



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

**Booking Party Center and Getting Laundry Service: For a  
Colony Based Area**

By

**Kabir Hossain**  
**152-35-1266**

A thesis submitted in partial fulfillment of the requirement for the degree of  
Bachelor of Science in Software Engineering

**Department of Software Engineering**  
**DAFFODIL INTERNATIONAL UNIVERSITY**

Spring - 2019

# APPROVAL

This **Project/Thesis** titled “**Booking Party Center and Getting Laundry Service : For a Colony Based Area**”, submitted by **Kabir Hossain, 152-35-1266** 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 B.Sc in Software Engineering and approved as to its style and contents.

## BOARD OF EXAMINERS

-----  
**Dr. Touhid Bhuiyan**  
**Professor and Head**

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

**Chairman**

-----  
**K. M. Imtiaz-Ud-Din**  
**Assistant Professor**

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

**Internal Examiner 1**

-----  
**Md. Shohel Arman**  
**Lecturer**

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

**Internal Examiner 2**

-----  
**Mr. Shafiul Hasan**  
**Managing Director**

Vivacom Solution, Dhaka

**External Examiner**

## DECLARATION

We hereby declare that we have taken this thesis under the supervision of **Mr. Iftekharul Alam Efat**, Senior Lecturer, **Department of Software Engineering, Daffodil International University**. We also declare that neither this thesis/project nor any part of this has been submitted elsewhere for award of any degree.



---

**Kabir Hossain**  
ID: 152-35-1266  
Batch : 17<sup>th</sup>  
Department of Software Engineering  
Faculty of Science & Information Technology  
Daffodil International University

Certified by:



---

**Mr. Iftekharul Alam Efat**  
**Senior Lecturer**  
Department of Software Engineering  
Faculty of Science & Information Technology  
Daffodil International University

©Daffodil International University

## Acknowledgement

First of all, I am grateful to the Almighty Allah for making me eligible to complete this project. Then I would like to thank my supervisor **Mr. Iftekharul Alam Efat**, Senior Lecturer, Department Of Software Engineering. I am extremely grateful and indebted to him for his expert, sincere and valuable guidance and encouragement extended to me.

I would like to thank them who were helped in my project by their very important suggestions without their passionate participation and input; the project could not be successfully conducted. I take this opportunity to record my sincere thanks to all the faculty members of the Department of Software Engineering for their help and encouragement.

## Executive Summary

The time when we started the development part it's in January. Like other projects, it is important to work hard, endurance, dedication, and concentration to complete the project. There are many reasons for the capitalization of the project development, such as the requirements of the stakeholder to be properly filled.

If you follow the requirements analysis properly, then it helps a lot in the development of the project. We first analyze our project requirements and then we do the next step design specification.

An application system database plays an important role. For this reason, we are focused on creating a database design. We have designed the drawing table to stay table with the right relationship. Admin part can also be called part of the maintenance. Admin plays a big role in our system. So the next part user interfaces. The user interface is easy to create if any user can easily understand. After that, we check everything again and go to the main functionality of the project.

Developing a project is not an easy task. But building the project is not the and actually. At the end of complete the project, you have to make sure that your project functionality works fine. For that, you have to come in the testing part, its part of quality assurance. The responsibility of quality assurance is to find the vulnerability of the system. If any bug can be found before the system release then there is a change to fix that bug. So testing the project we have assured the quality of the project

## Table of Contents

APPROVAL.....	ii
Acknowledgement.....	iv
Executive Summary .....	v
Chapter 1.....	1
1.1 Project Overview .....	2
1.2 Project Purpose.....	3
1.2.1 Background.....	3
1.2.2 Benefits & Beneficiaries.....	3
1.2.3 Goals .....	4
1.3 Stakeholders .....	4
1.4 Proposed System Model .....	5
1.5 Project Schedule .....	5
1.5.1 Gantt Chart .....	6
1.5.2 Release Plan or Milestone .....	6
Chapter 2.....	8
2.1 Functional Requirements .....	8
2.1.1 Login for Apartment Owner .....	10
2.1.2 Apartment Owner post for laundry service .....	11
2.1.3Apartment Owner View Laundry Service details .....	11
2.1.4 Apartment Owner booking party center .....	11
2.1.5 Apartment Owner View party center details .....	11
2.1.6 Apartment Owner View previous history.....	12
2.1.7 Apartment Owner can View recent post.....	12
2.1.8 Apartment Owner can view their own profile .....	12
2.1.9 Apartment Owner can edit their profile.....	12
2.1.10 Login for Admin .....	13
2.1.11 Apartment Owner Registration .....	13
2.1.12 Admin View History .....	13
2.1.13Admin View Recent Post .....	13

2.1.14 Admin Delete Apartment Owner profile .....	14
2.1.15 Party center booking confirmation .....	14
2.1.16 Update party center details .....	14
2.1.17 Update Laundry Service details .....	14
2.2 Data Requirements .....	15
2.3.1 Speed and Latency Requirements .....	15
2.3.2 Precision or Accuracy Requirements .....	16
2.3.3 Capacity Requirements.....	16
2.4 Dependability Requirements .....	16
2.4.1 Reliability Requirements .....	17
2.4.2 Availability Requirements .....	17
2.4.3 Robustness or Fault-Tolerance Requirements .....	17
2.4.4 Safety-Critical Requirements.....	18
2.5 Maintainability & Supportability Requirements .....	18
2.5.1 Maintenance Requirements .....	18
2.5.2 Supportability Requirements .....	19
2.5.3 Adaptability Requirements .....	19
2.5.4 Scalability or Extensibility Requirements .....	19
2.6 Security Requirements .....	19
2.6.1 Access Requirements .....	19
2.6.2 Integrity Requirements .....	20
2.6.3 Privacy Requirements .....	20
2.7 Usability and Human-Interaction Requirements .....	20
2.7.1 Ease of Use Requirements .....	20
2.7.2 Personalization and Internationalization Requirements .....	20
2.7.3 Understandability and Politeness Requirements .....	21
2.7.4 Accessibility Requirements .....	21
2.7.5 User Documentation Requirements .....	21
2.7.6 Training Requirements .....	21
2.8 Look and Feel Requirements.....	21
2.8.1 Appearance Requirements .....	22
2.8.2 Style Requirements .....	22

2.9 Operational and Environmental Requirements.....	22
2.9.1 Expected Physical Requirements .....	22
2.9.2 Requirements for Interfacing with Adjacent Systems.....	23
2.9.3 Release Requirements .....	23
2.10 Legal Requirements .....	23
2.10.1 Compliance Requirements.....	23
2.10.2 Standards Requirements .....	23
Chapter 3.....	24
3.1 Use Case Diagram .....	25
Use Case Description.....	27
3.1.1 Login for Apartment owner .....	27
3.1.2 Apartment owner post for laundry service .....	27
3.1.3 Apartment owner view laundry service details.....	28
3.1.4 Apartment owner booking party center .....	28
3.1.5 Apartment owner view party center details .....	29
3.1.6 Apartment owner sees previous history .....	29
3.1.7 Apartment owner sees recent post.....	30
3.1.8 Apartment owner can view profile.....	30
3.1.9 Apartment owner can edit their profile .....	31
3.1.10 Get confirmation notification.....	31
3.1.11 Login for Admin .....	32
3.1.12 Apartment owner registration .....	32
3.1.13 Party center booking confirmation.....	33
3.1.14 Admin view history.....	33
3.1.15 Admin view recent post.....	34
3.1.16 Admin can delete profile .....	34
3.1.17 Admin update's party center details.....	35
3.1.18 Admin update's laundry service details.....	35
3.1.19 Admin Can Edit Profile .....	36
3.2 Activity Diagram.....	36
3.2.1 Login for Apartment Owner .....	37
3.2.2 Apartment Owner Post for Laundry Service .....	38



3.2.3 Apartment Owner view laundry service details .....	39
3.2.4 Apartment Owner post for booking party center .....	40
3.2.5 Apartment Owner view party center details .....	41
3.2.6 Apartment Owner can view profile .....	42
3.2.7 Apartment Owner sees previous history .....	43
3.2.8 Apartment Owner sees recent post .....	44
3.2.9 Apartment Owner can edit his profile .....	45
3.2.10 Login for Admin .....	46
3.2.11 Admin create apartment owner registration .....	47
3.2.12 Admin sees previous history .....	48
3.2.13 Admin sees recent post .....	49
3.2.14 Admin can edit and delete apartment owner profile .....	50
3.2.16 Party center booking confirmation .....	51
3.2.17 Admin can update party center details .....	52
3.3 Sequence Diagrams .....	53
3.3.1 Login for Apartment Owner .....	53
3.3.2 Apartment Owner can view their profile .....	54
3.3.3 Apartment Owner sees his previous post history .....	54
3.3.4 Apartment Owner sees their recent post .....	55
3.3.5 Apartment Owner edits their profile .....	55
3.3.6 Apartment Owner post for laundry service .....	56
3.3.7 Apartment Owner post for booking party center .....	56
3.3.8 See Laundry service details .....	57
3.3.9 See Party Center details .....	57
3.3.10 Login for Admin .....	58
3.3.11 Admin registration for Apartment Owner .....	58
3.3.12 Admin view Apartment Owner's post history .....	59
3.3.13 Admin view Apartment Owner's recent post .....	59
3.3.14 Admin edits Apartment Owner/maid-servant profile .....	60
3.3.15 Admin deletes Apartment Owner/maid-servant profile .....	60
Chapter 4 .....	61
4.1 Development tools and technology .....	62

4.1.1 User Interface Technology .....	62
4.1.2 Implemented tools and platform .....	63
4.2 Class Diagram .....	64
4.3 Database Design Diagram.....	65
Chapter 5.....	66
5.1 Testing Features .....	67
5.1.1 Feature to be tested .....	67
5.2 Testing Strategy.....	68
5.2.1 Test approach.....	68
5.2.2 Pass / Fail Criteria .....	70
5.3 Testing Schedule .....	70
5.4 Trace Ability Matrix .....	71
5.5 Testing Environment .....	72
5.6 Test Cases .....	73
5.6.1 Log In.....	74
5.6.2 Apartment Owner Post for Laundry Service .....	75
5.6.3Apartment Owner Booking Party Center .....	75
5.6.4 Apartment owner view laundry service details.....	76
5.6.5 Apartment owner view party center booking details.....	76
5.6.6Get party center booking confirmation message.....	77
5.6.7See previous history .....	78
5.6.8See recent post .....	79
5.6.9View own profile .....	80
5.6.10Edit Profile.....	81
5.6.11 Login for Admin .....	82
5.6.12Confirm or Reject party center booking request .....	83
5.6.13Update party center and laundry service details .....	84
5.6.14 Registration apartment owner profile.....	85
5.6.15Admin view previous history.....	86
5.6.16Admin view recent post.....	87
5.6.17 Admin Edit Profile.....	88
5.6.18 Admin Delete Profile .....	89

Chapter 6.....	90
6.1 Apartment owner login page .....	91
6.2 Home page (Apartment owner) .....	92
6.3 Apartment owner post for laundry service .....	93
6.4 Apartment Owner post for booking party center .....	94
6.5 Apartment Owner can view their profile.....	95
6.6 Admin login page.....	96
6.7 Apartment owner registration .....	97
6.8 Apartment owner list & add apartment owner .....	98
Chapter 7.....	99
7.1 GitHub Link .....	100
7.2 Project Summary .....	100
7.3 Limitations .....	100
7.4 Obstacles and Achievements.....	101
7.5 Future Scope.....	101
7.6 References .....	102

## List of Figures

Figure 1.1: Proposed system model.....	14
Figure 1.2: Gantt chart.....	15
Figure 3.1: Use Case diagram for “Apartment Colony Service System”.....	34
Figure 3.2: Use Case diagram for “Apartment Colony Service System”.....	35
Figure 3.3: Login for apartment owner.....	46
Figure 3.4: Apartment owner post for laundry service.....	47
Figure 3.5: Apartment owner view laundry service details.....	48
Figure 3.6: Apartment owner post for party center booking.....	49
Figure 3.7: Apartment owner view party center booking details.....	50

..Figure 3.8: Apartment Owner can view profile.....	51
Figure 3.9: Apartment Owner sees previous history.....	52
Figure 3.10: Apartment owner see recent post.....	53
Figure 3.11: Edit apartment owner profile.....	54
Figure 3.12: Admin login.....	55
Figure 3.13: Create apartment owner registration.....	56
Figure 3.14: Admin sees previous history.....	57
Figure 3.15: Admin sees recent posts.....	58
Figure 3.16: Admin Can Edit/ Delete apartment owner profile.....	59
Figure 3.17: Party center booking confirmation.....	60
Figure 3.18: Admin can update party center details.....	61
Figure 3.19: Sequence Diagram for Apartment owner Login.....	62
Figure 3.20: Sequence Diagram for Apartment owner view his/her Profile.....	63
Figure 3.21: Sequence Diagram for Apartment owner sees his/her post History.....	63
Figure 3.22: Sequence Diagram for Apartment owner sees his/her recent post.....	64
Figure 3.23: Sequence Diagram for Apartment owner edits their profile.....	64
Figure 3.24: Sequence Diagram for Apartment owner post for laundry service.....	65
Figure 3.25: Sequence Diagram for Apartment owner post for party center booking.....	65
Figure 3.26: Sequence Diagram for Apartment owner see laundry service details.....	66
Figure 3.27: Sequence Diagram for Apartment owner see laundry service details.....	66
Figure 3.28: Sequence Diagram for Admin Login.....	67
Figure 3.29: Sequence Diagram, registration for Apartment Owner.....	67
Figure 3.30: Sequence Diagram, Admin view Apartment Owner’s post history.....	68
Figure 3.31: Sequence Diagram, Admin view Apartment Owner’s recent post.....	68
Figure 3.32: Sequence Diagram, Admin edits Apartment Owner profile.....	69

Figure 3.33: Sequence Diagram, Admin deletes Apartment Owner profile.....	69
Figure 4.1: Class Diagram.....	73
Figure 4.2: Database Diagram.....	74
Figure 6.1: Login page.....	100
Figure 6.2: Home page of apartment owner.....	101
Figure 6.3: Apartment owner post for laundry service.....	102
Figure 6.4: Apartment owner post for booking party center.....	103
Figure 6.5: Apartment owner view their own profile.....	104
Figure 6.6: Admin login page.....	105
Figure 6.7: Apartment owner registration form.....	106
Figure 6.8: Apartment owner list & add apartment owner.....	107

# Chapter 1

## Introduction

## 1.1 Project Overview

Day by day people are getting busy with their life. They are traveling, eating, shopping, exercise, sporting and, many other things. But sometimes they can't manage time for doing their necessary task. Here the good parts of technology that can provide a good and easier lifestyle for people.

