



**Daffodil**  
*International*  
**University**

---

**Vara-varatia**

**Submitted By:**

**Kh. Afjal Hossen**

**ID: 151-35-1011**

**Department of Software Engineering  
Daffodil International University**

**Supervised By**

Mr. Md. Rittique Alam

Lecturer

Department of Software Engineering  
Faculty of Science & Information Technology  
Daffodil International University

This project report was submitted in accordance with the requirements for the Bachelor of Science in Software Engineering program.

## APPROVAL

This thesis titled on “**Vara-varatia**”, submitted by **Kh. Afjal Hossen (ID: 151-35-1011)** 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**



-----  
**Kaushik Sarker**  
**Associate Professor**  
Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Internal Examiner 1**



-----  
**Dr. Md. Fazla Elahe**  
**Assistant Professor**  
Department of Software Engineering  
Faculty of Science and Information Technology  
Daffodil International University

**Internal Examiner 2**



-----  
**Mohammad Abu Yousuf, PhD.**  
**Professor**  
Institute of Information Technology  
Jahangirnagar University

**External Examiner**

## DECLARATION

I hereby declare that I have taken this project under the supervision of **Md Rittique Alam**,  
**Lecturer, Department of Software Engineering, Daffodil International University**



.....  
**Kh. Afjal Hossen**  
ID: 151-35-1011  
Batch: 16th  
Department of Software Engineering  
Faculty of Science & Information Technology  
Daffodil International University

**Certified By:**



.....  
**Md Rittique Alam**  
Lecturer  
Department of Software Engineering  
Faculty of Science & Information Technology  
Daffodil International University

## **Acknowledgment**

At first, I would like to thank almighty Allah. However, 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 **Md Rittique Alam** 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.

## Abstract

Vara-varatia is a web application management system where home owner / apartment owner / home builder makes better communication with tenant. This project is intended to make the home related management system easier to the home owner. A huge number of tenant and landlord keep unnecessary document every transaction detail between them. And, End of story missing of some hard document they cannot estimate actual profit or expenses. In which, both of them suffering long. This system makes simple their every step. Every month building owner have to collect rent from tenant. That's not a big deal. It became more difficult when they calculate their rent or their dew. When an owner has more than one building, they have to remind it his memory or use note book that which tenant didn't pay this month rent. And it's disgusting to be owner that he has to remind his tenant dew for every month. Cause it cannot say surely every tenant paid his ren at due time. So, it's a huge amount for calculate and make an annual or monthly report. Some owner daily expense going based on this amount. Maximum owner maintains caretaker who maintains these criteria. And some of them make mismatch the transaction as intentionally or unintentionally. The average amount of such calculations per month becomes a good amount at the end of the year. After end of the day, there is no clear report of every transaction. Other side of tenant, before leaving home he has make appointment with owner and make payment clearance. Old process is a lengthy process. Cause before leaving home, he has to confirm his owner minimum one month ago. Because of communication gap, is there any need of repair of home. it's still remained as a pending. The old management has been going through some irregularities from the beginning. Both the landlord and the tenant suffer with this old management system. Vara-varatia is simple account management web base system that can resolve maximum issues.

## Table Content

<b>Contents</b>	<b>Page</b>
Approval	II
Declaration	III
Acknowledgement	IV
Dedication	V
Abstract	VI
<b>List of Figure</b>	<b>VIII</b>
<b>Chapter</b>	
<b>Chapter 1: Introduction</b>	1
1.1. Project Overview	2
1.2. Project Purpose	3
1.2.1. Background	4
1.2.2. Benefit	4
1.2.3. Goals	4
1.3. Stakeholders	5
1.4. Proposed Model System	6
1.5. Project Schedule	7
1.5.1. Gantt Chart	7
1.5.2. Milestone	7
1.5.3. Work Distribution	7
<b>Chapter 2: Software Requirements Specification</b>	
2.1. Functional Requirements	9
2.2. Data Requirements	10
2.3. Performants Requirements	10
2.4. Dependability Requirements	11
2.5. Maintainability and Supportability Requirements	12
2.6. Security Requirements	12
2.7. Usability and Human Interaction Requirements	13
2.8. Look & Feel Requirements	13
2.9. Operational and Environmental Requirements	16
2.10. Legal Requirements	16
<b>Chapter 3: System Analysis</b>	
3.1. Use Case Diagram	18
3.2. Use Case Description	19
3.3. Activity Diagram	26
3.4. System Sequence Diagram	32
<b>Chapter 4: System Design Specification</b>	
4.1. Sequence Diagram	39
4.2. Entity Relationship Diagram	40
4.3. Development Tools & Technology	40
<b>Chapter 5: System Testing</b>	
5.1. Testing Feature	43
5.2. Testing Strategies	44
5.3. Testing Environment	48
<b>Chapter 6: User Manual</b>	
Chapter 6, Section 6.1 To 6.11	50

<b>Chapter 7: Project Summary</b>	
7.1. GitHub Link	59
7.2. Critical Evolution	59
7.3. Limitation	59
<b>Chapter 8: Reference</b>	
Reference	62

### List Of Figure

<b>Figures</b>	<b>Page</b>
Figure 1.1: Figure of Vara-varatia block diagram	6
Figure 1.2: Four and half month work distribution (140 day`s)	7
Figure 3.1. Use case	18
Figure 3.2. Activity Diagram for Registration	26
Figure 3.3. Activity Diagram for login	26
Figure 3.4. Activity Diagram for create dashboard	27
Figure 3.5. Activity Diagram for select dashboard	27
Figure 3.6. Activity Diagram for create cost list	28
Figure 3.7. Activity Diagram for create tenant	28
Figure 3.8. Activity Diagram for create notice	29
Figure 3.9. Activity Diagram for tenant dashboard	29
Figure 3.10. Activity Diagram for create leave notice	30
Figure 3.11. Activity Diagram for send message	30
Figure 3.12. Activity Diagram for report	31
Figure 3.13. Activity Diagram for change password	31
Figure 3.14. System Sequence diagram for registration	32
Figure 3.15. System Sequence diagram for login	32
Figure 3.16. System Sequence diagram for create dashboard	33
Figure 3.17. System Sequence diagram for select dashboard	33
Figure 3.18. System Sequence diagram for create cost list	34
Figure 3.19. System Sequence diagram for create tenant	34
Figure 3.20. System Sequence diagram for create notice	35
Figure 3.21. System Sequence diagram for tenant dashboard	35
Figure 3.22. System Sequence diagram for create leave notice	36
Figure 3.23. System Sequence diagram for send message each other	36
Figure 3.24. System Sequence diagram for report	37
Figure 3.25. System Sequence diagram for change password	37
Figure 4.1. Sequence diagram for tenant	39
Figure 4.2. Sequence diagram for admin/home owner	39
Figure 4.3. Entity Diagram for Vara-varatia	40
Figure 5.1. Testing Schedule for Vara-varatia	46



# Chapter 1

## Introduction

## **Introduction**

Vara-varatia is a web application management system where home owner / apartment owner / home builder makes better communication with tenant. This project is intended to make the home related management system easier to the home owner. A huge number of tenant and landlord keep unnecessary document every transaction detail between them. And, End of story missing of some hard document they cannot estimate actual profit or expenses. In which, both of them suffering long. This system makes simple their every step.

### **1.1 Project Overview:**

Every month building owner have to collect rent from tenant. That's not a big deal. It became more difficult when they calculate their rent or their dew. When an owner has more than one building, they have to remind it his memory or use note book that which tenant didn't pay this month rent. And it's disgusting to be owner that he has to remind his tenant dew for every month. Cause it cannot say surely every tenant paid his ren at due time. So, it's a huge amount for calculate and make an annual or monthly report. Some owner daily expense going based on this amount. Maximum owner maintains caretaker who maintains these criteria. And some of them make mismatch the transaction as intentionally or unintentionally. The average amount of such calculations per month becomes a good amount at the end of the year. After end of the day, there is no clear report of every transaction. Other side of tenant, before leaving home he has make appointment with owner and make payment clearance. Old process is a lengthy process. Cause before leaving home, he has to confirm his owner minimum one month ago. Because of communication gap, is there any need of repair of home. it's still remained as a pending. The old management has been going through some irregularities from the beginning. Both the landlord and the tenant suffer with this old management system. Vara-varatia is simple account management web base system that can resolve maximum issues. For surfing this web application user not need to learn principle of accounting. Its easy more than nowadays social media site. One report, that owner can find his dues, expenses, income from home rent. Two level of user of this system. One Admin and other one is tenant. Owner can message tenant through web system and can post notice. This system can record all transection. Vara-varatia can record all the tenant dues and monthly payment for owner. An owner easily can check how much payment done by tenant this month. And also has a list of dues of payment. This report will auto generate and before the starting of the month he has a possible actual report how much money he will gain during the month. That's report helps him to make possible expenses list this month. This list can make into Vara-varatia. Owner has daily, monthly, yearly report. No needs to calculate or reminds payment, dues and possible expenses. Owner directly connects to the tenant through using messaging with our system and notice. Has a report of which floor occupied or not? Owner can save tenant details into system. If there any incident done by tenant, this information can help trance out the tenant. No need to searching his old file which he was provided when rent the flat. And the other side of tenant, He will get notice and message if the owner provided this to him. He can view his previous due and upcoming due. Also, can view total payment of his. Can create issue if there need any repairs to home. Before leaving home, one month before he can make leaving approval to owner.

## **1.2 Project Purpose:**

Vara-varatia can record all the tenant dues and monthly payment for owner. An owner easily can check how much payment done by tenant this month. And also has a list of dues of payment. This report will auto generate and before the starting of the month he has a possible actual report how much money he will gain during the month. That's report helps him to make possible expenses list this month. This list can make into Vara-varatia. Owner has daily, monthly, yearly report. No needs to calculate or reminds payment, dues and possible expenses. Owner directly connects to the tenant through using messaging with our system and notice. Has a report of which floor occupied or not? Owner can save tenant details into system. If there any incident done by tenant, this information can help trance out the tenant. No need to searching his old file which he was provided when rent the flat. And the other side of tenant, He will get notice and message if the owner provided this to him. He can view his previous due and upcoming due. Also, can view total payment of his. Can create issue if there need any repairs to home. Before leaving home, one month before he can make leaving approval to owner. This system can record all transection. Vara-varatia can record all the tenant dues and monthly payment for owner. An owner easily can check how much payment done by tenant this month. And also has a list of dues of payment. This report will auto generate and before the starting of the month he has a possible actual report how much money he will gain during the month. That's report helps him to make possible expenses list this month. This list can make into Vara-varatia. Owner has daily, monthly, yearly report. No needs to calculate or reminds payment, dues and possible expenses. Owner directly connects to the tenant through using messaging with our system and notice. Has a report of which floor occupied or not? Owner can save tenant details into system. If there any incident done by tenant, this information can help trance out the tenant. No need to searching his old file which is store drawer.

### **1.2.1. Background:**

**Technical Perspective:** We have been working in a real-life project using PHP and MySQL Database.

**Domain Perspective:** i have tried to do something new like Improvement / make-up existing home rental system. This type of system does not exist in our country currently.

### **1.2.2. Benefit:**

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

1. Auto generate tenant registration form
2. Annual Report of dew/expenses
3. Owner easily can send message to tenant
4. Can post emergency notice
5. Owner can add, edit or remove tenant history
6. Owner can create list of possible cost
7. Tenant and Owner both of them has simple & smart dashboard
8. Tenant can make leaving approval
9. Tenant can make home repair related issues

### **1.2.3. Goals:**

The main goals of this project are to-

#### **Objective 1 (Clear Report):**

An owner easily can check how much payment done by tenant this month. And also has a list of dues of payment. This report will auto generate and before the starting of the month he has a possible actual report how much money he will gain during the month. That's report helps him to make possible expenses list this month. This list can make into Vara-varatia. Owner has daily, monthly, yearly report. No needs to calculate or reminds payment, dues and possible expenses.

#### **Objective 2 (Estimate Cost):**

Dashboard will auto generate and before the starting of the month he has a possible actual report how much money he will gain during the month. That's report helps him to make possible expenses list this month.

### **Objective 3 (Manage Tenant):**

Owner can save tenant details into system. If there any incident done by tenant, this information can help trace out the tenant. No need to searching his old file which he was provided when rent the flat. And the other side of tenant, He will get notice and message if the owner provided this to him.

### **Objective 4 (Better Communication):**

Owner directly connects to the tenant through using messaging with our system and notice. Tenant will get notice and message if the owner provided this to him. He can view his previous due and upcoming due. Also, can view total payment of his. Can create issue if there need any repairs to home. Before leaving home, one month before he can make leaving approval to owner.

### **Objective 5 (Manage Apartment):**

Has a report of which floor occupied or not? Tenant and owner both of them can create issue if there need any repairs to home.

## **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  
Tester: Testing software  
Operational team: Deploy it to client

### **2. External Stakeholder:**

Super Admin: It executive of Home Soft  
Admin: Owner of the software  
User: Tenant of the home or apartment

## 1.4. Proposed System Model:

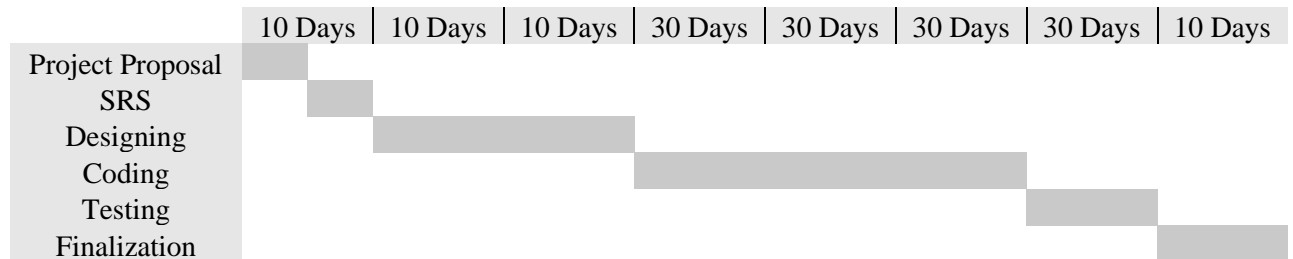
### Block Diagram



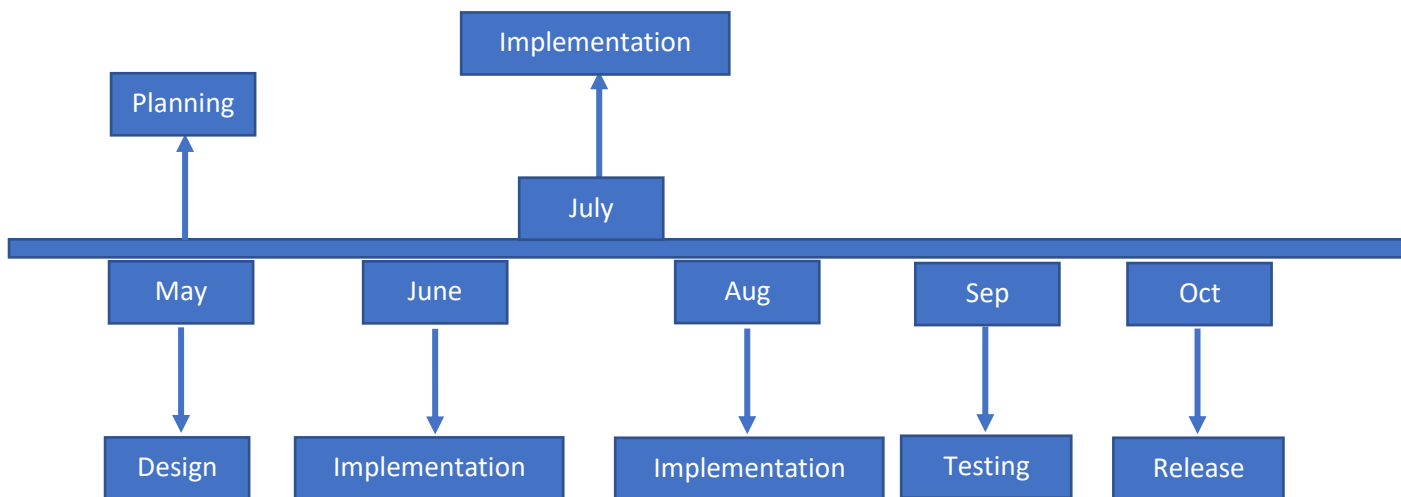
Figure 1.1: Figure of Vara-varatia block diagram

## 1.5. Project Schedule:

### 1.5.1. Gantt Chart



### 1.5.2. Milestone



### 1.5.3. Work Distribution

Task	Assigned to	Duration
SRS	Kh. Afjal Hossen	5 Days
Designing	Kh. Afjal Hossen	5 Days
Coding	Kh. Afjal Hossen	90 Days
Testing	Kh. Afjal Hossen	30 Days
Finalization	Kh. Afjal Hossen	10 Days

Figure 1.2: Four and half month work distribution (140 day`s)

## Chapter 2

# Software Requirement Specification



## SRS:

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

### 2.1. Functional Requirement:

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"><li>• Owner of the apartment</li><li>• Tenant of the apartment</li><li>• Software Developer</li></ul>

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

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

Send Message	Send message to user
Description	The user can message to each other
Originator	Null
Stakeholder	<ul style="list-style-type: none"><li>• Owner of the apartment</li><li>• Tenant of the apartment</li><li>• Software Developer</li></ul>

Create Notice	Notice for user
Description	The user can create notice to each other
Originator	Null
Stakeholder	<ul style="list-style-type: none"><li>• Owner of the apartment</li><li>• Tenant of the apartment</li><li>• Software Developer</li></ul>

Leave Notice	Create leaving home notice
Description	The user can create leave notice
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> <li>• Software Developer</li> </ul>

Bank Dashboard	Manage bank account details
Description	The user can manage bank related details
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> </ul>

Tenant Manage	Create tenant account
Description	The user can create tenant account and manage them
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Software Developer</li> </ul>

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

## 2.2. Data Requirement:

## 2.3. Performance Requirement:

### 2.3.1. Speed And Latency Requirement:

This system required a fair speed especially when a admin or super admin their activity and can manage their dashboard.

