We are so much concern about our project performance.
- Framework
We are using most common server site framework Laravel. Our project is API based.
- Security
Internet access is required for users to use the system. Use CSRF tokens to prevent CSRF attacks. The main thing is our system. If someone tries to hack our system, they can easily obtain information about users' private information. We will use stronger security methods in the future.

7.5. Future Scope:

- Automated Report
- Renter Dashboard
- Multi Authentication
- Message
- Renter Family Member Add
- NID Verification
- monthly/yearly payment
- annual cost

Chapter 8 Reference

8. References:

- 1. B property**
<https://www.bproperty.com>
- 2. Rents.com.bd**
<https://rents.com.bd>
- 3. Rental home bd**
<https://rentalhomebd.com>