Technology is not only for entertainment and communication. Technologies can solve the various problems of peoples in their life simultaneously. Think about the people daily lifestyle they need to be complete some task as usual.

Also, think about of laundry service, people have to go to a laundry service store and put their clothes for laundering or washing and get a serial token number to get back their clothes. Sometimes they lost their token number and sometimes they forget to bring the clothes on the time.

People have many celebration days with their friends and family. Booking a party center to nearby you that's meets your sufficient needs is also a time-consuming task. Sometimes people have family program therefor there need some security that's why a party center nearest there home is very important. And also sometimes outside environment have some problem like strike, traffic jam, agitation etc. This is why people need a party center within their area is blessed.

For the reason, we are going to make an application making this process easy and simple. By using this system people will not face any hassle to finding maid-servant. Also people will get the benefit of laundry services, for laundering their clothes they not have to go outside. And booking party center it easy process. They just need a request for booking party center. This process would be done within a few seconds.

## 1.2 Project Purpose

The main purpose of this project named “Booking Party Center and Getting Laundry Service” is to make an automation system which might be helpful for many users from a different perspective by solving their few problems. So that’s why we are going to develop such a project.

### 1.2.1 Background

I recognize that, condominium building is increasing day by day in our country. In those condominium apartment, lot of people are living there, they need some facilities as usual. So considering that, I have brainstormed and thinking that how we give some service based on this specific condominium people. And we select some facilities that make their life easier.

### 1.2.2 Benefits & Beneficiaries

In this applications would be beneficial for some point of view. Now, I am mentioning those below:

- Our system helps apartment owner to find and get a maid-servant easily
- It helps maid-servant to get a job easily
- It would be helpful for apartment owner for laundering their clothes
- Apartment owner don’t need to come into laundry store
- It is also helpful for apartment owner to book a party center
- It helps apartment owner to get some service within a moment
- This application makes authentication and trustful
- The system also going to reduce wasting time
- Apartment owner can choose the maid-servant as their want
- Apartment owner can rating to the maid-servant based on their performance



### 1.2.3 Goals

The main goal of this project is to develop an application based on web. The proposed system has some modules. The ultimate goal of this application is help the people to get some facility easily, simply and effectively because we really believed in quality product and consume the time and reduce the hassle for getting some facilities.

## 1.3 Stakeholders

The stakeholders of the system are:

- Apartment Owner
- Maintainers (Admin)
- Party Center
- Laundry Store

**Apartment Owner:** Apartment Owner can get some service like recruit maid-servant. For recruit maid-servant they have to do submit a post. Then admin send some of maid-servant list to apartment owner and apartment owner choose and confirm one which is good fits to him. Also they can book a party center and laundering their clothes. For booking party center they send a party center booking request.

**Maintainers (Admin):** Admin plays a huge role in this system. Admin selects maid-servant against apartment owner request and send them to apartment owner. When any party center request comes to admin, then admin confirms or reject the booking requests. Admin take all action of laundry service.

**Party Center:** Apartment Owner submit a party center booking request and then admin check the date if the date is available then admin confirm the request if date is not available then reject the request.

**Laundry Store:** Apartment owner post for laundering their clothes with cloths quantity, receiving date etc. Then admin collect the clothes their way and process the cloth and then return those.

## 1.4 Proposed System Model

Before going to develop a system it is very important to have a system model. We have already prepared a system model. This model will clarify our proposed system in brief.

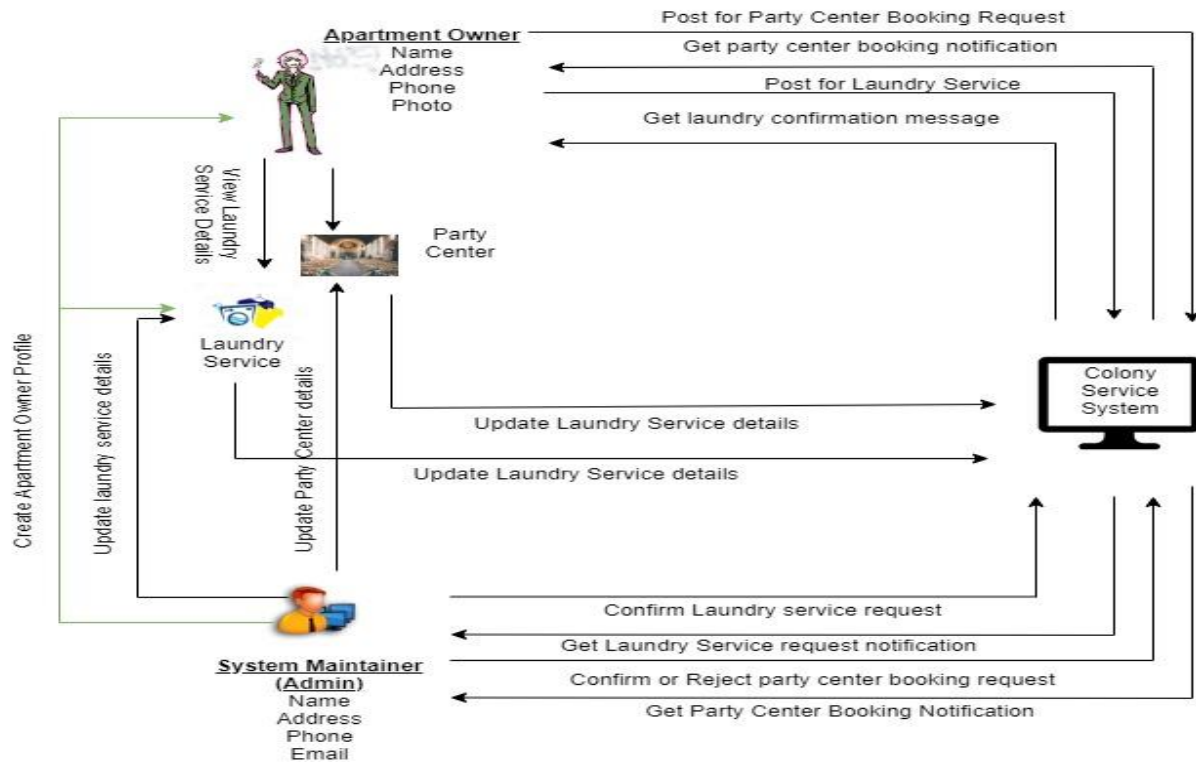


Figure 1.1: Proposed system model

## 1.5 Project Schedule

A proper project scheduling is helping to complete the project successfully and on time. Project scheduling makes a perfect work flow to step by step by timeframe. So project scheduling is must for a system.

### 1.5.1 Gantt Chart

Gantt chart is mainly a production control tools. It remained us to complete our assigned tasks within a certain period of time. For developing software, it is mostly used. Now I will show a Gantt chart for our project:

Activities		M1 H1	M1 H2	M2 H1	M2 H2	M3 H1	M3 H2	M4 H1	M4 H2	M5 H1	M5 H2	M6 H1	M6 H2	M7 H1	M7 H2
<b>Planning</b>	Ideas	█													
	Problem definition	█													
	Proposal planning		█												
<b>Requirements</b>	Requirement specification				█	█	█								
	Requirement analysis					█	█								
<b>QA-1</b>	Quality assurance						█	█							
<b>System design</b>	Sketching							█	█						
	Design specification							█	█	█					
	Database design								█						
	Post for laundry service								█	█					
	Booking party center									█	█				
<b>Implementation-1</b>	Booking party center							█	█	█					
<b>QA-2</b>	Test cases								█						
<b>Implementation-2</b>	Laundry Service								█	█	█				
<b>Testing</b>	Unit testing								█	█	█				
	Black box testing									█	█	█	█		
<b>Delivery</b>	Software release													█	█
Scheduled Time						█									
Buffered Time						█									
M	Month	H1		First Half				H2				Second Half			

Figure 1.2: Gantt chart

### 1.5.2 Release Plan or Milestone

The release plan or milestones are given below:

Activities	Duration in Month	Total Month
------------	-------------------	-------------

Brainstorming	First-half( First month)	0.5(Half month)
Problem identification	First month	1
Requirement specification	Second-half (Second month) & Third month	1.5
Requirement analysis	Third month	1
Sketching	Fourth month	1
Design specification	Fourth month & First-half( Fifth month)	1.5
Database design	Second-half (Fourth month)	0.5(Half month)
Post for booking party center	Fourth month & First-half (Fifth month)	1.5
Post for laundry service	Second-half( Fourth month) & First-half (Fifth month)	1
Quality assurance	Second-half (Third month) & First-half( Fourth month)	1
Test case	First-half (Fifth month)	0.5(Half month)
Unit testing	Second-half ( Fourth month) & Fifth month	1.5
Black-box testing	Fifth month & Sixth month	2
Software release	Seventh month	1

## **Chapter 2**

### **Software Requirements Specification (SRS)**

#### **2.1 Functional Requirements**

Functional requirements refer to the function which is must be belong to the system. Functional requirements are mandatory to perform the software system.

There is no system without functional requirements. Now, we are going to discuss functional requirements for our project.

## Priority Chart:

A Prioritization Matrix is a useful technique to identify which problems are the most important to work on solving first. The Matrix helps you rank problems or issues generated through brainstorming. Using Priority Chart we can identify which function should get High Priority and which one should be Medium and which one is Low.

In Priority Matrix there is two part one is “Important” another one is “Urgent”.  
If any function is:

- Important also Urgent = High Priority
- Important but Not Urgent = Medium Priority
- Not Important but Urgent = Low Priority
- Not Important and Not Urgent = Ignore It

<b>Important\Urgent</b>	<b>Yes</b>	<b>No</b>
<b>Yes</b>	High Priority	Medium Priority
<b>No</b>	Low Priority	Ignore

Priority of our system features based on Priority Chart:

<b>Freq. No:</b>	<b>Important</b>	<b>Urgent</b>	<b>Priority</b>
<b>FR01</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR02</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR03</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>
<b>FR04</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR05</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>
<b>FR06</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>
<b>FR07</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR08</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR09</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>
<b>FR10</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR11</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR12</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>
<b>FR13</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR14</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR15</b>	<b>Yes</b>	<b>Yes</b>	<b>High</b>
<b>FR16</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>
<b>FR17</b>	<b>No</b>	<b>Yes</b>	<b>Medium</b>

### 2.1.1 Login for Apartment Owner

<b>FR01</b>	<b>Login for Apartment Owner</b>		
<b>Description</b>	Admin created an apartment owner account and provide email & password to apartment owner. Then, apartment owner login into the application through email & password that are provided by admin.		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	High

### 2.1.2 Apartment Owner post for laundry service

<b>FR02</b>	<b>Apartment Owner post for laundry service</b>		
<b>Description</b>	Apartment Owner can post for laundering their clothes. Apartment Owner needed a laundry service for laundering clothes. For getting laundry service Apartment Owner need to submit a request post by clicking “Laundry” button for getting this service.		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	High

### 2.1.3 Apartment Owner View Laundry Service details

<b>FR03</b>	<b>Apartment Owner View Laundry Service details</b>		
<b>Description</b>	Apartment owner can see the Laundry Service details		
<b>Stakeholders</b>	Apartment owner	<b>Priority</b>	Medium

### 2.1.4 Apartment Owner booking party center

<b>FR04</b>	<b>Apartment Owner Booking party center</b>		
<b>Description</b>	For booking party center, Apartment Owner have to do fill up a field with require information and then submit the request for booking party center.		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	High

### 2.1.5 Apartment Owner View party center details

<b>FR05</b>	<b>Apartment Owner View party center details</b>		
<b>Description</b>	Apartment owner can see party center details		
<b>Stakeholders</b>	Apartment owner	<b>Priority</b>	Medium



### 2.1.6 Apartment Owner View previous history

<b>FR06</b>	<b>Apartment Owner View previous history</b>		
<b>Description</b>	Apartment Owner can able to see all previous history. After a period of session the all activity of apartment owner, goes to previous history. Suppose, after assigned a maid-Servant to , apartment owner then the post goes to previous history		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	Medium

### 2.1.7 Apartment Owner can View recent post

<b>FR07</b>	<b>Apartment Owner View recent post</b>		
<b>Description</b>	Apartment Owner also can able to see their recent post. The post that is performed recently by apartment owner and not complete yet then those post goes to recent post.		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	High

### 2.1.8 Apartment Owner can view their own profile

<b>FR08</b>	<b>Apartment Owner can view their own profile</b>		
<b>Description</b>	Apartment Owner can see their Profile. There is an option on menu bar called Profile and click on this button Apartment Owner can see their Profile		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	High

### 2.1.9 Apartment Owner can edit their profile

<b>FR09</b>	<b>Apartment Owner edit their profile</b>		
<b>Description</b>	Apartment Owner can Edit their Profile. For edit their profile firstly they have to go their profile through clicking profile button on menu bar then beside the profile there is an option for editing their Profile called “Edit”. And click the Edit button apartment owner can edit their own profile.		
<b>Stakeholders</b>	Apartment Owner	<b>Priority</b>	Medium

### 2.1.10 Login for Admin

<b>FR10</b>	<b>Login for Admin</b>		
<b>Description</b>	An admin email and password to be saved into database that provide to admin. And admin login the Application through using this saved email & password.		
<b>Stakeholders</b>	<b>Admin</b>	<b>Priority</b>	<b>High</b>

### 2.1.11 Apartment Owner Registration

<b>FR11</b>	<b>Apartment Owner Registration</b>		
<b>Description</b>	Admin can create the apartment owner profile with Valid require information like Name, Phone number, Address, Email etc. Admin must be login into the Application for perform this task.		
<b>Stakeholders</b>	<b>Admin</b>	<b>Priority</b>	<b>High</b>

### 2.1.12 Admin View History

<b>FR12</b>	<b>View History</b>		
<b>Description</b>	All of previous task that was completed like assigning Maid-Servant, Party center booking and other are goes to previous history. Admin can sell All previous history by clicking Previous History button.		
<b>Stakeholders</b>	<b>Admin</b>	<b>Priority</b>	<b>Medium</b>

### 2.1.13 Admin View Recent Post

<b>FR13</b>	<b>Admin View Recent Post</b>		
<b>Description</b>	Admin able to see all recent post and take the decision for going next step. The recent post that is requested by apartment owner and no action taken yet against of this recent post is going to recent post.		
<b>Stakeholders</b>	<b>Admin</b>	<b>Priority</b>	<b>High</b>

### 2.1.14 Admin Delete Apartment Owner profile

<b>FR14</b>	<b>Delete Profile</b>		
<b>Description</b>	Only admin can delete all type of profile such as apartment owner profile, Maid-Servant profile.		
<b>Stakeholders</b>	Admin	<b>Priority</b>	High

### 2.1.15 Party center booking confirmation

<b>FR15</b>	<b>Party center booking confirmation</b>		
<b>Description</b>	Admin getting party center booking notification from apartment owner, then admin check the schedule and other info and confirm the booking.		
<b>Stakeholders</b>	Admin	<b>Priority</b>	High

### 2.1.16 Update party center details

<b>FR16</b>	<b>Update party center details</b>		
<b>Description</b>	Admin update or add party center information		
<b>Stakeholders</b>	Admin	<b>Priority</b>	Medium

### 2.1.17 Update Laundry Service details

<b>FR17</b>	<b>Update party center details</b>		
<b>Description</b>	Admin update or add Laundry service information		
<b>Stakeholders</b>	Admin	<b>Priority</b>	Medium

## 2.2 Data Requirements

For achieve the model objectives data requirements prescribed scope and level of detail required. What data is required for building the model can be known by gathering system data. Instead of gathering general system data it's better to gathering specific system data. An overall process flow can provide more detailed information. A process flow also helps to build a model building process.