<b>PR-01</b>	Page Refresh Rate
<b>Description</b>	While the home owner and tenants browsing this system the page will show within a moment. It also depends on home owner and tenant's internet connection.
<b>Stakeholders</b>	Home owner, Tenants

### 2.3.2. Precision or Accuracy requirements:

There is nothing specific accuracy requirements.

### 2.3.3. Capacity Requirement:

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

<b>PR-02</b>	At first the system will contain all the registered home owner and tenants. information
<b>Description</b>	The information of registered home owner and tenants will be stored in database.
<b>Stakeholders</b>	Home owner, Tenants

### 2.4. Dependability Requirement:

#### 2.4.1. Reliability Requirement:

This is a home and tenant management related project that's why it will be very sensitive for home owner and tenants. The main goal of our project, it decreases the complexity of those managing process.

#### 2.4.2. Availability Requirement:

Availability is more necessary for this project.

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

#### 2.4.3. Robustness & Fault Tolerance Requirement:

<b>DR-02</b>	The system manages over access
<b>Description</b>	Sometime multiple users can over access to the system. This system can multiple user access.
<b>Stakeholders</b>	N/A

#### 2.4.4. Safety-Critical requirements:

There is nothing specific Safety-Critical requirements.

## 2.5. Maintainability and Supportability:

### 2.5.1. Maintenance Requirements:

<b>MS-01</b>	The system can support to browse this site in any time.
<b>Description</b>	Home owner can access this site to manage their apartment any time. Can also manage tenants.
<b>Stakeholders</b>	Home owner, Tenants and Super admin

### 2.5.2. Supportability requirements:

There is nothing specific supportability requirements.

### 2.5.3. Adaptability requirements:

There is nothing specific adaptability requirements.

## 2.6. Security requirements:

This system has some security requirements. Like-

- Username/Password
- Validation
- Authentication

### 2.6.1. Access Requirements:

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

<b>SR-01</b>	This system gives security policy.
<b>Description</b>	Without register tenants and home owner in this system, he/she will not able to access the system. The mechanism provided security services.
<b>Stakeholders</b>	Home owner, Tenants and Super admin

### 2.6.2. Integrity Requirements:

To protect data all username and password are stored in encrypted form. It's not easy to decrypt the username and password.

### 2.6.3. Privacy Requirements:

This system has some privacy strategies. Like- Validation and Authentication. All data are stored in database. All username and password are stored in encrypted form. It's hard to decrypt. This mechanism provides users privacy.

<b>SR-02</b>	All data will be protected
<b>Description</b>	All data are stored in database. All username and password are stored in encrypted form. It's hard to decrypt.
<b>Stakeholders</b>	Home owner, Tenants

### 2.7. Usability and Human Integrity requirements:

#### 2.7.1. Ease of Use requirements:

This system is very easy for use and also understandable.

#### 2.7.2. Understand-ability and Politeness requirements:

This system is very easy for understand and also usable.

#### 2.7.3. Accessibility requirements:

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

AR-1: Log in as a Super Admin

AR-2: Log in as an Admin

AR-3: Log in as a Tenant

#### 2.7.4. Accessibility requirements:

Trainers of Home Soft will be handled it.

### 2.8. Look and Feel Requirements:

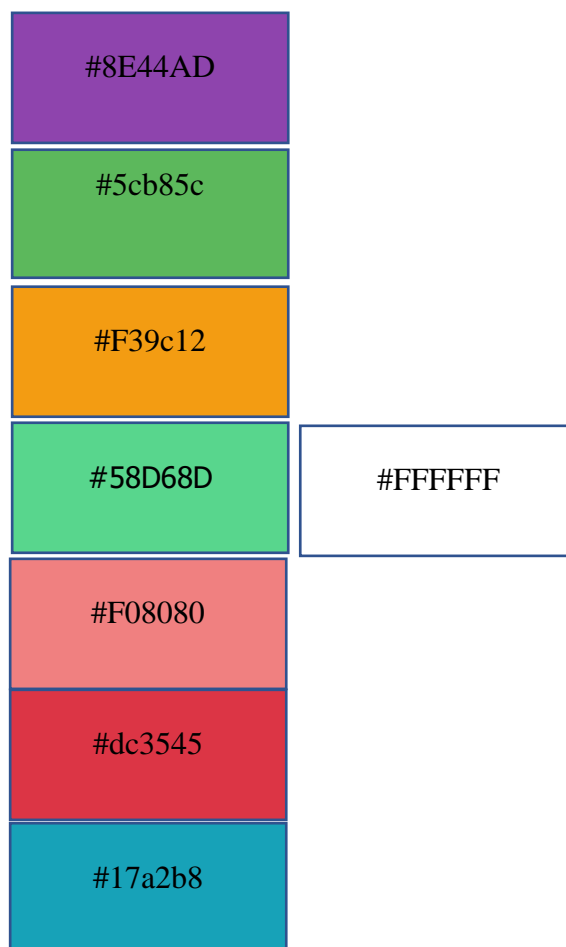
#### 2.8.1. Appearance requirements:

There is no Appearance Requirements.

#### 2.8.2. Style requirements:

In order to be able to maintain the look & feel and to adopt it to various (future) platforms the styling must be declarative. As we'll provide a web-based user interface cascading stylesheets (CSS) are the way to go. This requirement doesn't only define the necessity to use a CSS, but although the requirements regarding the CSS's content. Note that we might have to support client-specific styling late (custom styling per client; logo, ...)

<b>Style CSS</b>	The look & feel must be controllable using a stylesheet
<b>Description:</b>	The styling of the elements of the web-based user interface must be defined using a cascading stylesheet (CSS). For detailed requirements regarding the styling see the fit criteria.
<b>Originator:</b>	Null
<b>Stakeholders:</b>	<ul style="list-style-type: none"> <li>• Users</li> </ul>
<b>Fit Criterion:</b>	<ul style="list-style-type: none"> <li>-There's a CSS file that defines all appearance attributes</li> <li>-The layout grows to the actual screen size (no fixed width)</li> <li>-The style supports a 2-column layout with a sidebar to the left the style supports drop-down menus</li> <li>-The style supports internationalization (i.e., support for Mandarin)</li> <li>-The colors from the Vara-varatia website are used <ul style="list-style-type: none"> <li>➤ background: #8E44AD</li> <li>➤ #5cb85c</li> <li>➤ #F39c12</li> <li>➤ #58D68D</li> <li>➤ #F08080</li> </ul> </li> </ul>



The following image is taken from Vara-varatia web site. The color scheme can easily be seen



The dashboard displays financial information for a tenant. It features four summary cards at the top: 'Current Balance' (70,000/=), 'Current Due' (70,000/=), 'Previous Due' (70,000/=), and 'Estimet Cost' (95,000/=). Below these is a table of 'Possible Cost List' for the month of December, with a total of Tk. 95,000. The table lists two tasks: 'Roof/Tiles' (20,000) and 'Wash Room' (75,000). At the bottom, there is a table for 'Due of Sspecific Tenant Current Month: December' with columns for Name, Phone, Floor/Apt. Number, Last Payment date, and Current Due.

Task Name	Location	Amount	Action	
Roof/Tiles	2B	20000		
Wash Room	4A	75000		

Name	Phone	Floor/Apt. Number	Last Payment date	Current Due
------	-------	-------------------	-------------------	-------------

## **2.9. Operational and Environmental Requirement:**

### **2.9.1. Expected Physical Requirement:**

There is no specific requirement.

### **2.9.2. Requirement for Interfacing with Adjacent System:**

There is no specific requirement.

### **2.9.3. Productization Requirement:**

There is no specific requirement.

### **2.9.4. Release Requirement:**

There is no specific requirement.

## **2.10. Legal Requirement:**

### **2.10.1. Compliance Requirement:**

There is no specific requirement.

### **2.10.2. Standards Requirement:**

There is no specific requirement.



## Chapter 3

### System Analysis

## Scope of the study

### 3.1. Use Case Diagram:

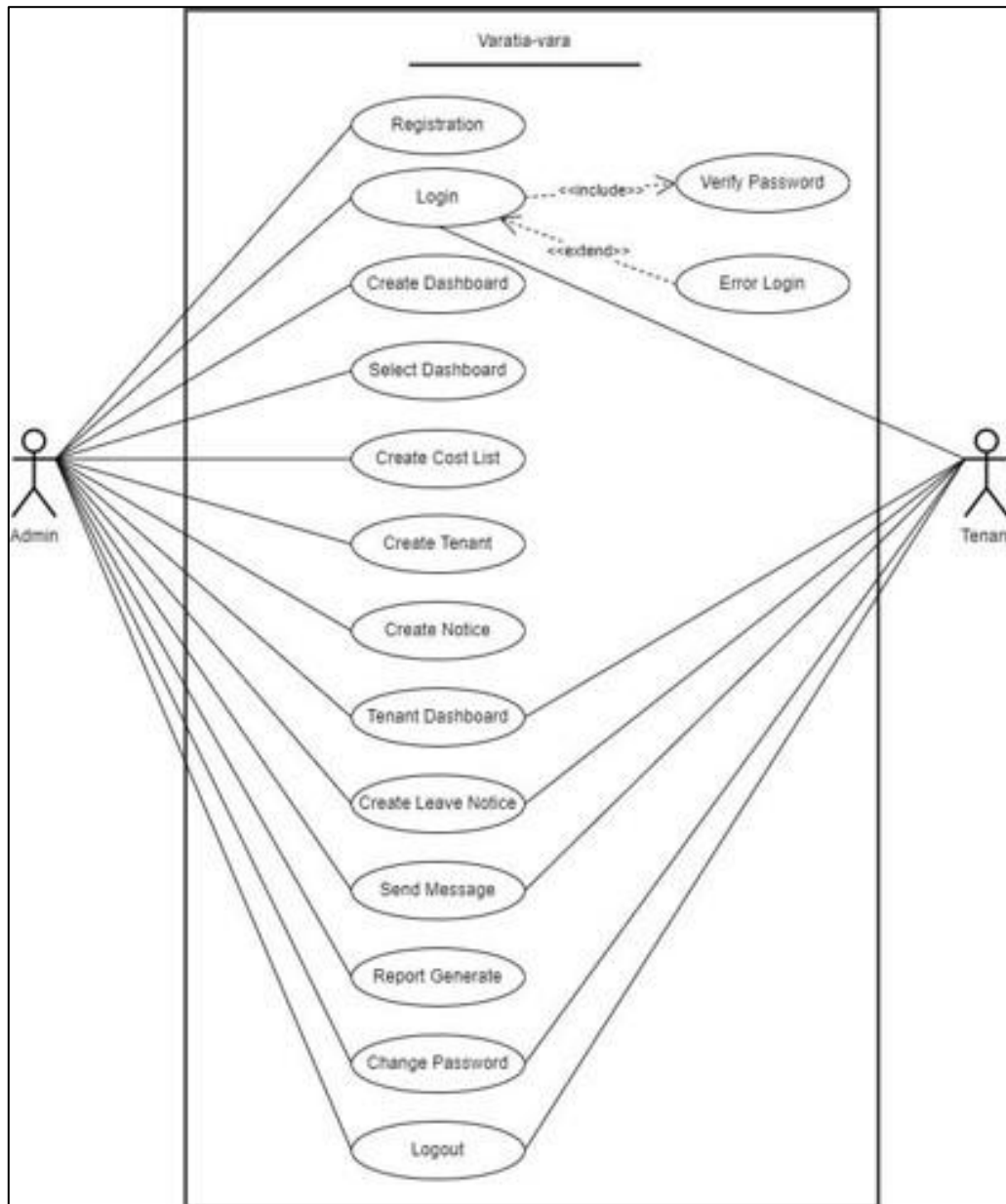


Figure 3.1. Use Case Diagram

### 3.1. Use Case Description:

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

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

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

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

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

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

<b>Create Cost List</b>	Miscellaneous cost for home owner
Description	The home owner can create miscellaneous cost list for future
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> </ul>
Scope	Create miscellaneous cost list through our system
Actor	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> </ul>
Trigger Type	Event Trigger
Triggered	The user wants to create cost list
Input	<ul style="list-style-type: none"> <li>• Apartment</li> <li>• Floor Number</li> <li>• Task Name</li> <li>• Estimated Cost</li> </ul>
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> <li>• The user opens miscellaneous option</li> <li>• The user select apartment</li> <li>• The user enters floor number</li> <li>• The user enters estimated cost</li> <li>• The user clicks the save button</li> </ul>
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> <li>➤ The user has created miscellaneous cost list</li> </ul>
Failed End Scenario	Wrong input

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

	<ul style="list-style-type: none"> <li>• Religion Status</li> <li>• Education</li> <li>• Job Title</li> <li>• Company Name &amp; Location</li> <li>• Father Name</li> <li>• Emergency Contact Name</li> <li>• Emergency Contact Address</li> <li>• Emergency Contact Number</li> <li>• Relation With Emergency Contact</li> <li>• Marital Status</li> <li>• Date of birth</li> </ul>
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> <li>• The user opens tenant add option</li> <li>• The user enters tenant name</li> <li>• The user enters tenant permanent address</li> <li>• The user enters tenant contact number</li> <li>• The user enters tenant NID number</li> <li>• The user enters tenant country</li> <li>• The user enters tenant religious status</li> <li>• The user enters tenant education qualification</li> <li>• The user enters tenant job qualification</li> <li>• The user enters tenant job company name</li> <li>• The user enters tenant father name</li> <li>• The user enters tenant emergency contact name</li> <li>• The user enters tenant emergency contact number</li> <li>• The user enters tenant relation with emergency contact</li> <li>• The user enters tenant emergency contact address</li> <li>• The user enters tenant martial status</li> <li>• The user enters tenant date of birth</li> <li>• The user clicks the save button</li> </ul>
Alternative Scenario	Null
Success End Scenario	➤ The user has added tenant member
Failed End Scenario	Wrong input

<b>Create Notice</b>	Notice for user
Description	The user can create notice to each other
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> </ul>
Scope	Create notice through our system
Actor	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> </ul>
Trigger Type	Event Trigger
Triggered	The user wants create notice
Input	<ul style="list-style-type: none"> <li>• Notice Header</li> <li>• Notice</li> </ul>
Precondition	Has an account

Main Success Scenario	<ul style="list-style-type: none"> <li>• The user opens Noticeboard option</li> <li>• The user create notice</li> <li>• The user enters notice header</li> <li>• The user enters the notice</li> <li>• The user clicks the save button</li> </ul>
Alternative Scenario	Null
Success End Scenario	➤ The user has created notice
Failed End Scenario	Null

<b>Tenant Dashboard</b>	Every tenant has his own dashboard
Description	The tenant views his own dashboard
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> </ul>
Scope	View dashboard in our system
Actor	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Software Developer</li> <li>• Tenant of the apartment</li> </ul>
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"> <li>• The user view dashboard</li> </ul>
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> <li>➤ The user has entered the to the system</li> <li>➤ The system creates a new database for user</li> </ul>
Failed End Scenario	Null

<b>Leave Notice</b>	Leaving home notice
Description	The user can create leave notice
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> <li>• Software Developer</li> </ul>
Scope	Create leave notice through our system
Actor	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> <li>• Tenant of the apartment</li> <li>• Software Developer</li> </ul>
Trigger Type	Event Trigger
Triggered	The user wants create leave notice
Input	<ul style="list-style-type: none"> <li>• Select Leave Category</li> <li>• Select Leave Date</li> <li>• Any Message</li> </ul>
Precondition	Has an account
Main Success Scenario	<ul style="list-style-type: none"> <li>• The user opens Leave Notice option</li> <li>• The user selects leave category</li> </ul>

	<ul style="list-style-type: none"> <li>• The user selects leave date</li> <li>• The user clicks the save button</li> </ul>
Alternative Scenario	Null
Success End Scenario	➤ The user has created leave notice
Failed End Scenario	Wrong input

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

<b>Report Generate</b>	Report generates automatically daily basis input
Description	The tenant views monthly and yearly
Originator	Null
Stakeholder	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> </ul>
Scope	View report in our system
Actor	<ul style="list-style-type: none"> <li>• Owner of the apartment</li> </ul>
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"> <li>• The user view report</li> </ul>
Alternative Scenario	Null
Success End Scenario	<ul style="list-style-type: none"> <li>➤ The user has entered the to the system</li> <li>➤ The system generate report for user</li> </ul>
Failed End Scenario	Null

<b>Change Password</b>	Change password of account
------------------------	----------------------------



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

### 3.3. Activity Diagram:

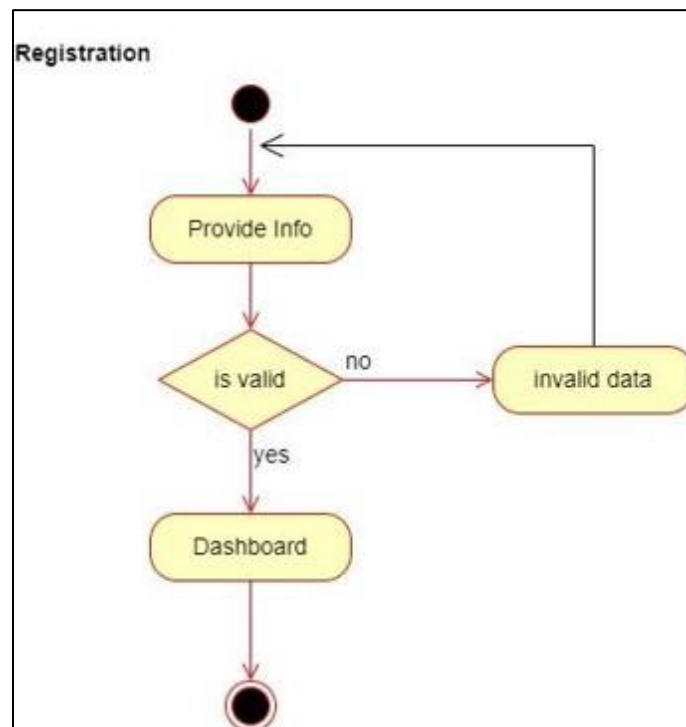


Figure 3.2. Activity Diagram for Registration

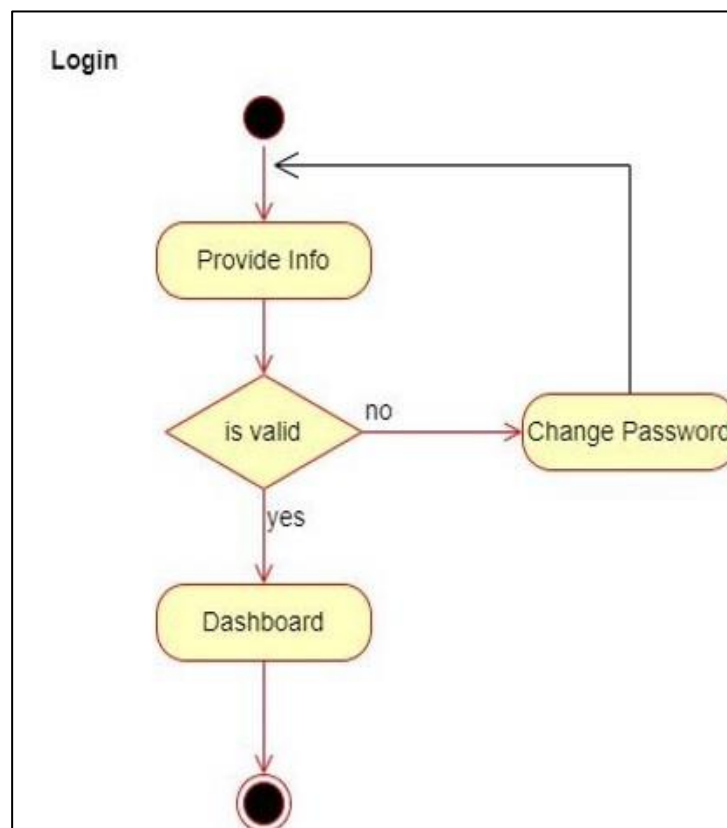


Figure 3.3. Activity Diagram for login

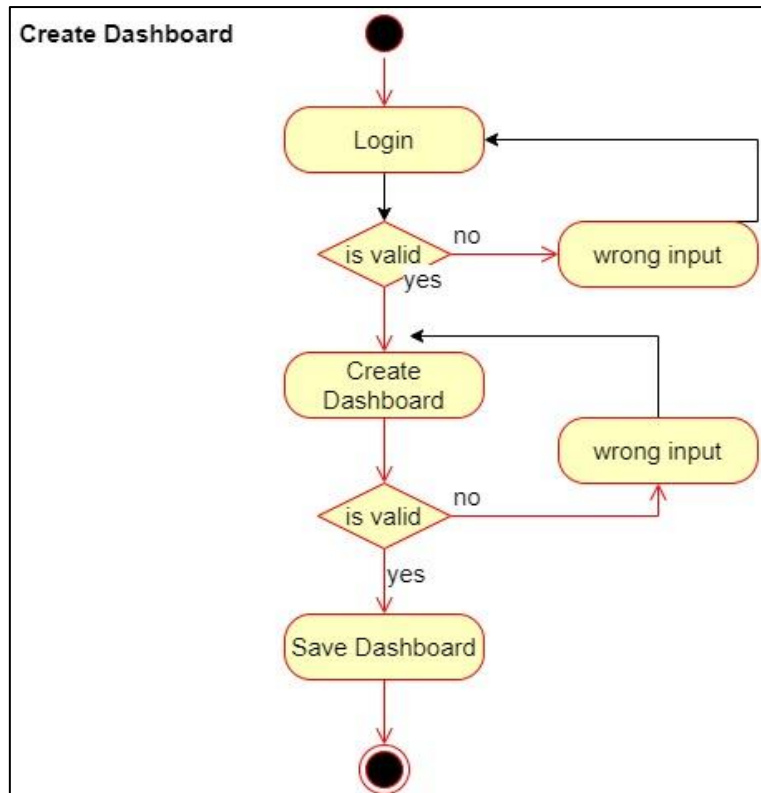


Figure 3.4. Activity Diagram for create dashboard

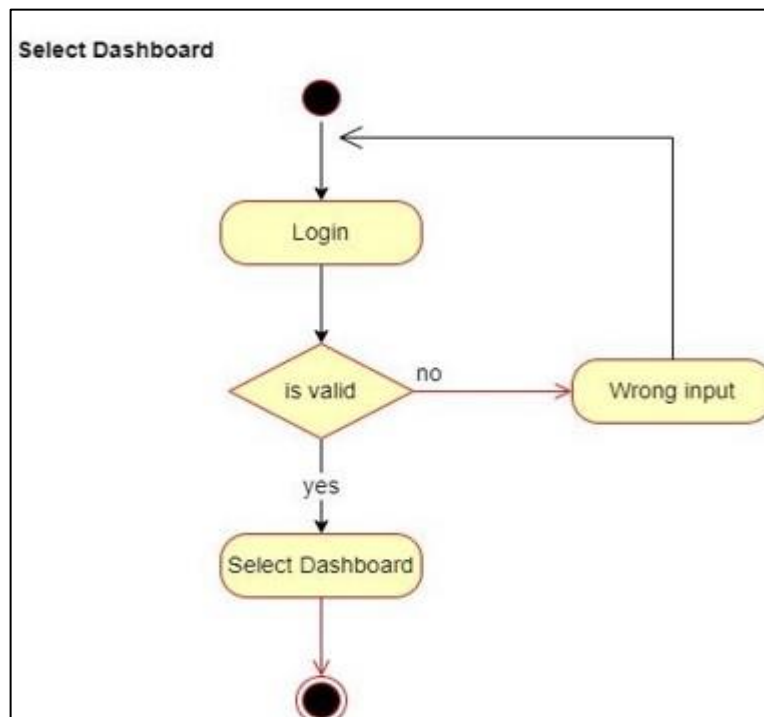


Figure 3.5. Activity Diagram for select dashboard

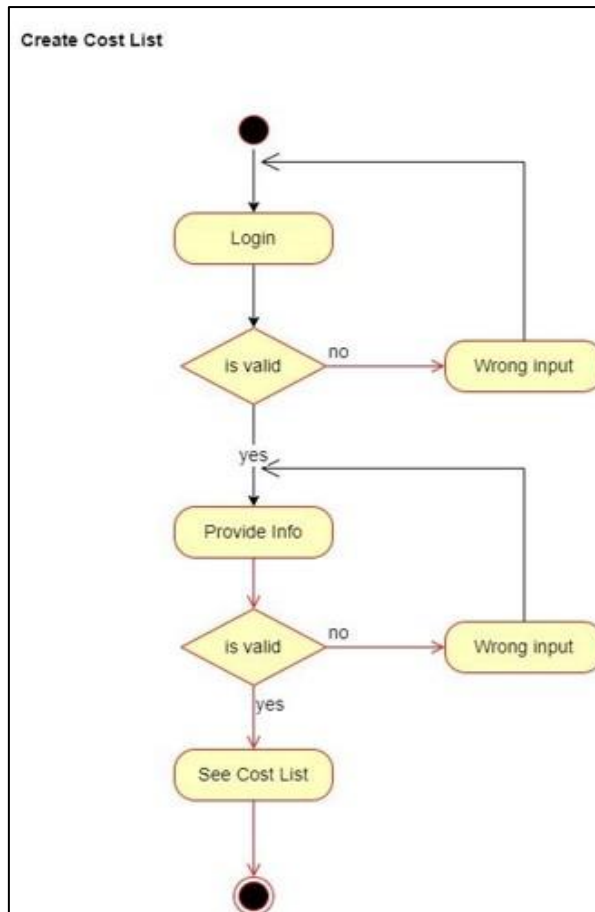


Figure 3.6. Activity Diagram for create cost list

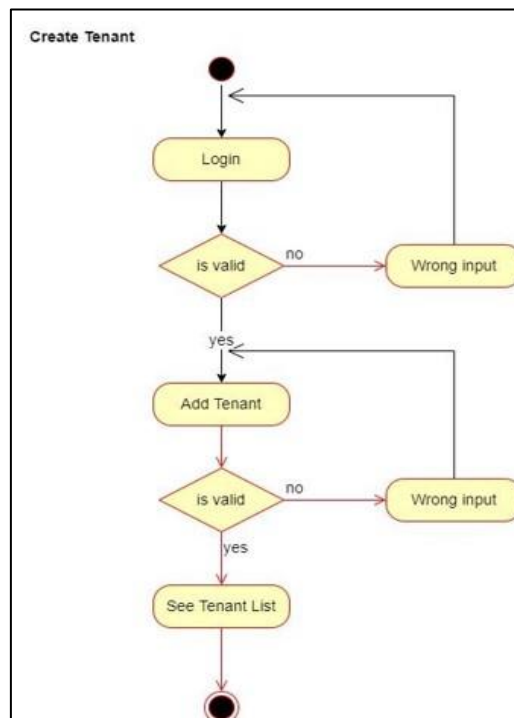
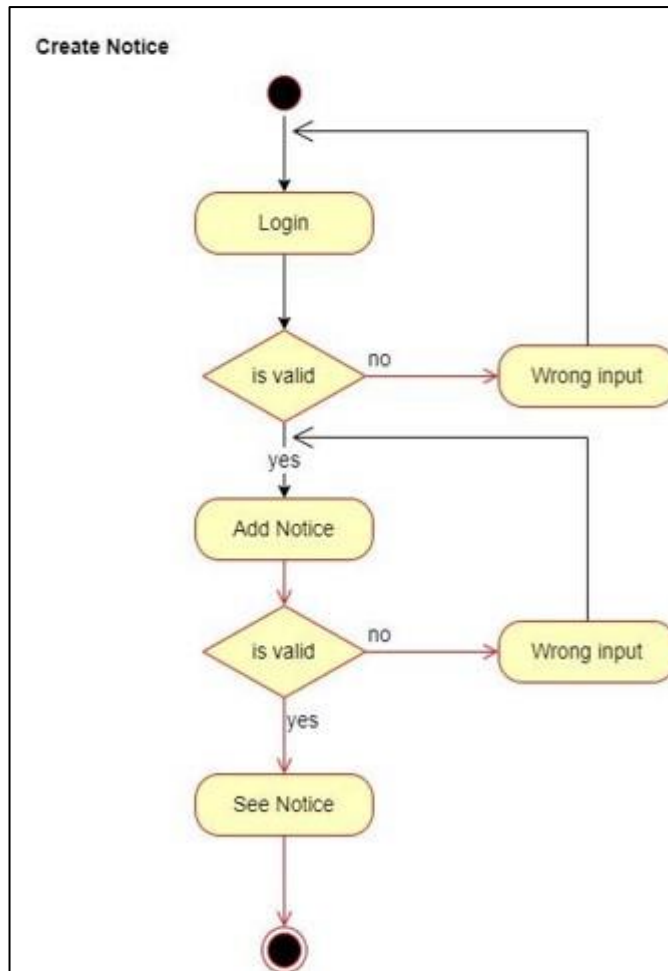
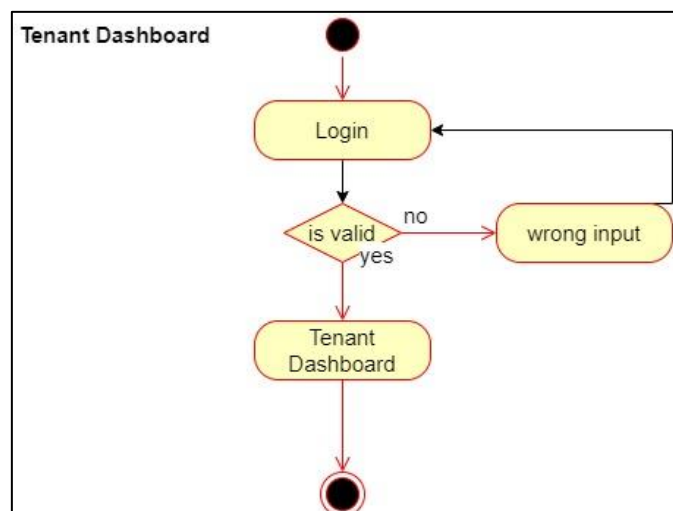