We focus some of main points. Such as:

- Route locations and their capacity
- Routing decision
- The type of entity and their attributes
- Types of resources
- Quantity of data
- What activity
- Entities movement

## 2.3 Performance Requirements

Defined performance specifications incorrectly can lead to disputes between clients and suppliers.

### 2.3.1 Speed and Latency Requirements

This is rare, in some case the response time will be dictated by legal requirements. Response time is important for user satisfaction. If any system loaded more than 10 seconds then user avoid them.

<b>RT1</b>	<b>View and post result should be displayed less than 4 seconds</b>		
<b>Description</b>	When the Apartment Owner or Admin want to see any information then the view result must be show less than 4 second.		
<b>Stakeholders</b>	Apartment Owner, Admin	<b>Priority</b>	Medium

### 2.3.2 Precision or Accuracy Requirements

Always show the accurate result to all type of user. Wrong information arise misunderstanding.

<b>PAR1</b>	View Result must be accurate		
<b>Description</b>	When the Apartment Owner or Admin want to see any information then the view result must accurate.		
<b>Stakeholders</b>	Apartment Owner, Admin	<b>Priority</b>	Medium

### 2.3.3 Capacity Requirements

The system should be capable of supporting a certain amount of customers and a certain amounts of interactions.

<b>RT1</b>	Supporting a certain amount of customers and a certain amounts of interactions.		
<b>Description</b>	Suppose the system must support 10,000 Apartment Owner and 2000 interactions per day.		
<b>Stakeholders</b>	Apartment Owner, Admin, Maid-Servant	<b>Priority</b>	Medium

## 2.4 Dependability Requirements

The dependability is measured based on four dimensions. Such as:

- Reliability
- Availability
- Durability
- Security
- Safety

Dependability is the ability to provide services that can defensibly be trusted within a time-period.

### 2.4.1 Reliability Requirements

Reliability is the probability of failure-free operation of a system over a specified time within a specified environment for a specified purpose.

<b>DR1</b>	System must be failure-free		
<b>Description</b>	The system must be failure-free operation system and technical error free over a specified time within this type of environment.		
<b>Stakeholders</b>	Apartment Owner, Admin	<b>Priority</b>	High

### 2.4.2 Availability Requirements

<b>AR1</b>	The ability of the system to deliver service when requested		
<b>Description</b>	The system must be available on 24 X 7. The ability of the system to deliver service when requested		
<b>Stakeholders</b>	Apartment Owner, Admin	<b>Priority</b>	High

### 2.4.3 Robustness or Fault-Tolerance Requirements

Ensure the system can cope with error during execution. Also ensure that the system continue properly even of the failure of some of its components.

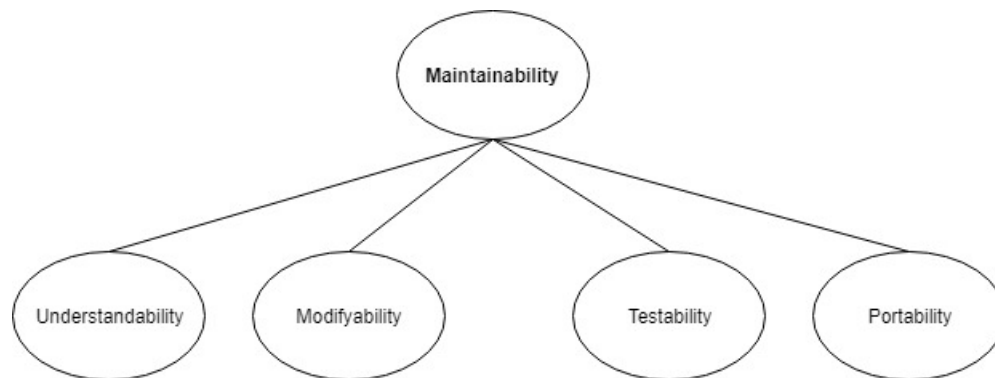
<b>RFT1</b>	Ensure that system can handle all user access or any external tolerance.		
<b>Description</b>	The system can cope with error during execution. Also ensure that the system continue properly even of the failure of some of its components		
<b>Stakeholders</b>	Apartment Owner, Admin	<b>Priority</b>	High

## 2.4.4 Safety-Critical Requirements

There are no safety-critical requirements in our project.

## 2.5 Maintainability & Supportability Requirements

Maintenance is the activity of modifying a software product after initial delivery.



### 2.5.1 Maintenance Requirements

Its main purpose is to modify and update software application after delivery to correct faults and to improve performance. Ensure the all type of maintenance requirements

- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance
- Preventive Maintenance

## 2.5.2 Supportability Requirements

Software Supportability is the capability of supporting a software system over its whole product life. It covers the following key aspects associated to the system

- Operation (install, loading or unloading, configuration, error recovery and execution)
- Logistics Management (once a new baseline has been produced)
- Modification (fixing bugs, or adding/changing functionality due to changing user needs)

## 2.5.3 Adaptability Requirements

There are no adaptability requirements in our system software

## 2.5.4 Scalability or Extensibility Requirements

Make sure that the system can adapt easily with new functionalities, interfaces, devices and new input types.

## 2.6 Security Requirements

System security is most important requirements. For lack of security the data would be stealing, manipulating data and causing denial of service.

Some security related categories

- Permission to access data
- Verification
- Securing information
- Security policies

### 2.6.1 Access Requirements

Reduce unauthorized access there have some barrier. There remain some authentication and authorization techniques.

- Strong Access Control Measures
- Maintain an Information Security Policy
- Regularly Monitor and Test Networks



- Build and Maintain a Secure Network

### **2.6.2 Integrity Requirements**

Ensure that data and communications are not intentionally corrupted via unauthorized creation, deletion, modification.

We use Hash value for sensitive data.

### **2.6.3 Privacy Requirements**

Privacy requirements are mostly needed for a system. In our system we associate the Users and admin area and everyone can see only the area that is associated with them.

## **2.7 Usability and Human-Interaction Requirements**

The system can be failed for Usability. User Experience is one of the most important factors to any system. The system must be easy to use, easy to understand and easy to learn.

### **2.7.1 Ease of Use Requirements**

Our system is very easy to use and easy to understandable. There is no long process to complete a task. User can complete their action with a few steps.

### **2.7.2 Personalization and Internationalization Requirements**

In our Project there are no personalization and internationalization requirements.

### **2.7.3 Understandability and Politeness Requirements**

We made this system based on a targeted area and all of those targeted area people have known about our system because this system is make their life easier. User can understand this system by own.

### **2.7.4 Accessibility Requirements**

The requirements for how easy it should be for people and our system is very much easy to use and understand

### **2.7.5 User Documentation Requirements**

Gather the user expects and must do it in system properly is mandatory because this requirement given by user directly. And after the complete the system user can give new requirements then it's also be added in system.

### **2.7.6 Training Requirements**

If it's need to training the user for use the system properly then it's must to training them properly. In our project we don't need this type of training for user but a simple guide for them.

## **2.8 Look and Feel Requirements**

If your systems look garbage then user can fell boring and not going to next step. Look and feel requirements are how the system will look like and how the user interface or graphical user interface of our system will display to the user.

## 2.8.1 Appearance Requirements

Apartment Owner and Admin have to know which field is require and which is not because we make it easy understandable. Such as, if there any optional field then the word “optional” appeared into the field.

<b>AR1</b>	In optional field the word “optional” must appear into the field		
<b>Description</b>	If there is no optional word appeared into the field that means this field is mandatory for input.		
<b>Stakeholders</b>	Apartment Owner, Admin	<b>Priority</b>	High

## 2.8.2 Style Requirements

Keeping all contents within a format is easy to understandable for user. A good style can attract to user keep them into the system long time.

<b>SR1</b>	All content must be appear within a format		
<b>Description</b>	Input field and other view result show an specific format		
<b>Stakeholders</b>	Admin	<b>Priority</b>	Medium

## 2.9 Operational and Environmental Requirements

Operational requirements are those statements that identify the essential capabilities, associated, and performance measures.

The operational environment stands for political, social, legislative, economic, cultural and natural environmental factors that significantly affect the implementation of any cooperation.

### 2.9.1 Expected Physical Requirements

There are no physical requirements in our system.

## **2.9.2 Requirements for Interfacing with Adjacent Systems**

There are no requirements for interfacing with adjacent system for our project.

## **2.9.3 Release Requirements**

There are no specific release requirements in our system.

## **2.10 Legal Requirements**

Legal Requirements means, as to any Person, any requirement under a Permit and any Governmental Rule. But our system there is no connection with Government.

### **2.10.1 Compliance Requirements**

There are no compliance requirements for our system.

### **2.10.2 Standards Requirements**

There are no standards requirements for our system.

# **Chapter 3**

## Requirement Analysis

### 3.1 Use Case Diagram

We have use case diagram. And there are two actors. Each actor plays different role. And those are already indicated to this use case diagram. This diagram will clarify our system in brief.

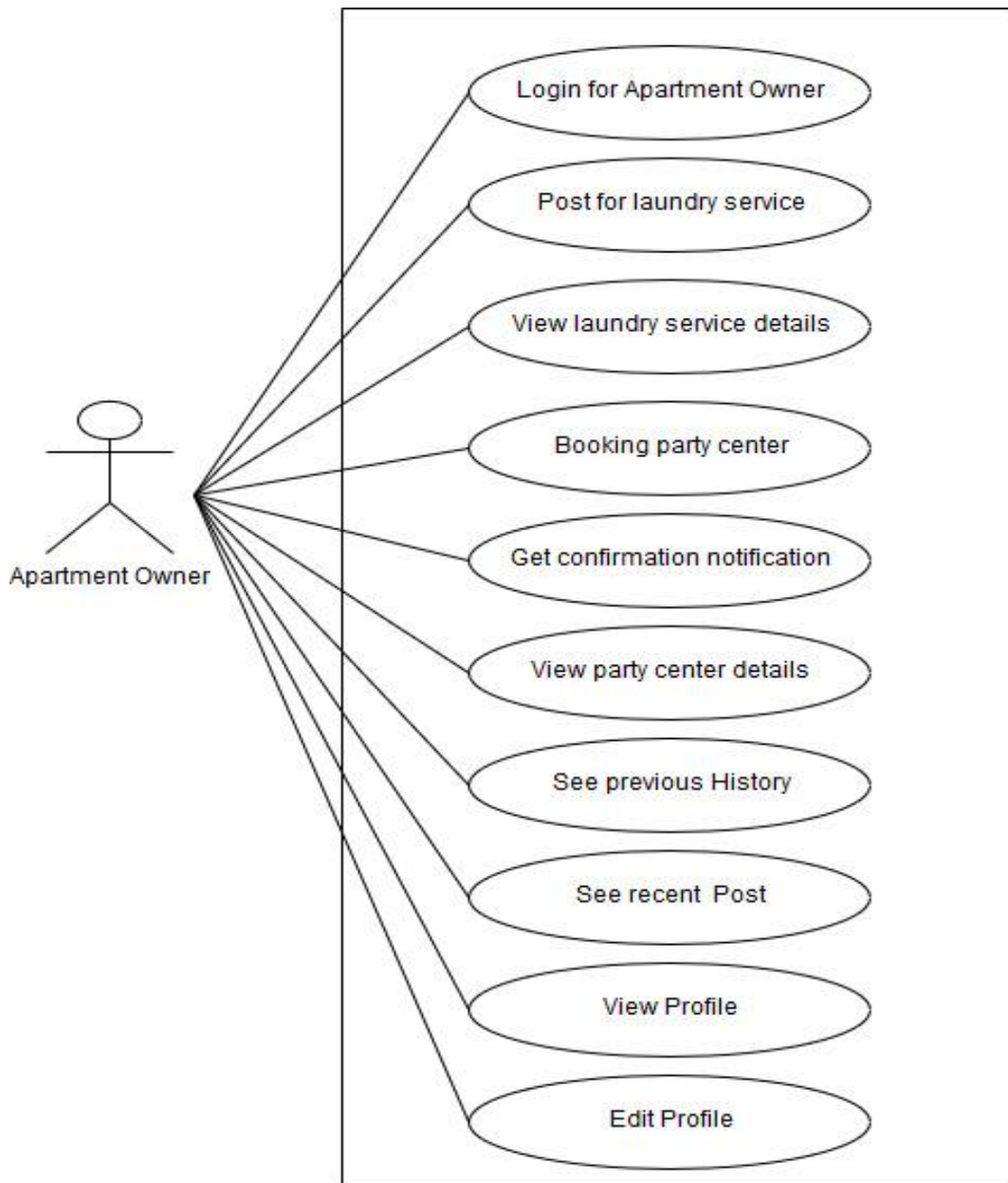


Figure 3.1: Use Case diagram for “Apartment Colony Service System”