Figure 3.7. Activity Diagram for create tenant



**Figure 3.8. Activity Diagram for create notice**



**Figure 3.9. Activity Diagram for tenant dashboard**

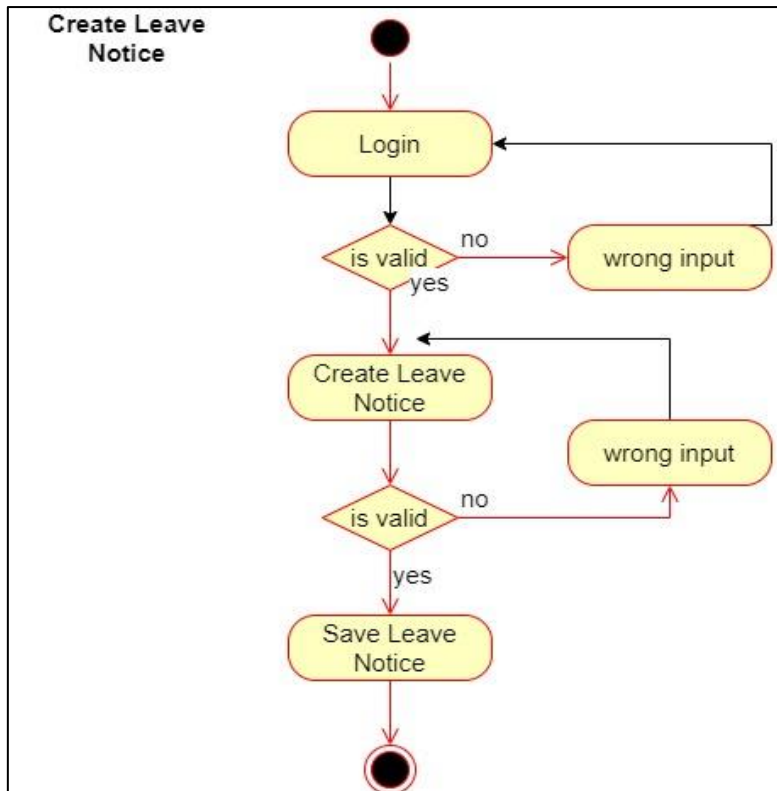


Figure 3.10. Activity Diagram for create leave notice

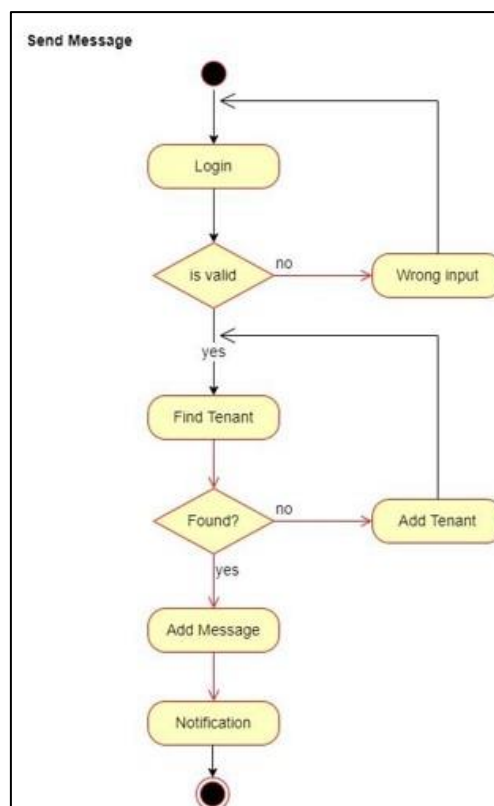
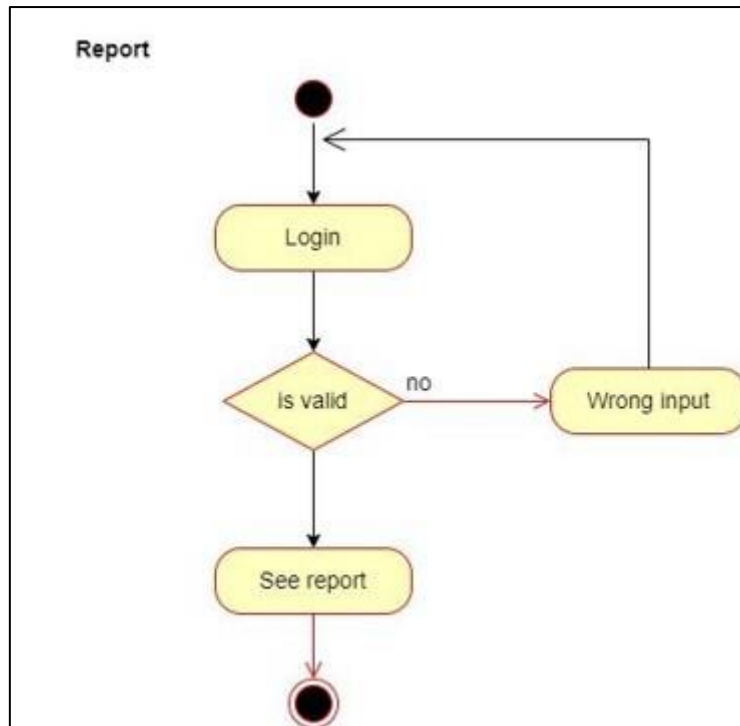
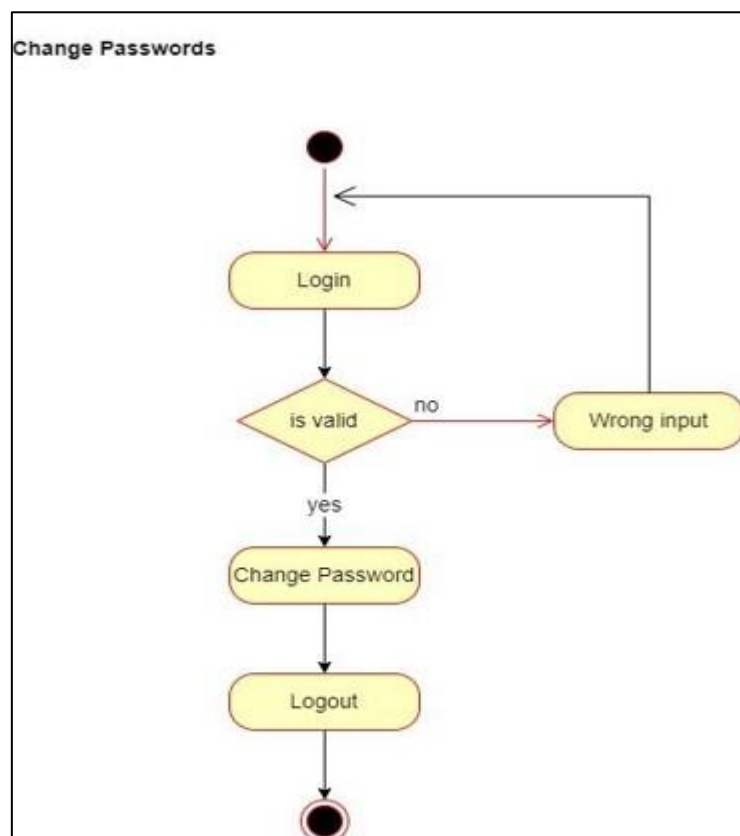