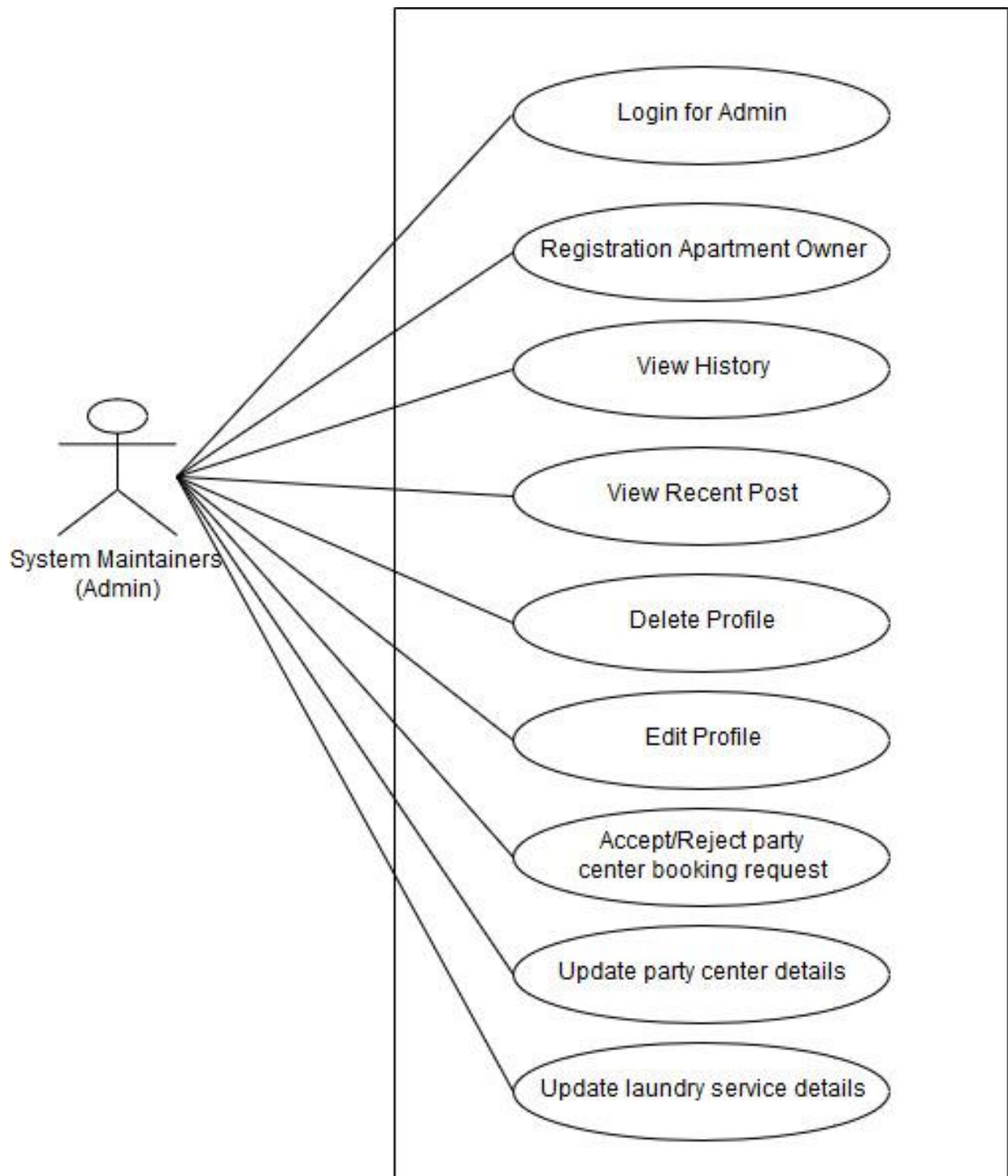


Figure 3.2: Use Case diagram for “Apartment Colony Service System”

## Use Case Description

### 3.1.1 Login for Apartment owner

<b>Use Case</b>	Login for Apartment owner	
<b>Goal</b>	Apartment Owner enter the application after the login successfully	
<b>Preconditions</b>	Apartment Owner must exist in database	
<b>Success End Condition</b>	Apartment Owner can access the system	
<b>Failed End Condition</b>	The user name and password is not valid	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	Apartment Owner click on the “login” button	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	System display the login page
	2	Apartment owner enter his Email and Password
	3	Apartment owner click on the login button
	4	Apartment owner access the system
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	3a	Email and Password in not valid
	4a	Can’t access the system
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.2 Apartment owner post for laundry service

<b>Use Case</b>	Apartment owner post for laundry service	
<b>Goal</b>	Apartment Owner post for Laundry Service	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Fill up all require information field</li> </ul>	
<b>Success End Condition</b>	Apartment owner successfully post for Laundry Service	
<b>Failed End Condition</b>	Apartment owner can’t post for Laundry Service	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	Post for Laundry Service data visible to Admin	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner provide all required information
	2	Submit the request post for Laundry Service
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment can’t post for Laundry Service
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A



### 3.1.3 Apartment owner view laundry service details

<b>Use Case</b>	Apartment owner view laundry service details	
<b>Goal</b>	Apartment Owner can see details information of laundry service	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Apartment Owner</li> <li>Apartment Owner must be logged in</li> </ul>	
<b>Success End Condition</b>	Apartment owner see laundry service details	
<b>Failed End Condition</b>	Apartment owner can't able see laundry service details	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	laundry service details data visible to the apartment owner	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner can be able to see laundry service details
	2	Laundry services data appear to apartment owner
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment Owner can't able to see his laundry service details
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.4 Apartment owner booking party center

<b>Use Case</b>	Apartment owner booking party center	
<b>Goal</b>	Apartment Owner send party center booking request	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Apartment Owner</li> <li>Apartment Owner must be logged in</li> <li>Fill up all require information field</li> </ul>	
<b>Success End Condition</b>	Apartment owner successfully send party center booking request	
<b>Failed End Condition</b>	Apartment owner can't send party center booking request	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	send party center booking request data visible to Admin	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner provide all required information
	2	Send party center booking request
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment Owner can't send party center booking request
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.5 Apartment owner view party center details

<b>Use Case</b>	Apartment owner view party center details	
<b>Goal</b>	Apartment Owner can see details information of party center	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> </ul>	
<b>Success End Condition</b>	Apartment owner see party center details	
<b>Failed End Condition</b>	Apartment owner can't able see party center details	
<b>Primary Actors: Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	party center information data visible to the apartment owner	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner can be able to see party center details information
	2	party center information data appear to apartment owner
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment Owner can't able to see party center info
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.6 Apartment owner sees previous history

<b>Use Case</b>	Apartment owner sees previous history	
<b>Goal</b>	Apartment Owner see his all previous history	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment owner have some previous activities</li> </ul>	
<b>Success End Condition</b>	Apartment owner see his previous history	
<b>Failed End Condition</b>	Apartment owner can't able see his previous history	
<b>Primary Actors: Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	Previous history data visible to the apartment owner	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner click on "Previous History" button to see his previous history
	2	Previous history data appear to apartment owner
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment Owner can't able to see his previous history
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	Apartment Owner must be have some previous activities

### 3.1.7 Apartment owner sees recent post

<b>Use Case</b>	Apartment owner sees recent post	
<b>Goal</b>	Apartment Owner see his recent post	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment owner have some recent post</li> </ul>	
<b>Success End Condition</b>	Recent post view on Apartment owner	
<b>Failed End Condition</b>	Apartment owner can't see recent post	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	Recent post visible to the apartment owner	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner click on "Recent Post" button to see his previous history
	2	Previous history data appear to apartment owner
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment can't able to see his previous history
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	Apartment Owner have a recent post

### 3.1.8 Apartment owner can view profile

<b>Use Case</b>	Apartment owner can view profile	
<b>Goal</b>	Apartment Owner can see his profile	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment owner have a profile</li> </ul>	
<b>Success End Condition</b>	Apartment owner can see his profile	
<b>Failed End Condition</b>	Apartment owner can't see his profile	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	Apartment owner profile appear	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner click on "Profile" button to see his profile
	2	Apartment owner profile appear to apartment owner
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment can't able to see his profile
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.9 Apartment owner can edit their profile

<b>Use Case</b>	Apartment owner can edit their profile	
<b>Goal</b>	Apartment Owner can edit his profile	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Apartment Owner</li> <li>Apartment Owner must be logged in</li> <li>Apartment Owner have a profile</li> </ul>	
<b>Success End Condition</b>	Apartment owner can edit his profile	
<b>Failed End Condition</b>	Apartment owner can't edit his profile	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner	
<b>Trigger</b>	Apartment owner profile edited	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner go to his profile
	2	Apartment owner edit his profile
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Apartment can't able to edit his profile
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.10 Get confirmation notification

<b>Use Case</b>	Apartment owner get party center booking confirmation message	
<b>Goal</b>	Apartment owner get confirmation message	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Apartment owner</li> <li>Apartment owner has a party center booking request</li> </ul>	
<b>Success End Condition</b>	Getting party center booking confirm message	
<b>Failed End Condition</b>	Not getting party center booking not confirmation message	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Apartment Owner Admin	
<b>Trigger</b>	Party center booking confirmation message send	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin Confirm the party center booking request
	2	Apartment owner getting party center booking confirmation message
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Party center booking confirmation message not sending
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.11 Login for Admin

<b>Use Case</b>	Login for Admin	
<b>Goal</b>	Admin enter the application after the login successfully	
<b>Preconditions</b>	Admin must exist in database	
<b>Success End Condition</b>	Admin can access the system	
<b>Failed End Condition</b>	The user name and password is not valid	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin	
<b>Trigger</b>	Admin click on the “login” button	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	System display the login page
	2	Admin enter his Email and Password
	3	Admin click on the login button
	4	Admin access the system
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	3a	Email and Password in not valid
	4a	Can’t access the system
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.12 Apartment owner registration

<b>Use Case</b>	Apartment owner registration	
<b>Goal</b>	Admin create apartment owner profile	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be logged in</li> </ul>	
<b>Success End Condition</b>	Admin create apartment owner profile successfully	
<b>Failed End Condition</b>	Admin can’t create apartment owner profile	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin Apartment Owner	
<b>Trigger</b>	An apartment owner profile created	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Fill up all require information
	2	Create apartment owner profile
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Admin can’t create apartment owner profile
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.13 Party center booking confirmation

<b>Use Case</b>	Party center booking confirmation	
<b>Goal</b>	Admin confirm the booking of party center	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Have a party center booking request by apartment owner</li> </ul>	
<b>Success End Condition</b>	Party center booking confirm	
<b>Failed End Condition</b>	Party center booking not confirm	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin Apartment Owner	
<b>Trigger</b>	Party center booking confirm	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin Confirm the party center booking request
	2	Party center booking confirm
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Party center booking not confirm
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.14 Admin view history

<b>Use Case</b>	Admin view history	
<b>Goal</b>	Apartment Owner see all previous history	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Apartment Owner must be logged in</li> </ul>	
<b>Success End Condition</b>	Admin can view previous history	
<b>Failed End Condition</b>	Admin can't able to view previous history	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin Apartment Owner, Maid-Servant	
<b>Trigger</b>	Previous history data visible to Admin	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Apartment owner click on "History" button to see previous history
	2	Previous history data appear to Admin
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Admin can't able to see previous history
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.15 Admin view recent post

<b>Use Case</b>	Admin view recent post	
<b>Goal</b>	Admin can see recent post	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Admin must be logged in</li> <li>Apartment owner have some recent post</li> </ul>	
<b>Success End Condition</b>	Recent post appear to Admin	
<b>Failed End Condition</b>	Admin can't see recent post	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin Apartment Owner	
<b>Trigger</b>	Recent post is visible to Admin	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin click on "Recent Post" button to see recent history
	2	Recent post appear to apartment owner
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Admin can't able to see recent post
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	Apartment Owner have a recent post

### 3.1.16 Admin can delete profile

<b>Use Case</b>	Admin can delete profile	
<b>Goal</b>	Admin can delete the all type of profile	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Admin must be logged in</li> </ul>	
<b>Success End Condition</b>	The profile is deleted	
<b>Failed End Condition</b>	The profile is not deleted	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin Apartment Owner, Maid-Servant	
<b>Trigger</b>	A profile is deleted	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin can delete apartment owner profile
	2	Apartment Owner profile is deleted
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	The profile is not deleted
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	There have some profile

### 3.1.17 Admin update's party center details

<b>Use Case</b>	Admin update's party center details	
<b>Goal</b>	Party center details updated	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Have a party center details chart</li> </ul>	
<b>Success End Condition</b>	Admin update party center details successfully	
<b>Failed End Condition</b>	Party center details not updated	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin	
<b>Trigger</b>	Party center details is updated	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin update the existing party center details
	2	Party center details updated
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Party center details not updated
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A

### 3.1.18 Admin update's laundry service details

<b>Use Case</b>	Admin update's laundry service details	
<b>Goal</b>	Laundry service details updated	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Have a laundry service details</li> </ul>	
<b>Success End Condition</b>	Admin update laundry service details successfully	
<b>Failed End Condition</b>	Laundry service details not updated	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin	
<b>Trigger</b>	Laundry service details is updated	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin update the existing party center details
	2	Party center details updated
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	Laundry service details not updated
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	N/A



### 3.1.19 Admin Can Edit Profile

<b>Use Case</b>	Admin Can Edit Profile	
<b>Goal</b>	Admin can edit the all type of profile	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be logged in</li> </ul>	
<b>Success End Condition</b>	The profile is edit	
<b>Failed End Condition</b>	The profile is not edit	
<b>Primary Actors:</b> <b>Secondary Actors:</b>	Admin Apartment Owner, Maid-Servant	
<b>Trigger</b>	profile is edited	
<b>Description / Main Success Scenario</b>	<b>Step</b>	<b>Action</b>
	1	Admin can edit apartment owner and maid-servant profile
	2	Apartment Owner or Maid-Servant profile is edit
<b>Alternative Flows</b>	<b>Step</b>	<b>Branching Action</b>
	2a	The profile is not edit
<b>Quality Requirements</b>	<b>Step</b>	<b>Requirements</b>
	1	There have some profile

## 3.2 Activity Diagram

We have prepared some activity diagram according to our use case. These activity diagrams are properly referring the flow of the individual conditions of our project.

### 3.2.1 Login for Apartment Owner

Apartment Owner login into the application through Email & Password that are provided by admin.

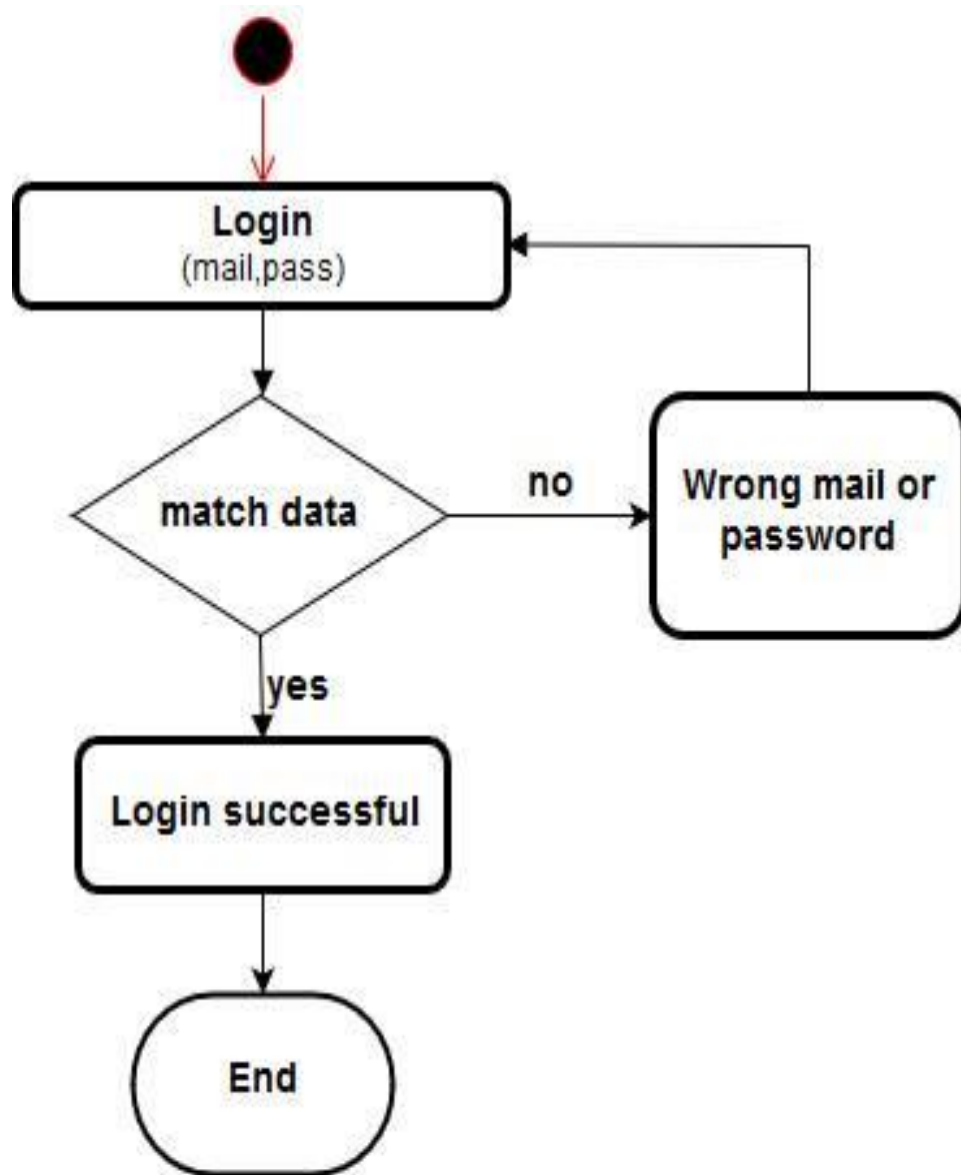


Figure 3.3: Login for apartment owner

### 3.2.2 Apartment Owner Post for Laundry Service

Apartment owner can post for laundry service when they have need. For post the laundry service they have to provide some information.

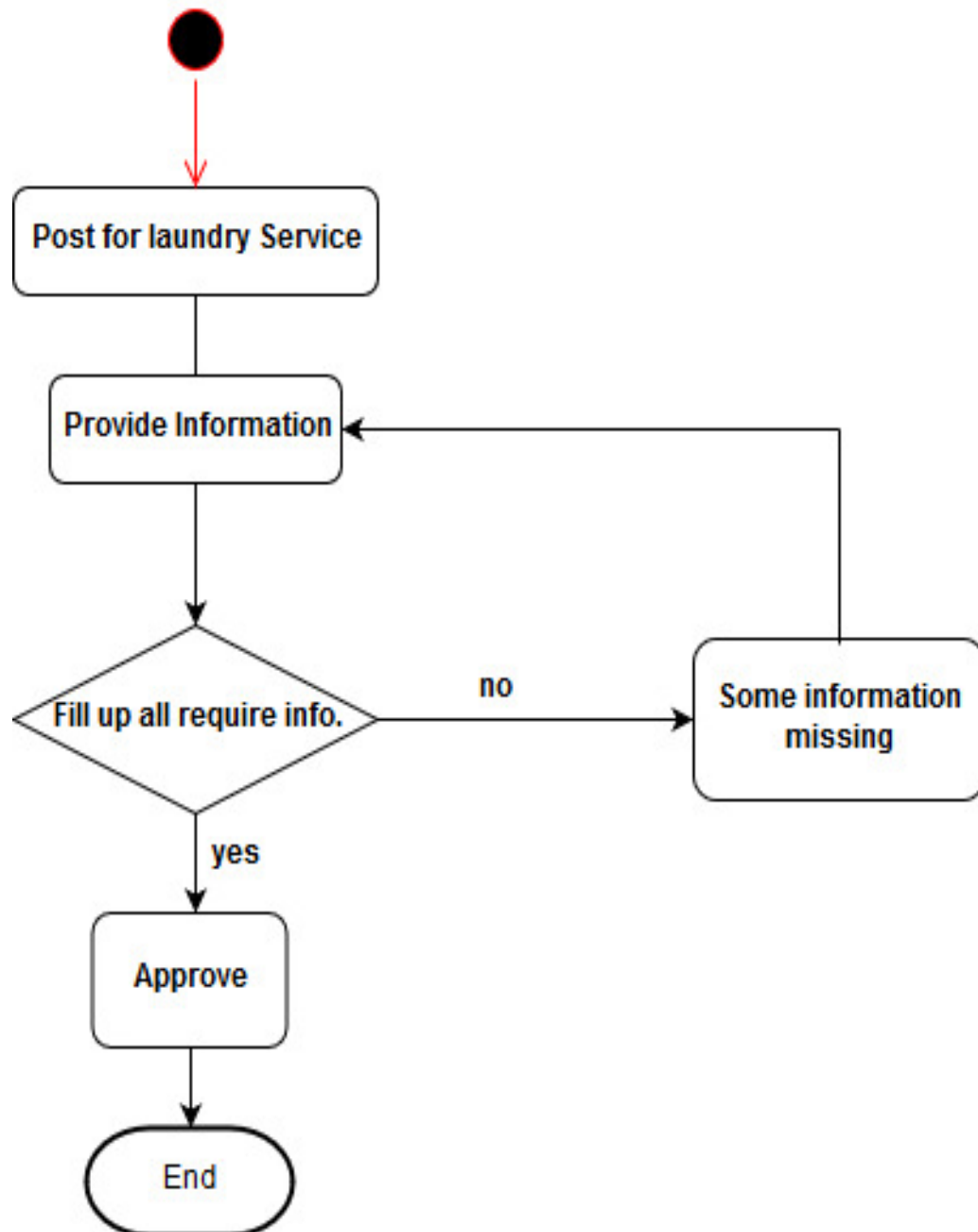


Figure 3.4: Apartment owner post for laundry service

### 3.2.3 Apartment Owner view laundry service details

Apartment owner can view laundry service details. There is a laundry service details chart and apartment owner can view it.

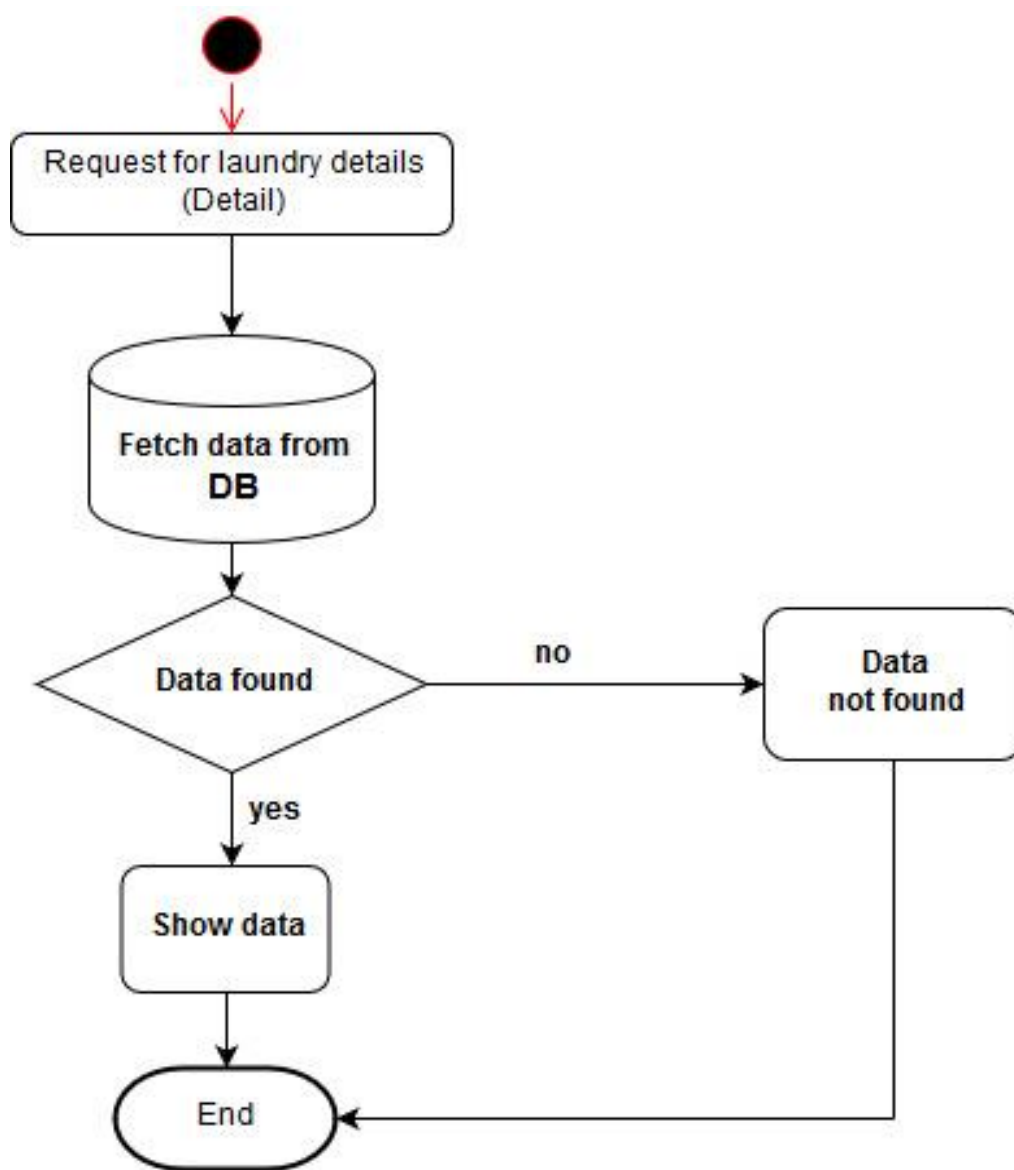


Figure 3.5: Apartment owner view laundry service details

### 3.2.4 Apartment Owner post for booking party center

Apartment owner can post for party center booking. For post the party center they have to provide some information.

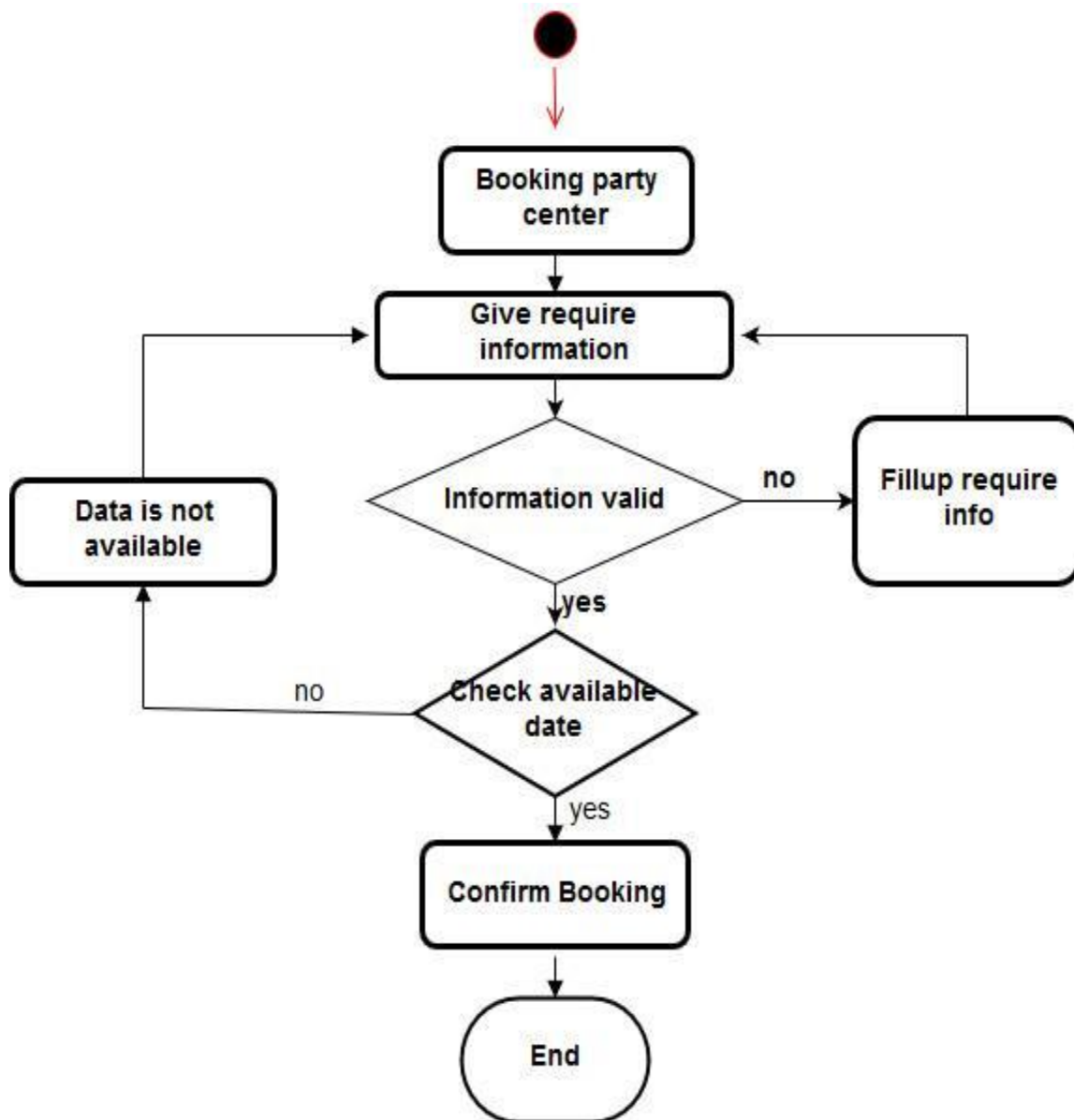


Figure 3.6: Apartment owner post for party center booking

### 3.2.5 Apartment Owner view party center details

Apartment owner can view party center booking details. There is a party center booking details chart and apartment owner can view it.