Figure 3.11. Activity Diagram for send message



**Figure 3.12. Activity Diagram for report**



**Figure 3.13. Activity Diagram for change password**

### 3.4. System Sequence Diagram:

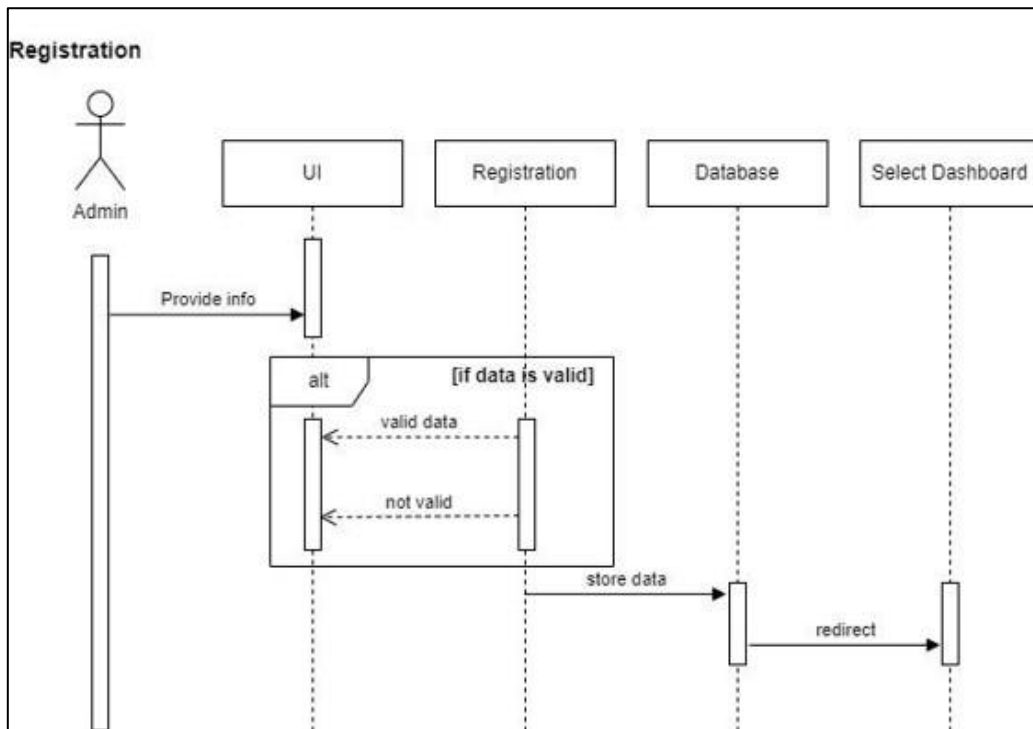


Figure 3.14. System Sequence diagram for registration

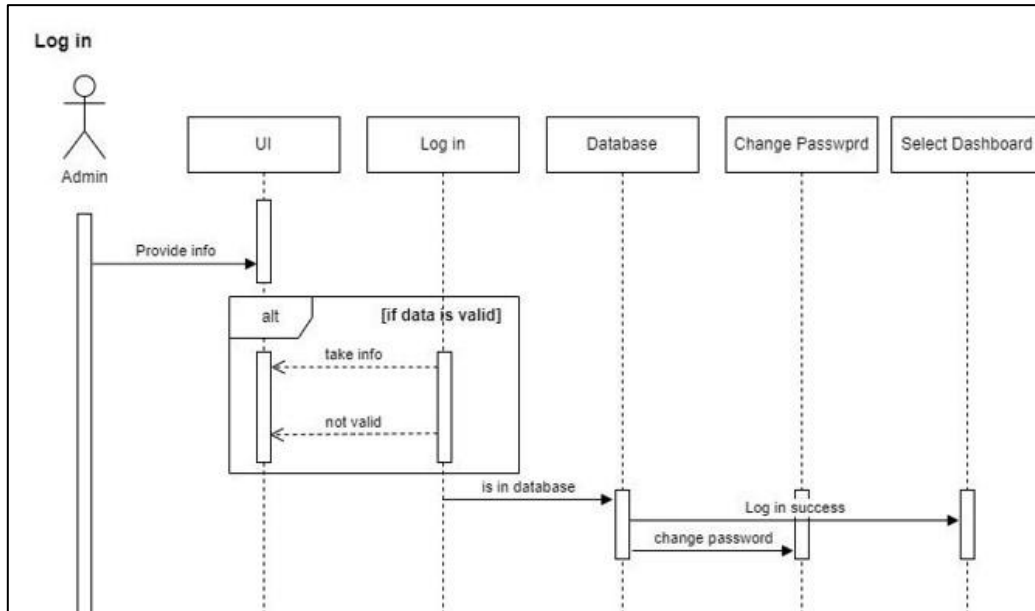


Figure 3.15. System Sequence diagram for login



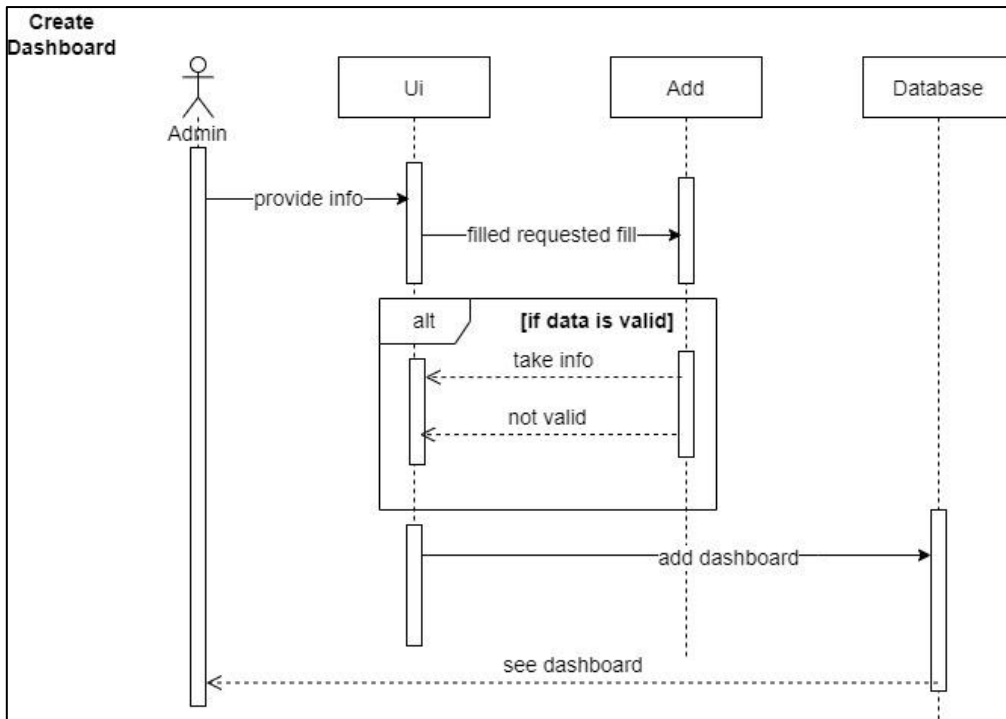


Figure 3.16. System Sequence diagram for create dashboard

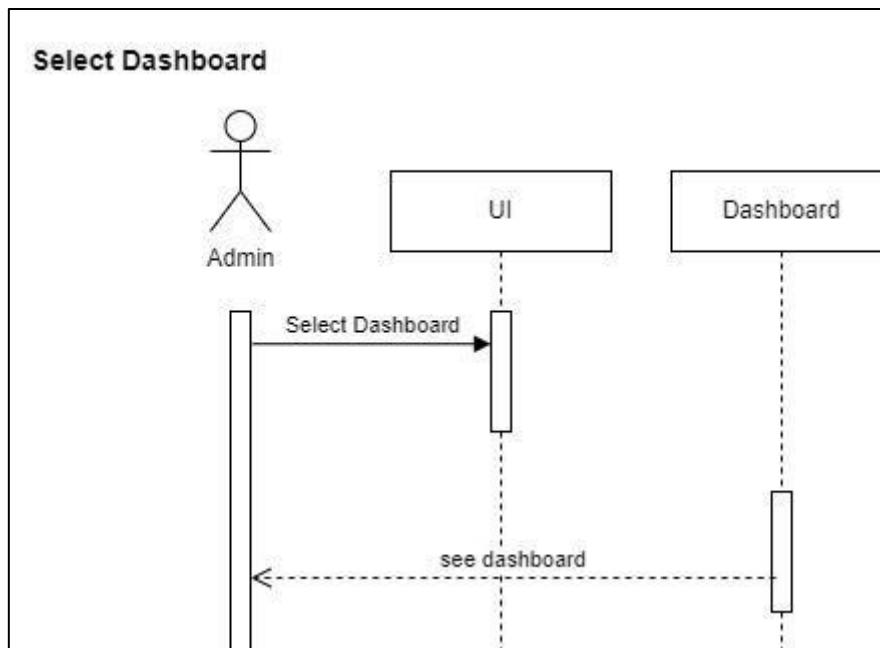


Figure 3.17. System Sequence diagram for select dashboard

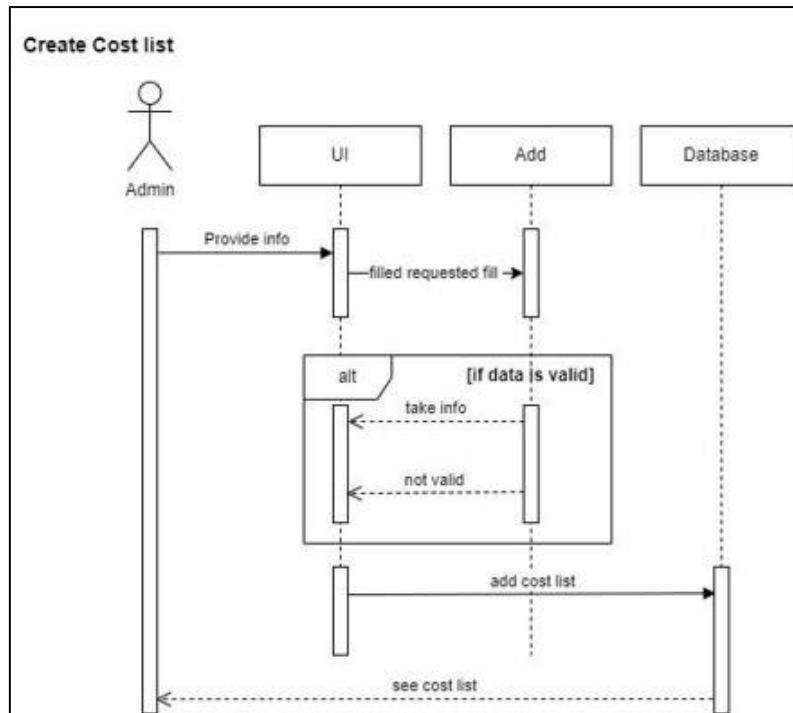


Figure 3.18. System Sequence diagram for create cost list

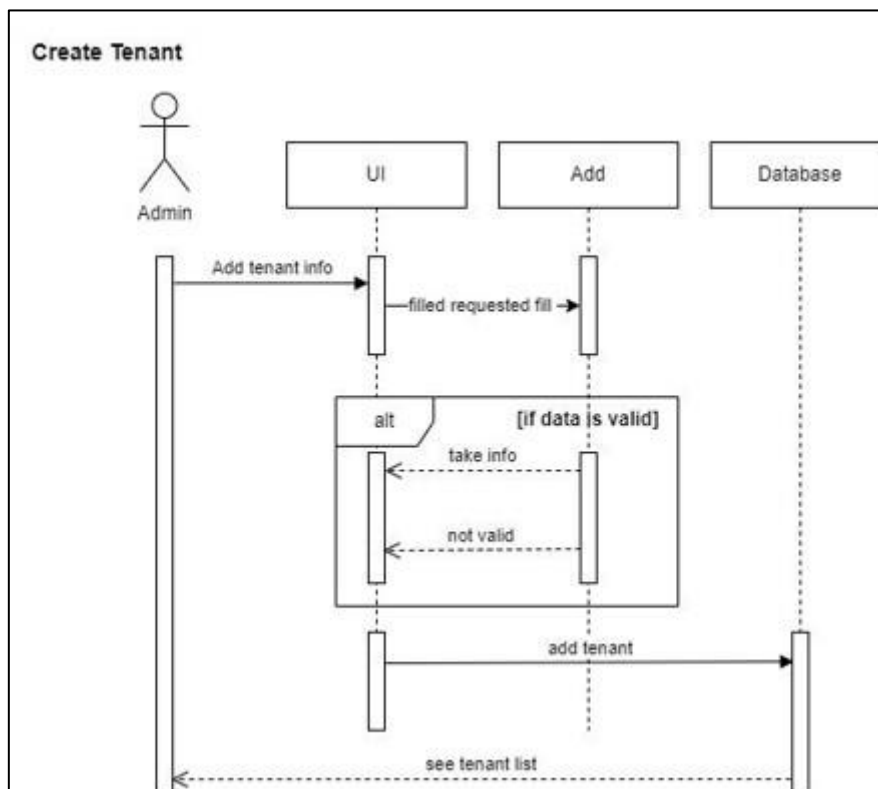


Figure 3.19. System Sequence diagram for create tenant

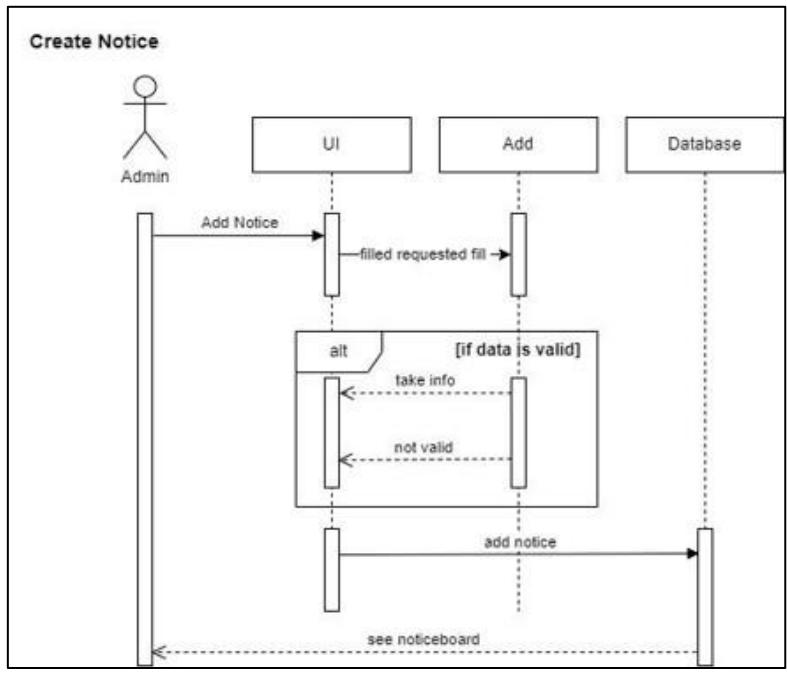


Figure 3.20. System Sequence diagram for create notice

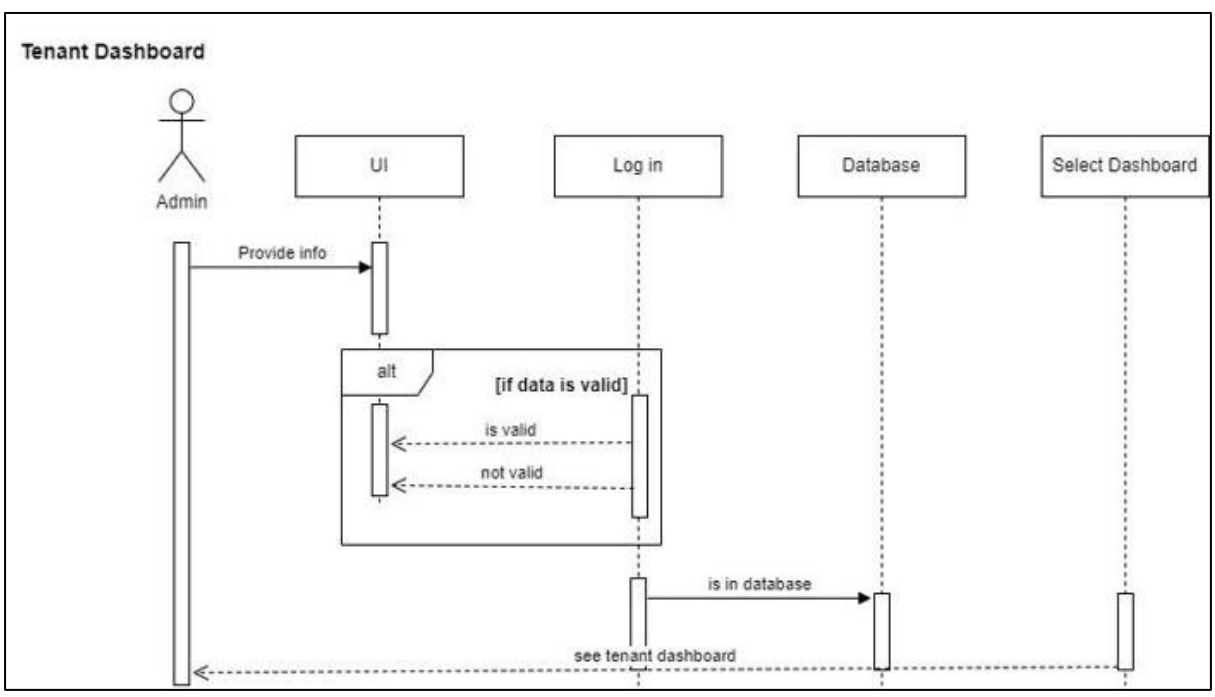


Figure 3.21. System Sequence diagram for tenant dashboard

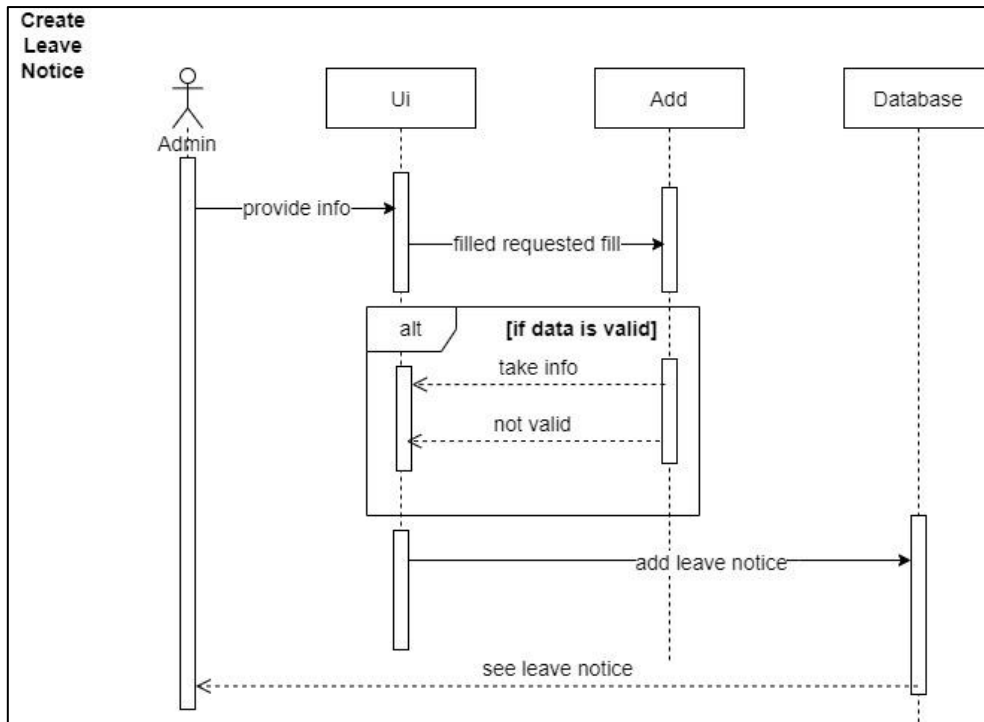


Figure 3.22. System Sequence diagram for create leave notice

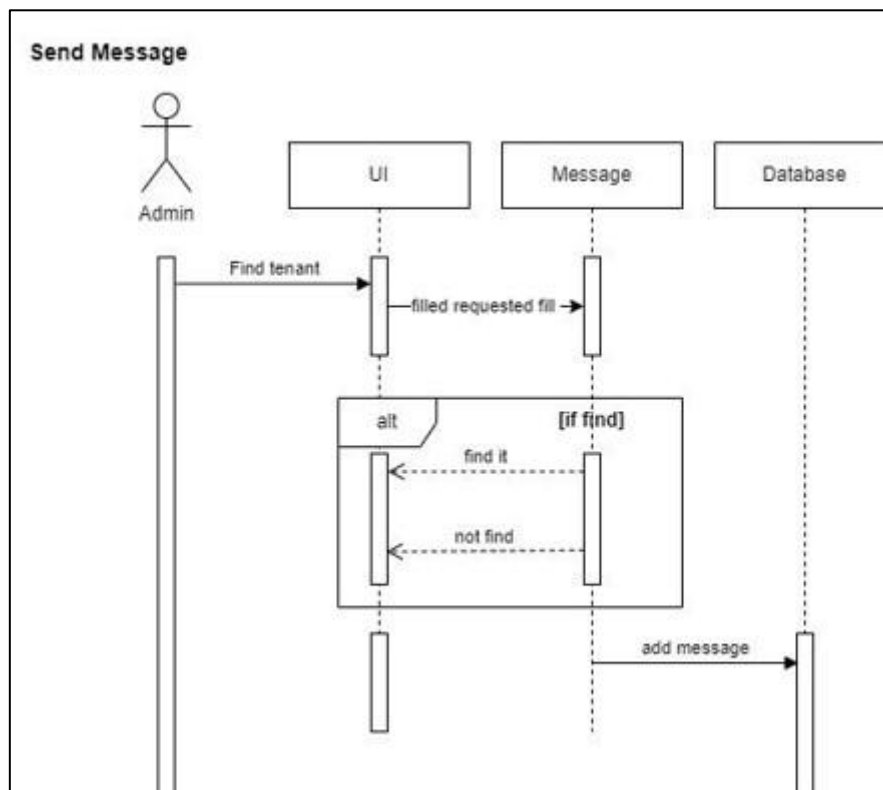


Figure 3.23. System Sequence diagram for send message each other

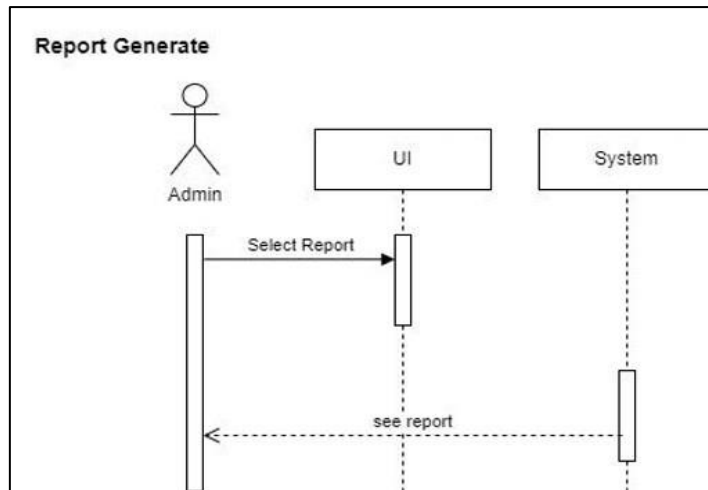


Figure 3.24. System Sequence diagram for report

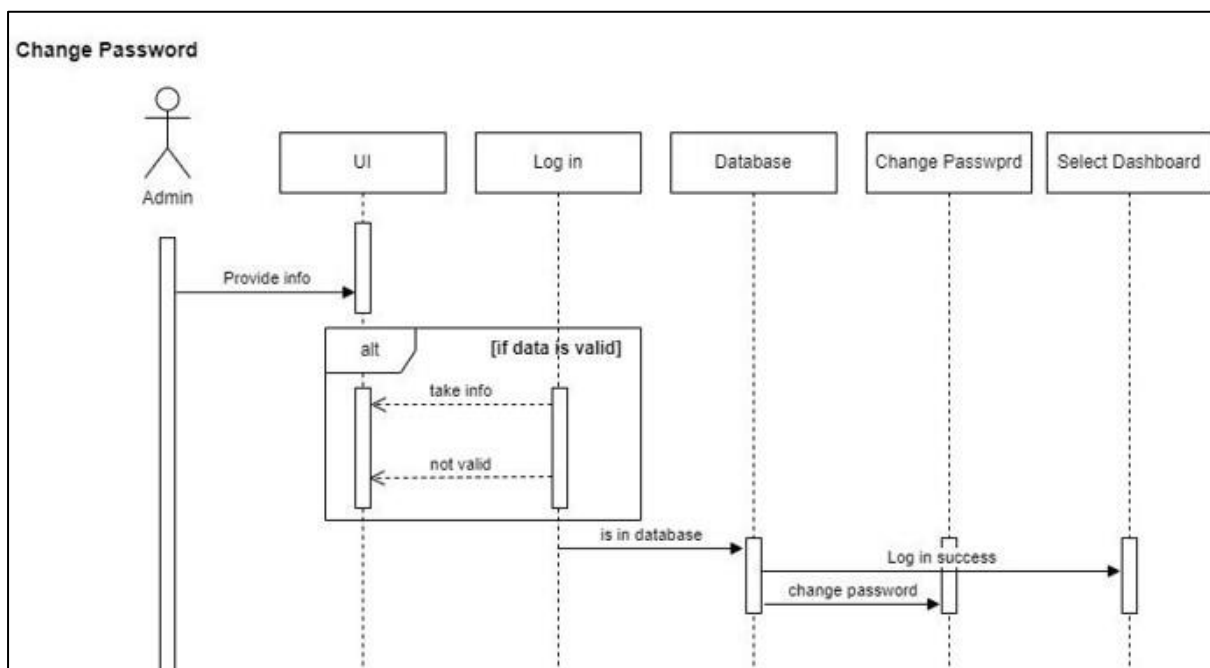


Figure 3.25. System Sequence diagram for change password

## Chapter 4

# System Design Specification

## System Design Specification:

### 4.1. Sequence Diagram:

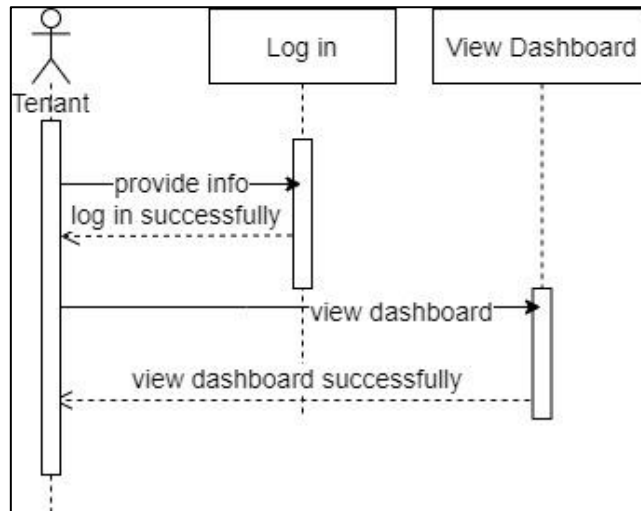


Figure 4.1. Sequence diagram for tenant

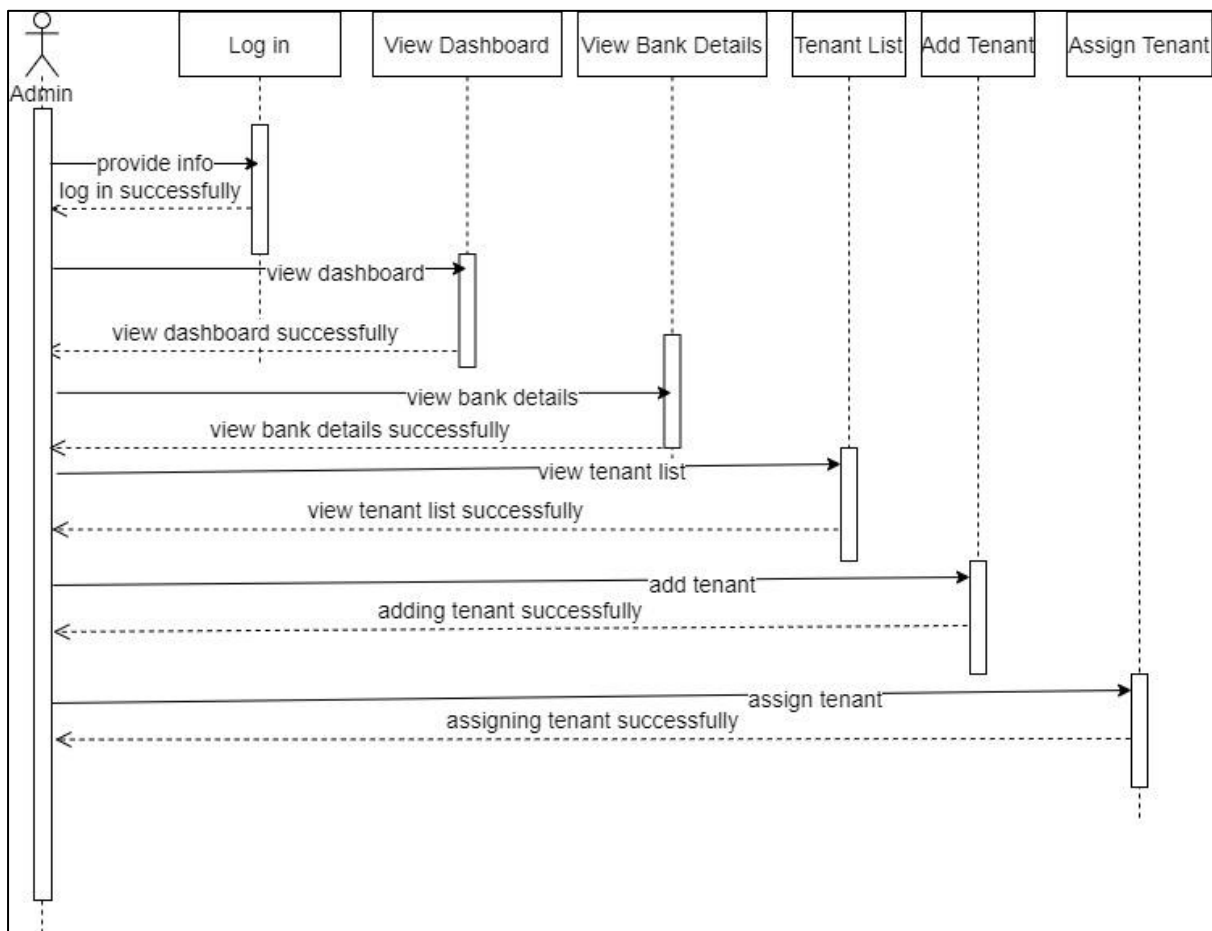


Figure 4.2. Sequence diagram for admin/home owner

## 4.2. Entity Relation Diagram:

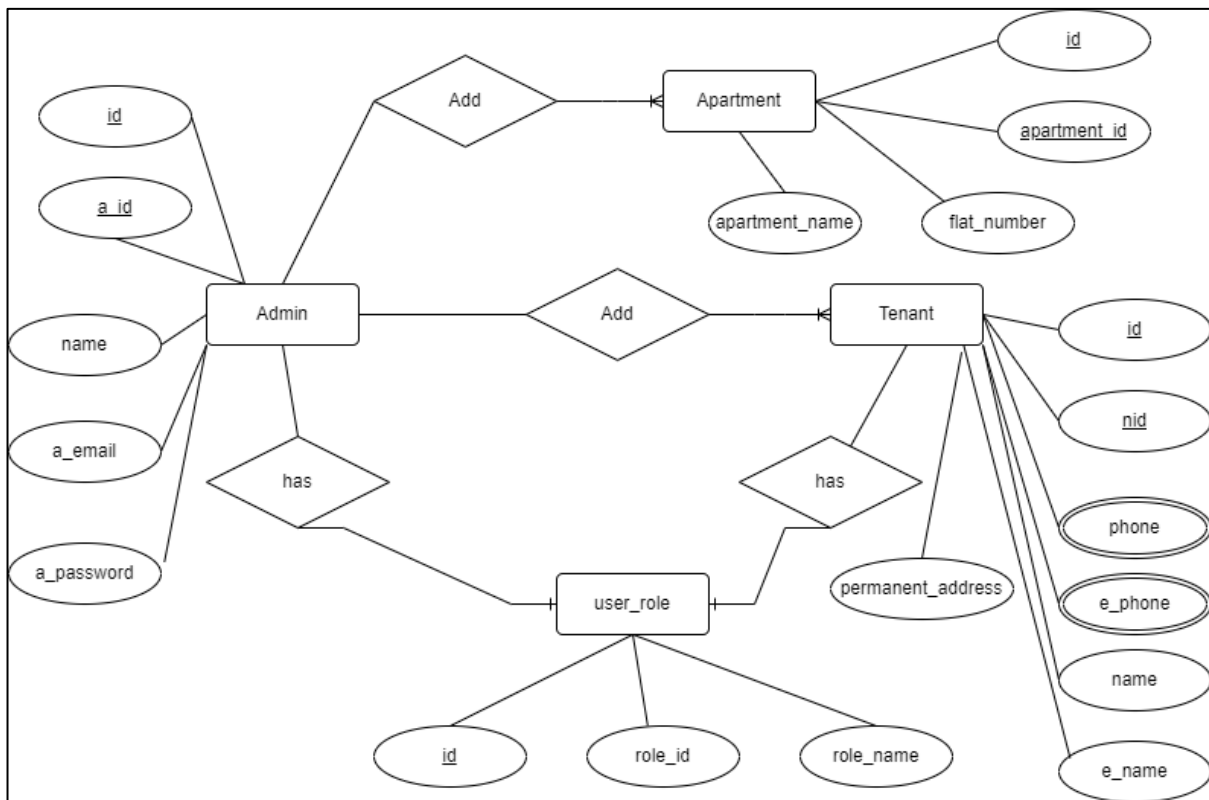


Figure 4.3. Entity Diagram for Vara-varatia

## 4.3. Development Tools & Technology:

### 4.3.1. User Interface Technology:

#### 4.3.1.1 Framework:

Laravel Framework v7.30.4 used

#### 4.3.1.2. JS:

Java Script v15.11.0 used

#### 4.3.1.3. CSS Framework:

Bootstrap v4 and CSS in this project used

#### 4.3.1.4. Icon Library & Toolkit:

Font awesome v6 used



## **4.3.2. Implementation Tools & Platforms:**

### **4.3.2.1. Source Code Editor:**

Microsoft Visual Studio v1.71.2 used.

### **4.3.2.2. Language:**

PHP v7.3.28 used as Language.

### **4.3.2.3. MySQL Server:**

MySQL Server used as Database

### **4.3.2.4. Apache HTTP Server:**

For run this Project use Apache HTTP Server as local server.

## Chapter 5

### System Testing

## **System Testing**

### **5.1. Testing Feature:**

**Unit Testing** is a software technique method. Here test every separate unit of code. I actually used this method after finish the project. I found some error in my system. Then I try to resolve it.

**Acceptance Testing** is used before the system release. When I use this method, I actually found some minor error. Then I try to resolve it. After solve the system bug I release it for use.

#### **5.1.1. Features To Be Tested:**

- Request For Account Open
- Login
- Log Out
- Create Dashboard
- Add Miscellaneous Cost
- Add Tenant
- Manage Tenant
- Assign Tenant
- Manage Dashboard
- Add Notice
- Add Leave Notice
- Profile

#### **5.1.2. Features Not to Be Tested:**

- Report
- Bank Details
- Tenant Dashboard
- Tenant Login

## **5.2. Testing Strategies:**

### **5.2.1. Test Approach:**

#### **Test Case 01:**

##### **5.2.1.1. Request for account, Login & Logout:**

- After Successfully Request for account open, this request serve to Super Admin. Super Admin check the user is eligible or not. After approved admin/home owner can login and logout successfully.
- After login as a admin/home owner it takes to select dashboard.

#### **Test Case 02:**

##### **5.2.1.2. Add Dashboard:**

- After login as an admin/home owner I check to create a dashboard for specific apartment.
- After create dashboard it redirects to dashboard successfully.

#### **Test Case 03:**

##### **5.2.1.3. Add Miscellaneous Cost:**

- In specific dashboard, I check to create a miscellaneous cost for apartment.
- Enter required item for miscellaneous cost and click submit button and check that it works properly.
- After create dashboard it redirects to dashboard successfully.

#### **Test Case 04:**

##### **5.2.1.4. Add & Manage Tenant:**

- In tenant module, I check to add a tenant in our system.
- Enter required item for tenant and click submit button and check that it works properly.
- After create tenant it takes us to tenant list successfully.
- After add tenant I checked to update, edit or delete tenant for managing them and add rent to their id.

## **Test Case 05:**

### **5.2.1.5. Assign Tenant:**

- In tenant module, I check to assign a tenant in specific apartment.
- Enter required item for tenant and click submit button and check that it works properly.
- After assigning tenant it takes us to tenant assign list successfully.

## **Test Case 06:**

### **5.2.1.6. Manage Dashboard:**

- In tenant module, I check to go to our in our system.
- Add tenant rent, dues and check that it works properly.
- It shows dashboard property properly.

## **Test Case 07:**

### **5.2.1.7. Add & Notice & Leave Notice:**

- In notice module, I check to add a notice & leave notice in our system.
- Enter required item for tenant and click submit button and check that it works properly.
- After create notice & leave notice it takes us to notice board successfully.

## **Test Case 08:**

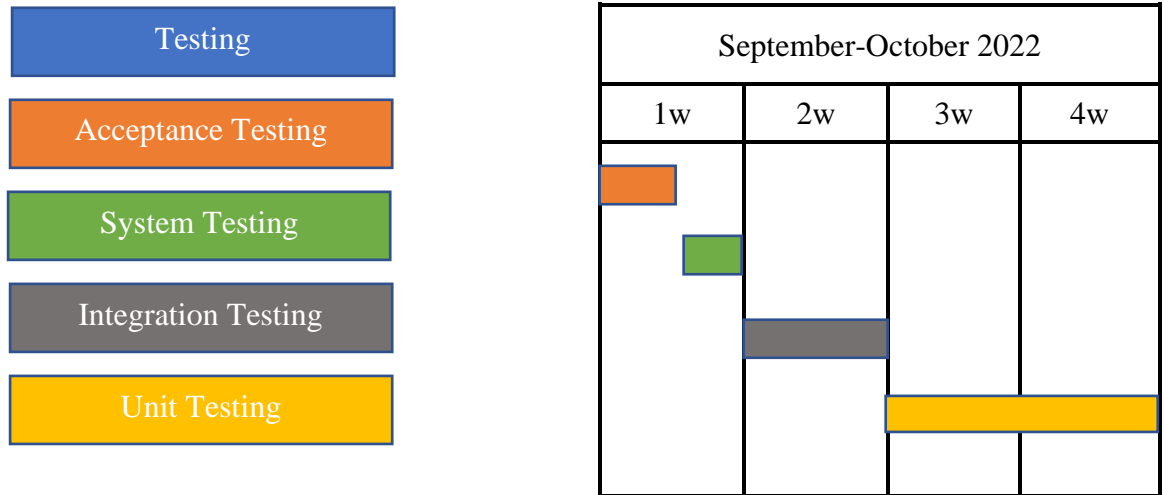
### **5.2.1.8. Profile:**

- In profile module, I check see what all data is accurate or not.
- All date come from database which entered for creating every module from our system.

### **5.2.2. Pass/Fail Criteria:**

My Project passes all the criteria.