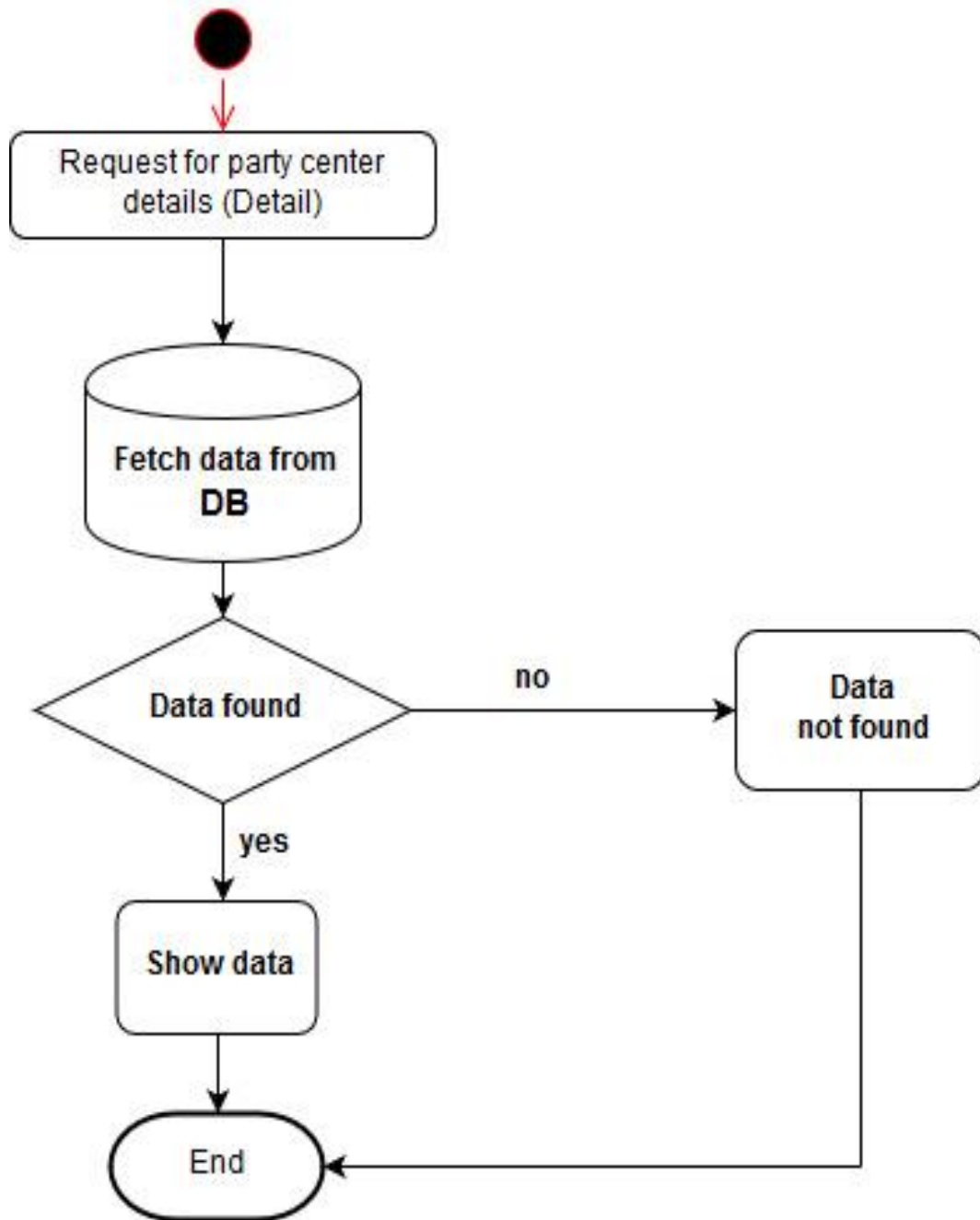


Figure 3.7: Apartment owner view party center booking details

### 3.2.6 Apartment Owner can view profile

Apartment owner can able to see their own profile. When they feel they need to update their profile then they can see their profile

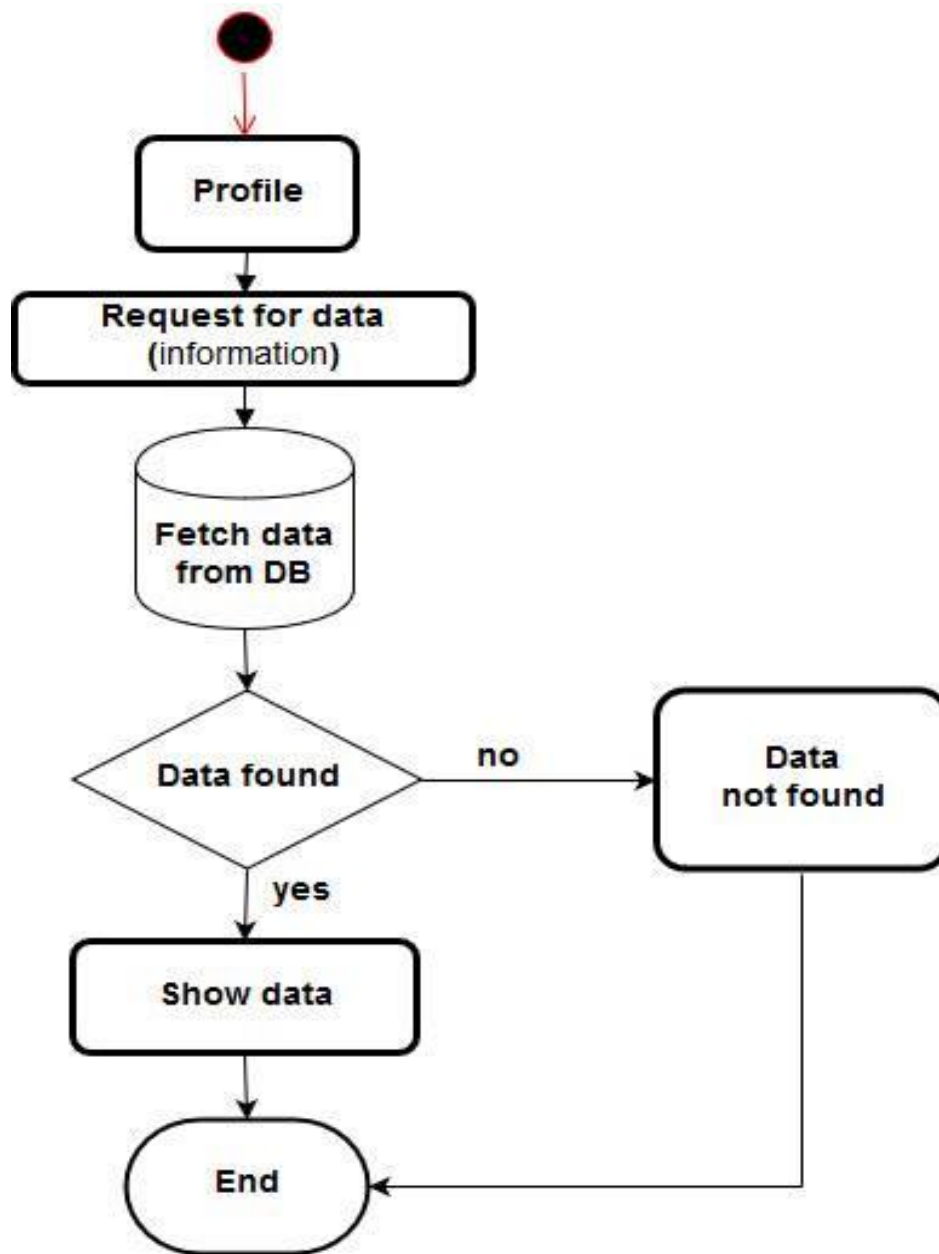


Figure 3.8: Apartment Owner can view profile

### 3.2.7 Apartment Owner sees previous history

When apartment owner doing some activities such as post for maid-servant, booking party center, etc. Then this activities will be recorded and apartment owner can see that activities in previous history part.

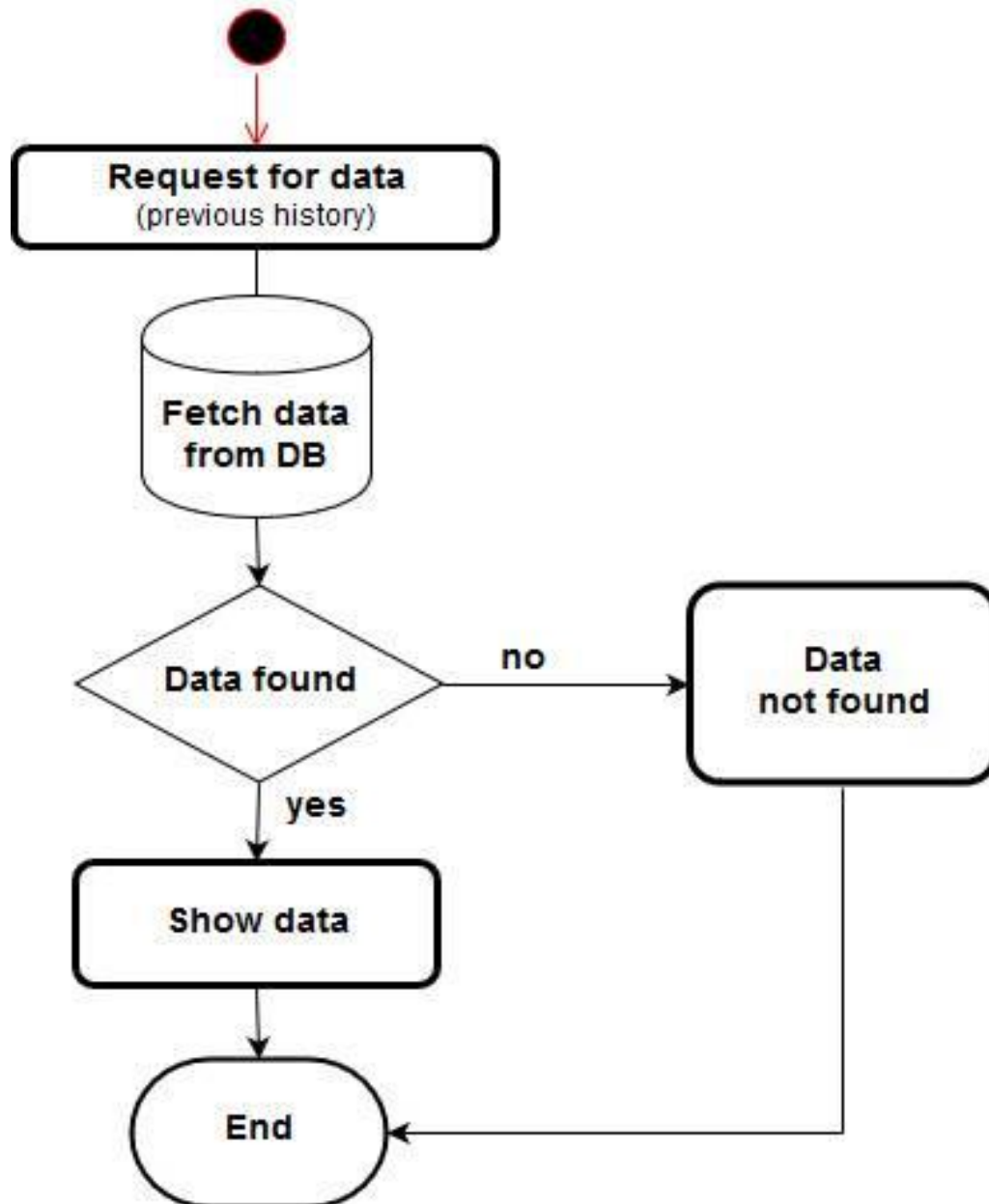


Figure 3.9: Apartment Owner sees previous history



### 3.2.8 Apartment Owner sees recent post

When apartment owner perform any post and no action has be taken against that post yet then apartment owner can see that post in recent post area.

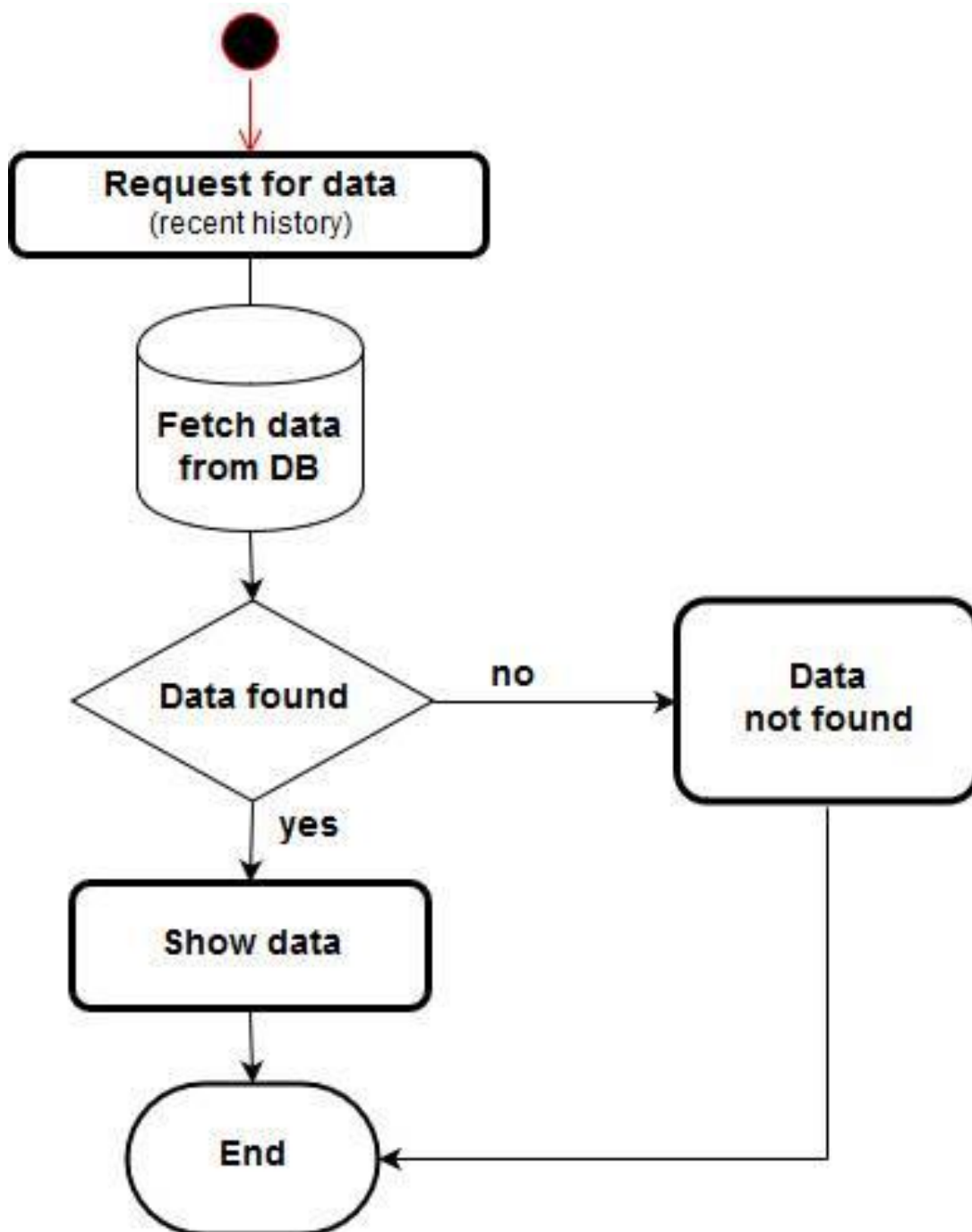


Figure 3.10: Apartment owner see recent post

### 3.2.9 Apartment Owner can edit his profile

When apartment owner want to that, they have to update their profile then they can easily edit their own profile