### 5.2.3. Testing Schedule:



**Figure 5.1. Testing Schedule for Vara-varatia**

## 5.2.4. Traceability Matrix:

S no.	Req. Id	Req. Description	Scenario	TC Id	TC Description	Test Result	Defect Id	Defect Status
01	BR 1.0	1.Login	1. Account Open	TC_varatia_vara_account_open_01	Request to super admin for create account	Pass	N/A	N/A
			2. Login	TC_varatia_vara_account_open_01	Login into system	Pass	N/A	N/A
			3. Logout	TC_varatia_vara_account_open_01	Logout from system	Pass	N/A	N/A
02	BR 2.0	2.Dashboard	1. Add Dashboard	TC_varatia_vara_add_dashboar_d_02	Create Dashboard	Pass	N/A	N/A
			2. Go To Dashboard	TC_varatia_vara_go_to_dashboar_d_02	Go To Dashboard	Pass	N/A	N/A
03	BR 3.0	3.Miscellaneous	1. Add Miscellaneous	TC_varatia_vara_add_miscellaneous_03	Create Miscellaneous	Pass	N/A	N/A
			2. View Miscellaneous	TC_varatia_vara_view_miscellaneous_03	View Miscellaneous	Pass	N/A	N/A
04	BR 4.0	4.Tenant	1. Add Tenant	TC_varatia_vara_add_tenant_04	Add tenant to system	Pass	N/A	N/A
			2. View Tenant Details	TC_varatia_vara_view_tenant_04	View Added Tenant	Pass	N/A	N/A
			3. Edit Tenant	TC_varatia_vara_edit_tenant_04	Edit Tenant Details	Pass	N/A	N/A
			4. Delete Tenant	TC_varatia_vara_delete_tenant_04	Delete Specific Tenant	Pass	N/A	N/A
05	BR 5.0	5.Assign Tenant	1. Assign Tenant	TC_varatia_vara_assign_tenan_05	Assign Tenant to Apartment	Pass	N/A	N/A
06	BR 6.0	6.Manage Dashboard	1. Add Rent	TC_varatia_add_rent_tenan_06	Add Tenant Rent	Pass	N/A	N/A
			2. Add Dues	TC_varatia_vara_add_dues_tenan_06	Add Tenant Dues	Pass	N/A	N/A
07	BR 7.0	7.Notice	1. Notice	TC_varatia_vara_add_notice_07	Add Notice to system	Pass	N/A	N/A
			2. Leave Notice	TC_varatia_vara_add_leave_notice_07	Add Leave Notice To system	Pass	N/A	N/A
08	BR 8.0	8. Profile	1. View	TC_varatia_vara_view_08	View Profile	Pass	N/A	N/A

## 5.3. Testing Environment:

### 5.3.1. Web

#### 5.3.1.1. Windows 10

- Chrome Version 106.0.5249.103 (Official Build) (64-bit)
- Firefox 105.0.3 (64-bit)

### 5.3.1. Mobile Browser

#### 5.3.1.2. Android 13

- Chrome Browser
- Google Browser
- 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.		

---

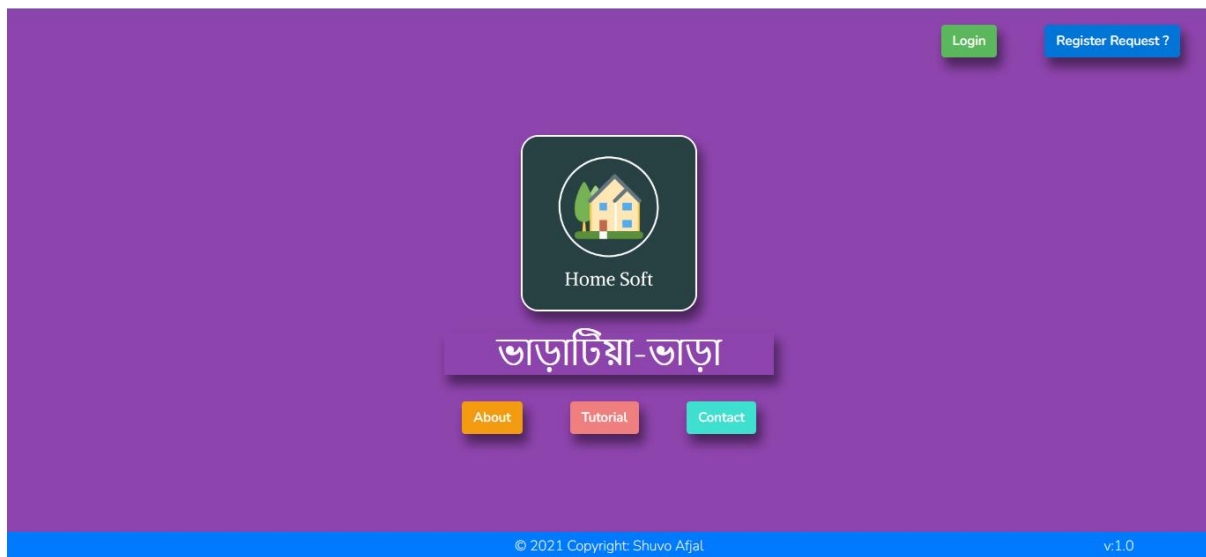


## Chapter 6

### User Manual

## 6. User Manual:

### 6.1. Landing Page:

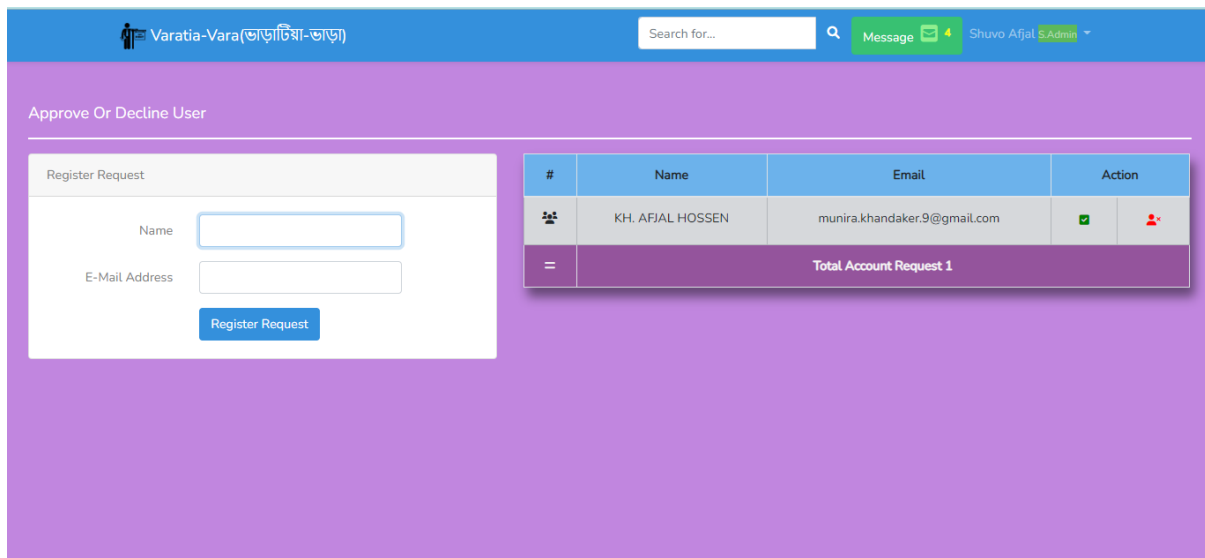


This landing pages. Where registered user can login and unregistered user request to super admin register them. Here they also see tutorial and about our company.

### 6.2. Register Request:

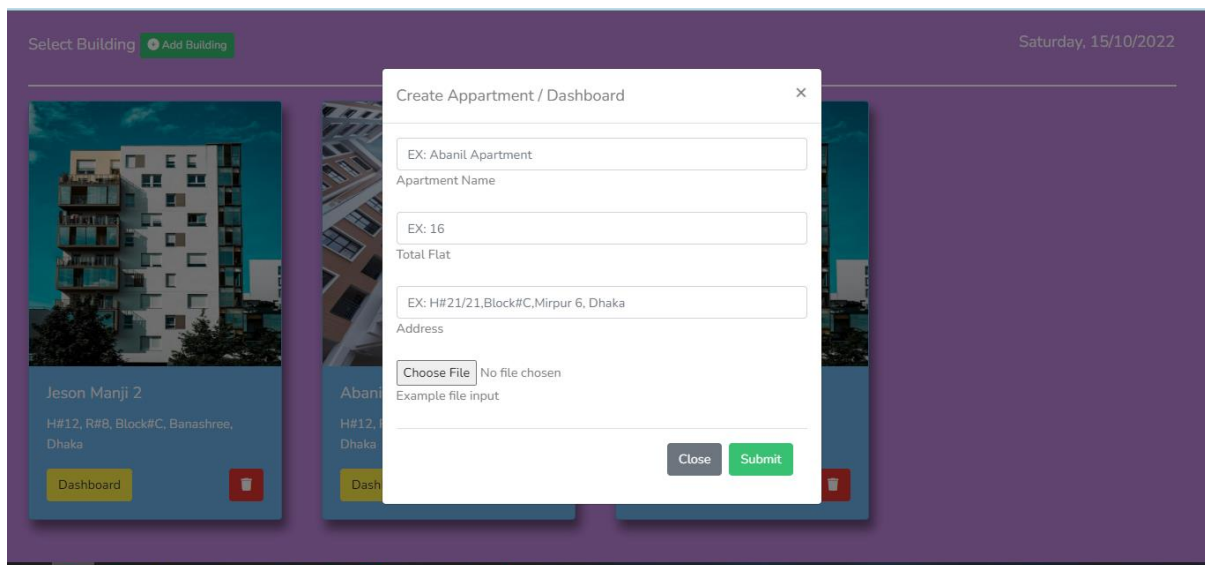
Unregistered user request to super admin registered them.

### 6.3. Approve or Decline Request:



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

### 6.4. Create Dashboard:



User have to create dashboard by giving apartment name, total flat of apartment, address and image.

## 6.5. Dashboard:

The dashboard displays the following information:

- Current Balance:** 70,000/=
- Current Due:** 70,000/=
- Previous Due:** 70,000/=
- Miscellaneous Cost List:** October Total Tk: 18720

Created At	Created By	Task Name	Apartment	Floor/Flat	Estimated Cost	Action
2022-09-30 08:35:00	Shuvo Afjal	Washroom Pipeline	Abanil Apartment	7b	5720	
2022-09-30 09:33:39	Shuvo Afjal	Tiles	Abanil Apartment	7b	10000	
2022-09-30 09:35:08	Shuvo Afjal	Roof	Jeson Manji	7b	3000	

Dashboard will show to home owner his current balance which his collect from rent. Also show current due, previous due, miscellaneous cost and list of current user rent.

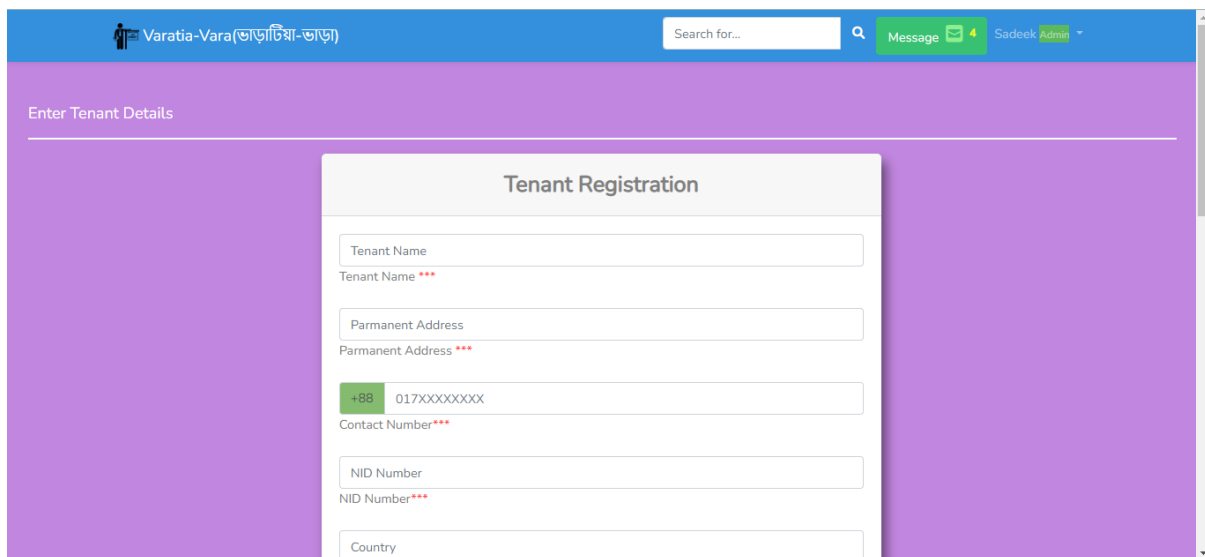
## 6.6. Tenant List:

The tenant list displays the following information:

#	Name	Phone	Address	Father Name	Emergency Contact Name	Emergency Contact Phone	Action
1	Takdeer Mosarf	01558965555	Dhaka	Kamal Hossen	Kamal Hossen	01558965555	
2	Shuvo Afjal	015555555	Dhaka	Md Abdul Mazid	Md Abdul Mazid	015555555	

Tenant list show all tenant from all apartment. Can manage their profile them by edit and delete.

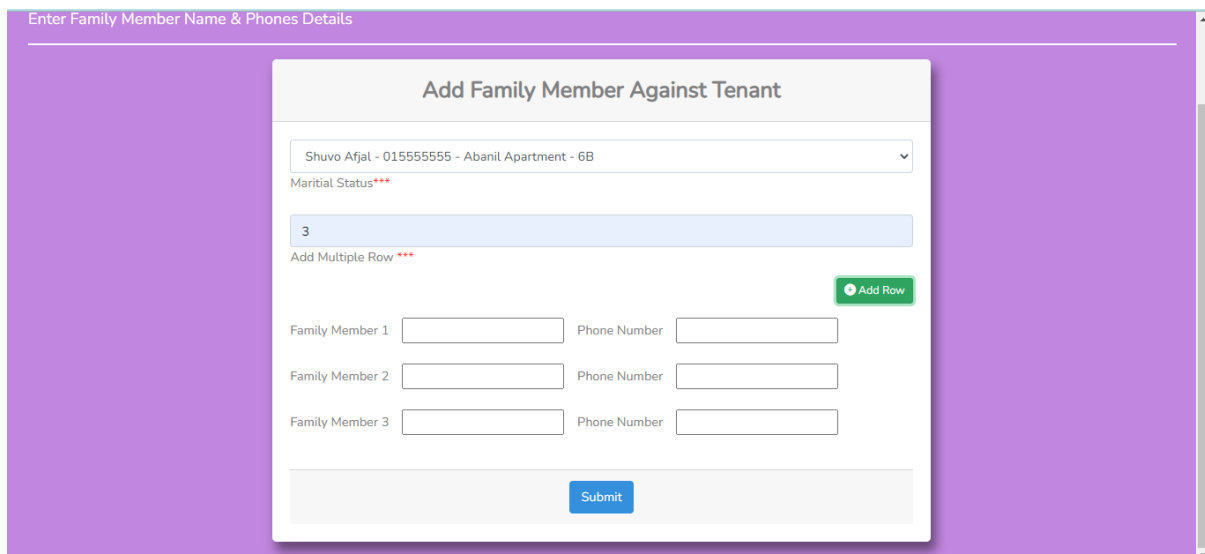
## 6.7. Tenant Registration:



The screenshot shows a web application interface for tenant registration. At the top, there is a blue header with the logo and name 'Varatia-Vara (ভাড়াটিয়া-ভাড়া)', a search bar, a message notification for 'Sadeek Admin', and a user profile dropdown. Below the header, a purple banner contains the text 'Enter Tenant Details'. The main content area features a white form titled 'Tenant Registration' with the following fields: 'Tenant Name' (with a red asterisk), 'Parmanent Address' (with a red asterisk), 'Contact Number' (with a green '+88' icon and a red asterisk), 'NID Number' (with a red asterisk), and 'Country'. The form is set against a purple background.

Tenant registration form help register tenant. Must filled every field by giving accurate data.

## 6.8. Family Member Add:



The screenshot displays the 'Add Family Member Against Tenant' form. At the top, a purple banner says 'Enter Family Member Name & Phones Details'. The form is white and includes a dropdown menu for tenant selection (currently showing 'Shuvo Afjal - 015555555 - Abanil Apartment - 6B'), a 'Maritial Status' field with a red asterisk, a row count field set to '3' with an 'Add Multiple Row' label and a red asterisk, and a green 'Add Row' button. Below these are three rows of input fields for 'Family Member' and 'Phone Number'. A blue 'Submit' button is located at the bottom of the form.

This module helps to add family member against tenant. Who are currently live with tenant.

## 6.9. Assign Tenant to Apartment:

Varatia-Vara (ভাড়াটিয়া-ভাড়া)

Search for... Message 4 Sadeek Admin

### Assign Tenant

#	Name	Phone	Address	Father Name	Emergency Contact Name	Emergency Contact Phone	Action
1	Shuvo Afjal	015555555	Dhaka	Md Abdul Mazid	Md Abdul Mazid	01555555	

### Already Assign To Apartment

#	Name	Phone	Address	Father Name	Emergency Contact Name	Emergency Contact Phone	Apartment Name	Floor/Flat	Action
2	Takdeer Mosarf	01558965555	Dhaka	Kamal Hossen	Kamal Hossen	01558965555	Jeson Manji	1B	

### Add Tenant To Apartment

**Add Tenant To Apartment**

Takdeer Mosarf  
Tenant Name\*\*\*

Jeson Manji  
Select Apartment\*\*\*

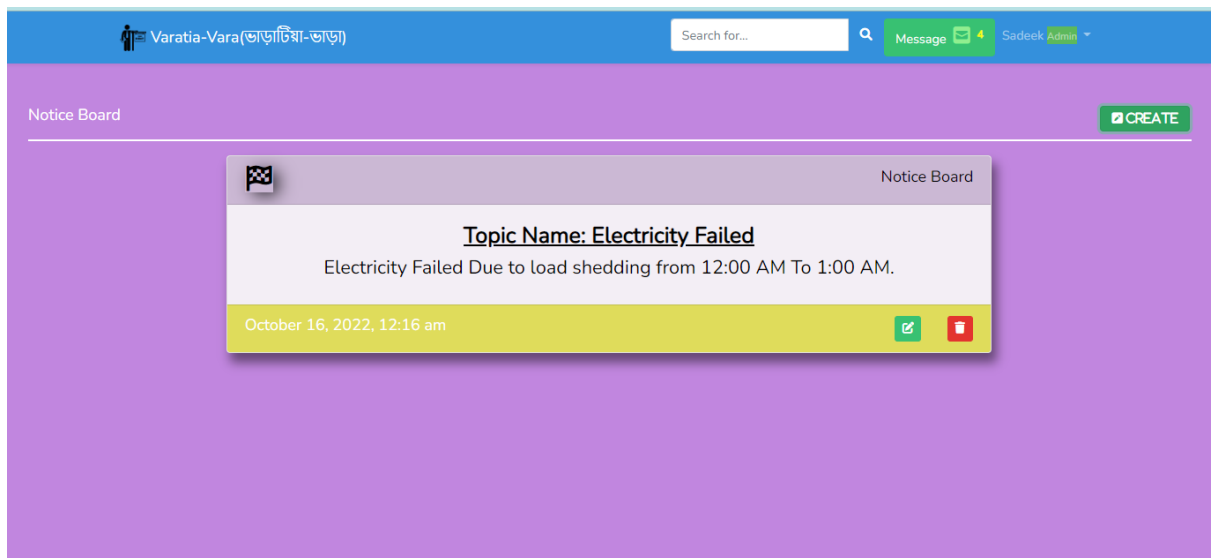
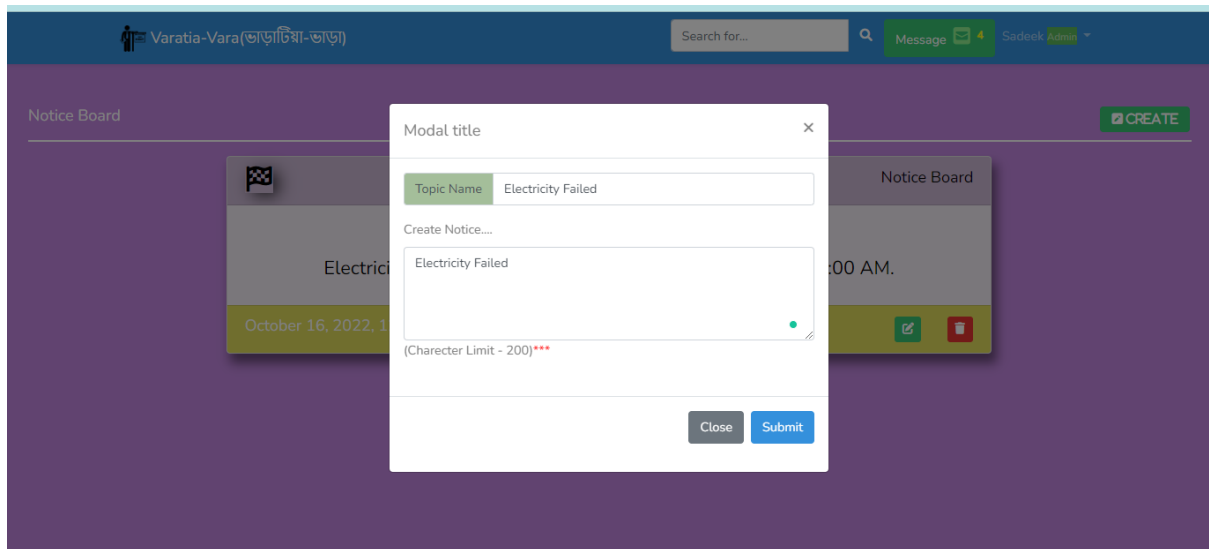
1B  
Floor Number\*\*\*

12000  
Rent\*\*\*

[Submit](#)

Add family member against tenant member. Add dynamic row depend on family member has tenant.

## 6.10. Notice:



Add any notice for tenant.

## 6.10. Leave Notice:

The screenshot shows a web application interface for creating a leave notice. A modal window titled "Create Leave Notice" is open over a blurred background of a "Leave Notice" list. The modal has a close button (X) in the top right corner. It contains two radio buttons: "Temporary" (selected) and "Permantly". Below these are two date pickers: "Select Start Date" with the value "10/22/2022" and "Select End Date" with the value "10/29/2022". There is a text area labeled "Any Message" containing the text "I will be back." and a character limit indicator "(Charecter Limit - 200)\*\*\*". At the bottom of the modal are "Close" and "Submit" buttons. A green "CREATE" button is visible in the top right of the background page.

This screenshot is identical to the one above, showing the "Create Leave Notice" modal window. The "Temporary" option is selected, the start date is 10/22/2022, the end date is 10/29/2022, and the message is "I will be back." The character limit is 200. The "Submit" button is highlighted in blue.

All user can add permeant or temporary leave notice which can see all type user in our system.



## 6.11. Contract Paper:

Varatia-Vara (ভাড়াটিয়া-ভাড়া)

Search for

Message 4 Sadiek Akim

Contract Paper

Automated Contract Paper List

Advanced Payment Date: 21-April-2022, Tenant Name: Mark Otto

Advanced Amount: 10,000 Tk, Action: [Icons]

**Create Contract**

Apartment: Abanil Apartment  
Select Apartment\*\*\*

Tenant: Mark Otto  
Select Tenant\*\*\*

Date: 10/25/2022  
In which month will you get up in the flat?\*\*\*

Payable: 120000  
Home/Flat Rent\*\*\*

Paid: 120000  
Advance Payment\*\*\*

Close Save changes

CREATE

Contract Paper

Automated Contract Paper List

CREATE

Advanced Payment Date	Tenant Name	Phone	Apartment Hold	Flat Hold	Advanced Amount	Action
21-April-2022	Mark Otto	0171777777	Abanil Apartment	6B	10,000 Tk	[Icons]

Contract Paper

Owner Copy

20-Apr-2022, I have receive advance money from jim Hopper for Apartment: Abanil Apartment, Flat: 6B. The actual rent will be increase after two years. He/She was insure that he will follow the generale rules of home rent. From the next Month:August, he will live at Abanil Apartment, Flat: 6B.

signature: \_\_\_\_\_

Jim Hoper (Tenant)

Contract Paper

Tenant Copy

20-Apr-2022, I have receive advance money from jim Hopper for Apartment: Abanil Apartment, Flat: 6B. The actual rent will be increase after two years. He/She was insure that he will follow the generale rules of home rent. From the next Month:August, he will live at Abanil Apartment, Flat: 6B.

After register a new tenant in system must be print tenant contract paper which is generate automatically. By selecting which apartment he will hold and advance payment.

## Chapter 7

### Project Summary

## 7.1. GitHub Link:

### 7.1.1. HTTPS

<https://github.com/shafjal/Vara-varatia.git>

### 7.1.2. SSH

`git@github.com:shafjal/`

### 7.1.3. GitHub CLI

`gh repo clone shafjal/Vara-varatia`

## 7.2. Critical Evolution:

- **Outcomes and The Original Objective**  
our project achieved its intended aims and meet the correct quality standards
- **Purpose achieves**  
Our project successfully handles all data from user and generate a user-friendly dashboard.
- **Human resources used in project**  
There is a good system of training. If any training is required in future, we will arrange it.
- **Identify problems from running project**  
We will monitor it all the time and develop it if necessary.
- **Further project works**  
We will bring some more features in the future.

## 7.3. Limitation:

- **Overall Aim**

Our aim was we try to build a management software which build a better communication between home owner and tenant. Time and money both are valuable. Our system reduce time which are expense in communication between tenant and owner. Some of people get revenue from flat or home rent business. Our system reduce time and increase revenue. Owner get actual report of cost, dues and revenue. Our system builds better communication system between owner and tenant. Owner can send message or attach notice for all tenant at a time. Not need to face to face to communication. Tenant can make leaving approval for leave the home. Tenant can create home related issue. At the end, our system is a management system which can remove all communication gap between tenant and owner.

- **Current status of the project**
  - Can records all the tenant dues & monthly payment
  - Can check current total balance
  - Can make possible expenses list this month
  - Assign Tenant To apartment
  - Can create notice and leave notice

#### **7.4. Obstacles and Achievements:**

- **User Interface**  
We develop our all-module user friendly and mobile friendly. Almost all module is responsive except some module. In future, we will use react is for interactive UI.
- **Scalability**  
We are making our project making good use of computing power and bandwidth. When the load increases additional servers can be added to balance it. In future, for more faster we will use Vue JS.
- **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**  
User need to web access for using system. We are using CSRF token prevent CSRF attacks. Main fact is of our system. If someone try to hack our system, he can easily get some information which is related with user personal information. In future, we use strong security method.

#### **7.5. Future Scope:**

- Automated Report
- Tenant Dashboard
- Multi Authentication
- Message
- Tenant Family Member Add
- NID Verification

## Chapter 8 Reference

## 8. References:

1. Easthope, H. (2014). Making a rental property home. *Housing studies*, 29(5), 579-596.
2. Singh, C., Shelor, R., Jiang, J., & Klein, G. (2004). Rental software valuation in IT investment decisions. *Decision Support Systems*, 38(1), 115-130.
3. Van Duijne, R. J., & Ronald, R. (2018). The unraveling of Amsterdam's unitary rental system. *Journal of Housing and the Built Environment*, 33(4), 633-651.

## Payment Ledger

Student Info	Payment Ledger Summary
<p>Name: Kh. Afjal Hossen</p> <p>ID: 151-35-1011</p> <p>Email: afjal35-1011@diu.edu.bd</p>	<p>Total Payable: <b>841964</b></p> <p>Total Paid: <b>841970</b></p> <p>Total Due: <b>-6</b></p> <p>Total Other: <b>1600</b></p> <p style="color: orange;">Waiver / Scholarship calculation may vary the amounts</p>

## Plagiarism Report

11/5/22, 6:04 PM

Turnitin - Originality Report - 151-35-1011

**Turnitin Originality Report**

Processed on: 05-Nov-2022 18:03 +06  
 ID: 1945275652  
 Word Count: 7835  
 Submitted: 1

151-35-1011 By Kh. Afjal Hossen

Similarity Index	Similarity by Source
<b>30%</b>	Internet Sources: 29% Publications: 3% Student Papers: 12%

7% match (Internet from 04-Apr-2022)  
[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7633/152-35-1285%20%2815\\_%29.pdf?isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7633/152-35-1285%20%2815_%29.pdf?isAllowed=y&sequence=1)

---

4% match (Internet from 25-Oct-2022)  
[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5681/171-35-1817%20%2823\\_%29.pdf?isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5681/171-35-1817%20%2823_%29.pdf?isAllowed=y&sequence=1)

---

3% match (student papers from 11-Feb-2018)  
 Class: Project  
 Assignment: Project  
 Paper ID: [914092425](#)

---

3% match (Internet from 18-Jul-2020)  
<https://cordis.europa.eu/docs/projects/cnect/9/287119/080/deliverables/001-D31SoftwareRequirementsSpecificationandUseCases.pdf>

---

2% match (Internet from 25-Oct-2022)  
[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5674/171-35-1805%20%2822\\_%29.pdf?isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5674/171-35-1805%20%2822_%29.pdf?isAllowed=y&sequence=1)

---

2% match (Internet from 01-May-2016)  
[http://www.fupol.eu/pdf/public\\_deliverables/d3.6-revised-requirements-specification-and-use-cases.pdf](http://www.fupol.eu/pdf/public_deliverables/d3.6-revised-requirements-specification-and-use-cases.pdf)

---

1% match (Internet from 26-Oct-2022)  
[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/20.500.11948/2855/Bookshelf\\_Documentation%20%281%29.pdf?isAllowed=y&sequence=2](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/20.500.11948/2855/Bookshelf_Documentation%20%281%29.pdf?isAllowed=y&sequence=2)

---

1% match (Internet from 26-Oct-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8555/181-35-318%20%2825%25%29.pdf?isAllowed=y&sequence=1>

---

1% match (Internet from 06-Mar-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7367/171-35-1967%20%2825%25%29.pdf?isAllowed=y&sequence=1>

---

1% match (Internet from 25-Oct-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/2404/P12998%20%2828%25%29.pdf?isAllowed=y&sequence=1>

---

1% match (Internet from 06-Aug-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8182/171-35-226%20%2823%25%29.pdf?isAllowed=y&sequence=1>

---

< 1% match (Internet from 12-Jun-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8132/171-35-172%20%2819%25%29.pdf?isAllowed=y&sequence=1>

---

< 1% match (Internet from 07-Jun-2022)  
[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8106/161-35-1586%20%2818\\_%29.pdf?isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/8106/161-35-1586%20%2818_%29.pdf?isAllowed=y&sequence=1)

---

< 1% match (Internet from 02-Mar-2022)  
<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/7385/171-35-1873%20%2c%2022%25.pdf?isAllowed=y&sequence=1>

---

< 1% match (Internet from 25-Oct-2022)  
[http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3809/P15045%20%2828\\_%29.pdf?isAllowed=y&sequence=1](http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/3809/P15045%20%2828_%29.pdf?isAllowed=y&sequence=1)

[https://www.turnitin.com/newreport\\_printview.asp?eq=1&eb=1&esm=10&oid=1945275652&sid=0&n=0&m=2&svr=22&r=41.99291907770733&lang=en...](https://www.turnitin.com/newreport_printview.asp?eq=1&eb=1&esm=10&oid=1945275652&sid=0&n=0&m=2&svr=22&r=41.99291907770733&lang=en...) 1/8