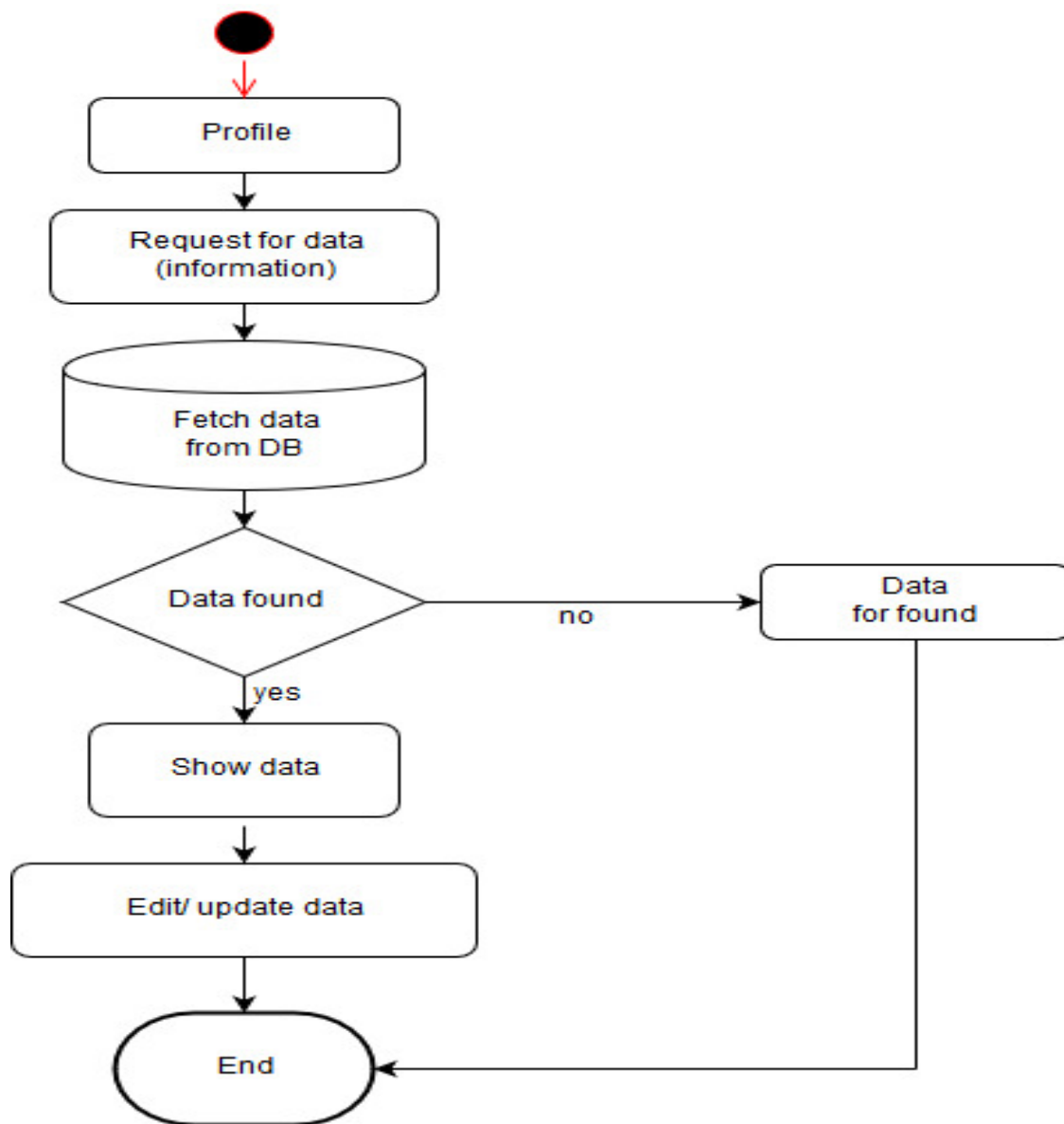


Figure 3.11: Edit apartment owner profile

### 3.2.10 Login for Admin

Admin also needs to verify them by login into the system, using email and password. Admin can added more admin.

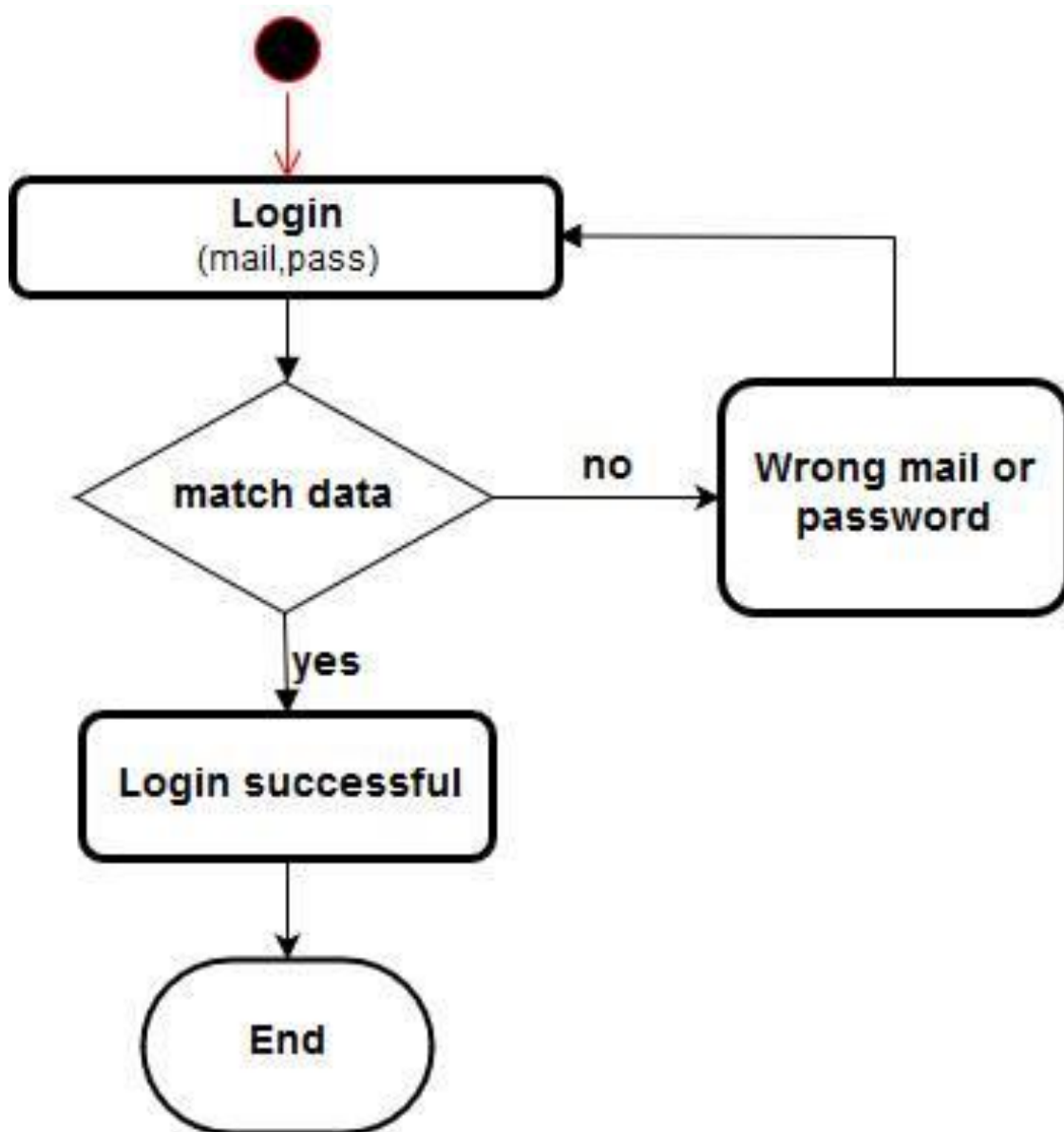


Figure 3.12: Admin login

### 3.2.11 Admin create apartment owner registration

Admin created apartment owner profile and provide the profile to apartment owner, and using this profile apartment owner entry the system.

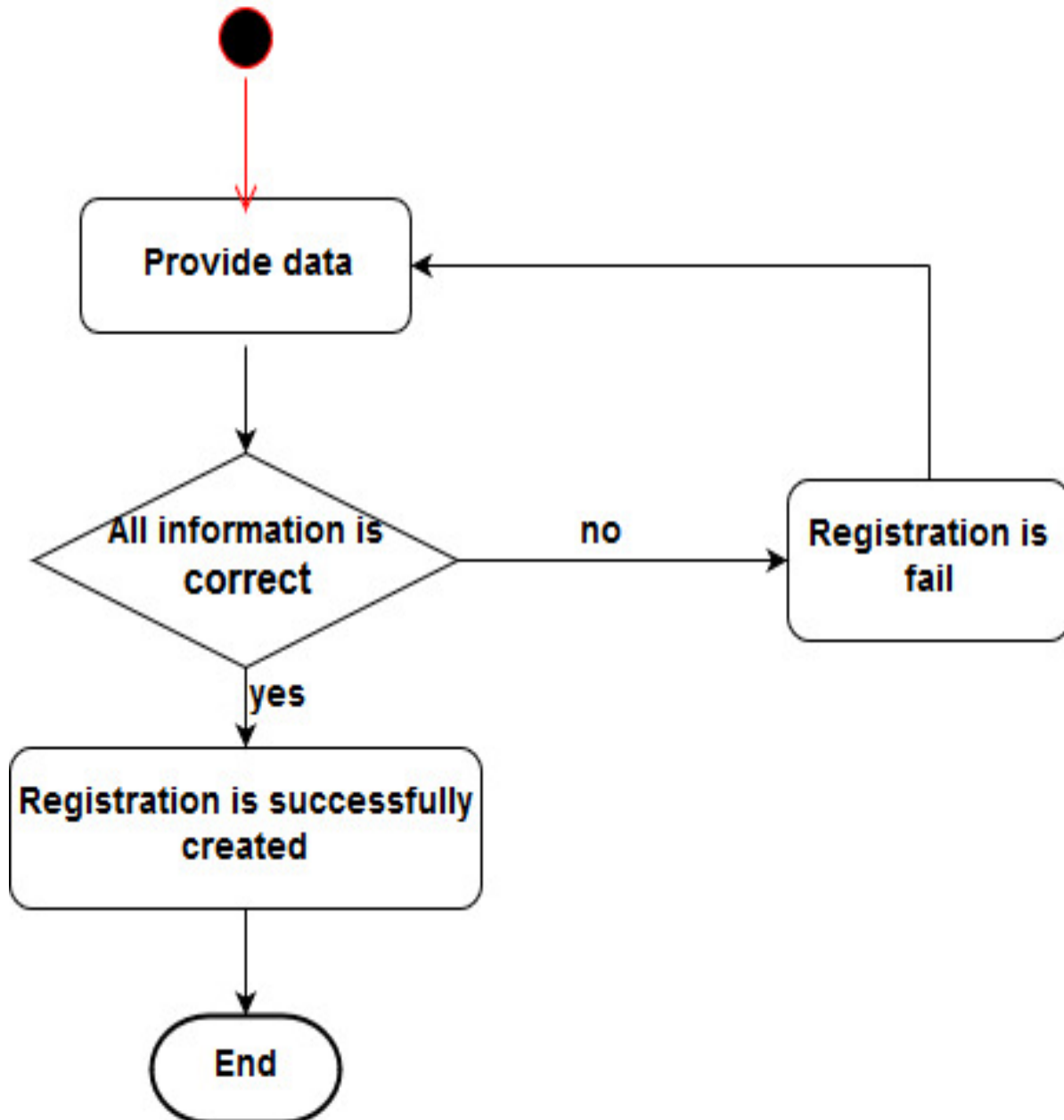


Figure 3.13: Create apartment owner registration

### 3.2.12 Admin sees previous history

After a certain time the all activities of apartment owner goes to the previous history. Then this activities will be recorded admin can see that activities in previous history part

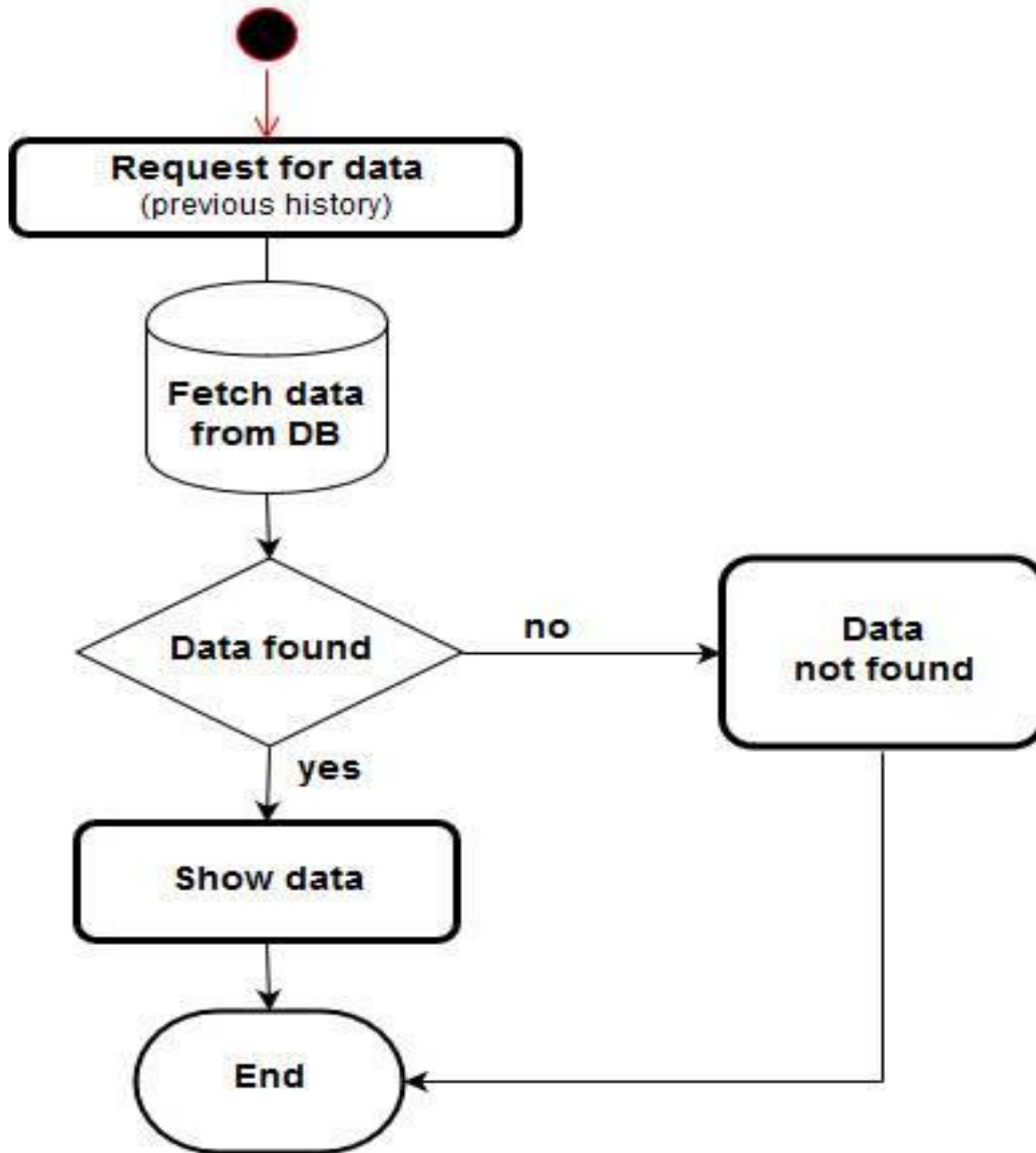


Figure 3.14: Admin sees previous history

### 3.2.13 Admin sees recent post

When apartment owner perform any post and admin don't takes any action against that post yet then this post appear on recent post and admin can this recent post.

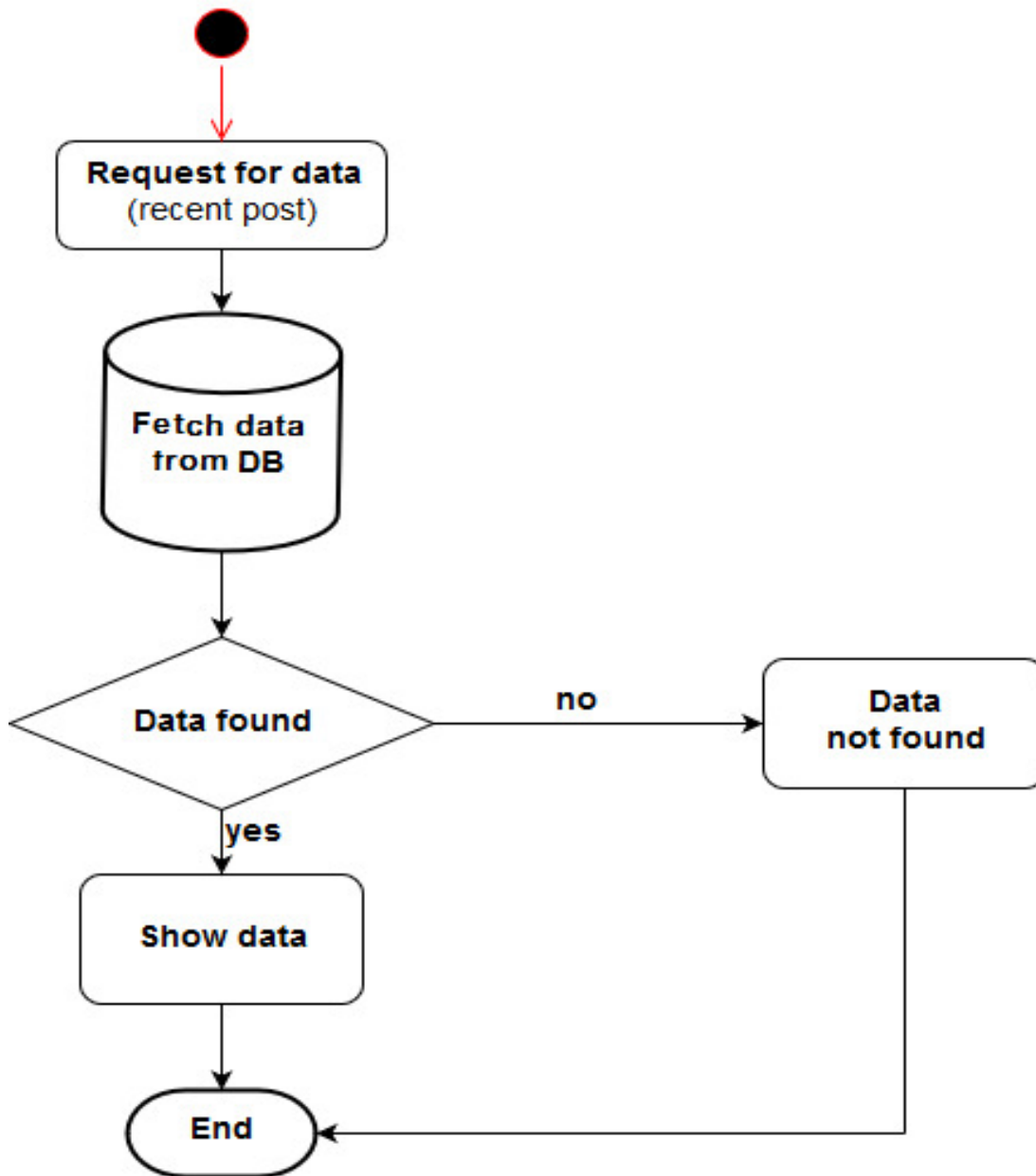


Figure 3.15: Admin sees recent posts

### 3.2.14 Admin can edit and delete apartment owner profile

Admin can edit apartment owner and Maid-Servant Profile. When it will be need to edit an apartment owner or maid-servant profile then admin can edit it.

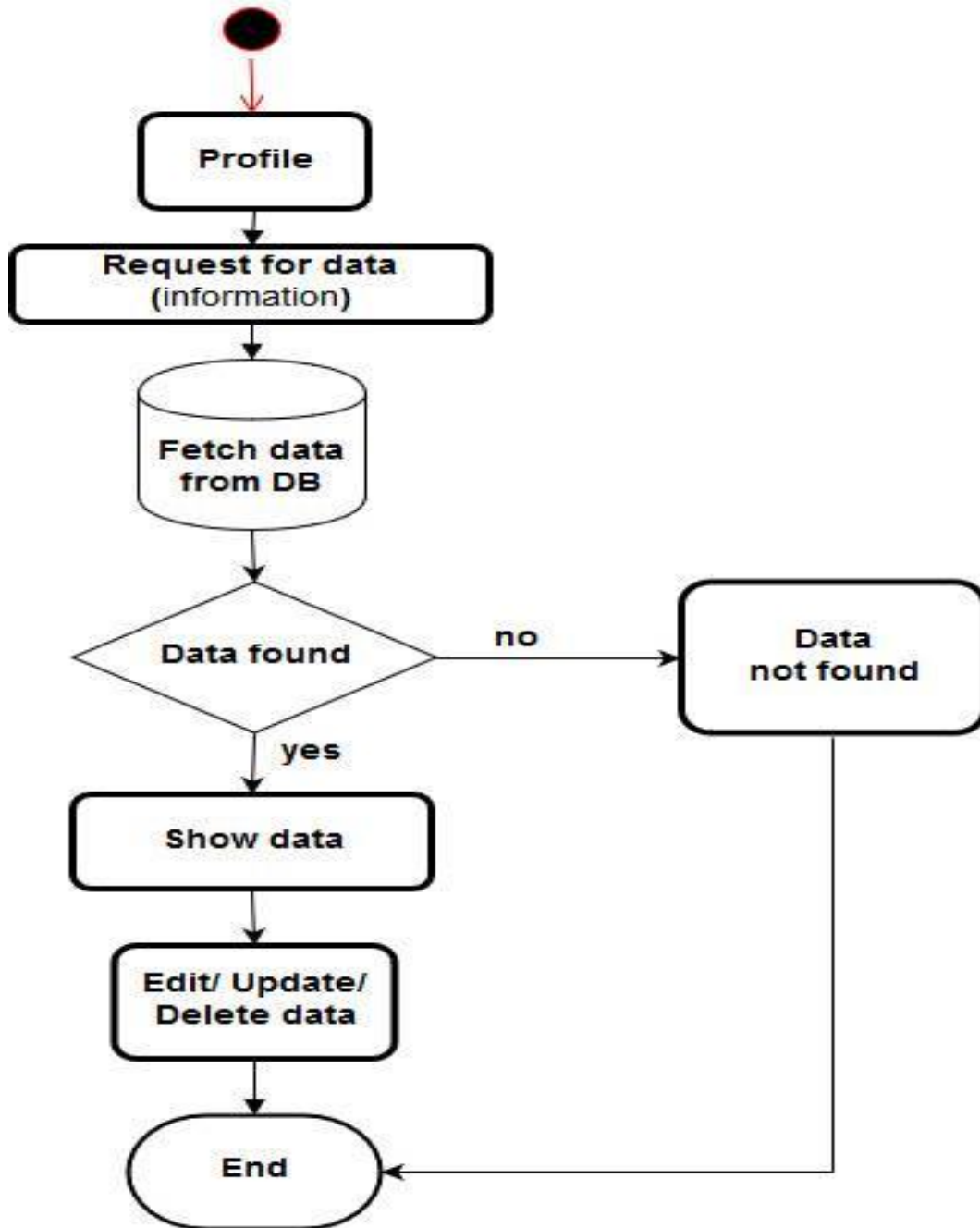


Figure 3.16: Admin Can Edit/ Delete apartment owner profile

### 3.2.16 Party center booking confirmation

After party center booking request come to the admin, admin check the date if the day is available then admin confirm the booking

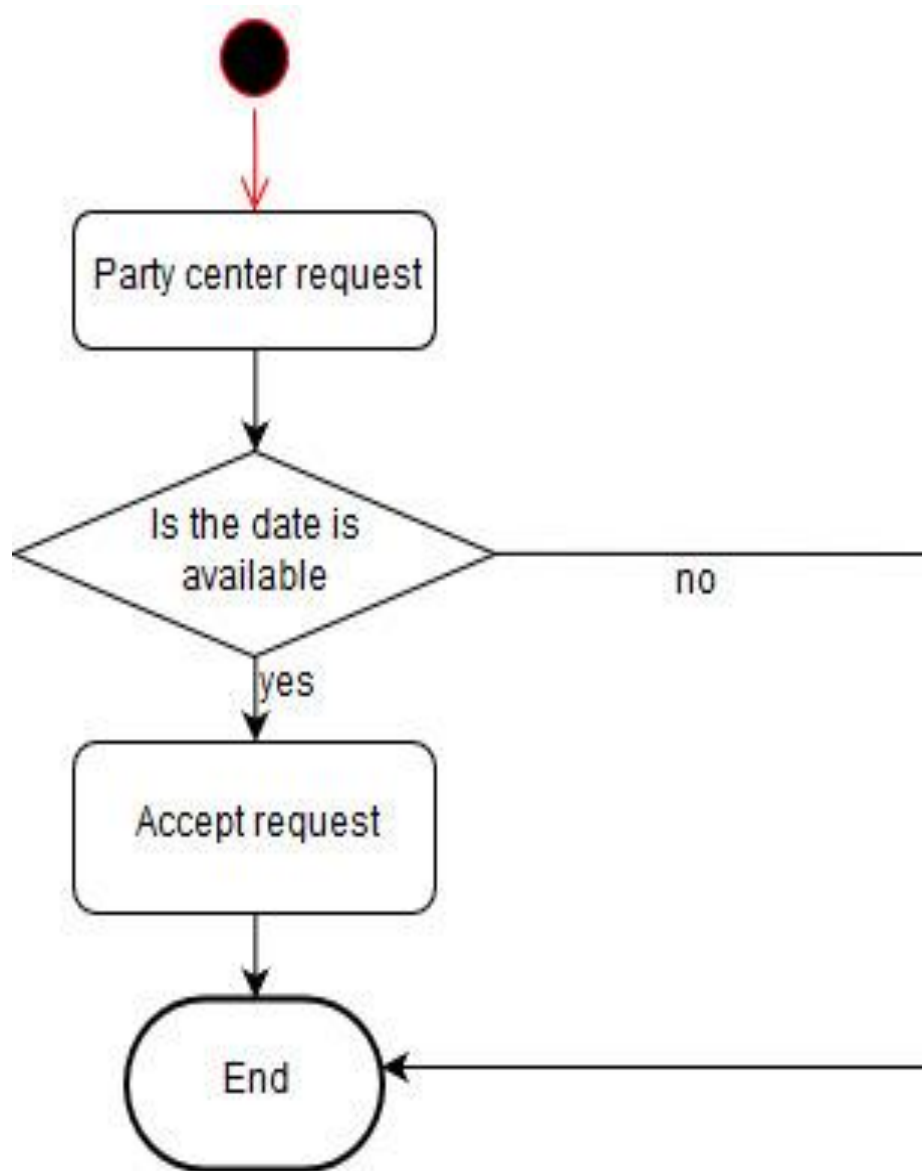


Figure 3.17: Party center booking confirmation



### 3.2.17 Admin can update party center details

Admin can update the party center details information

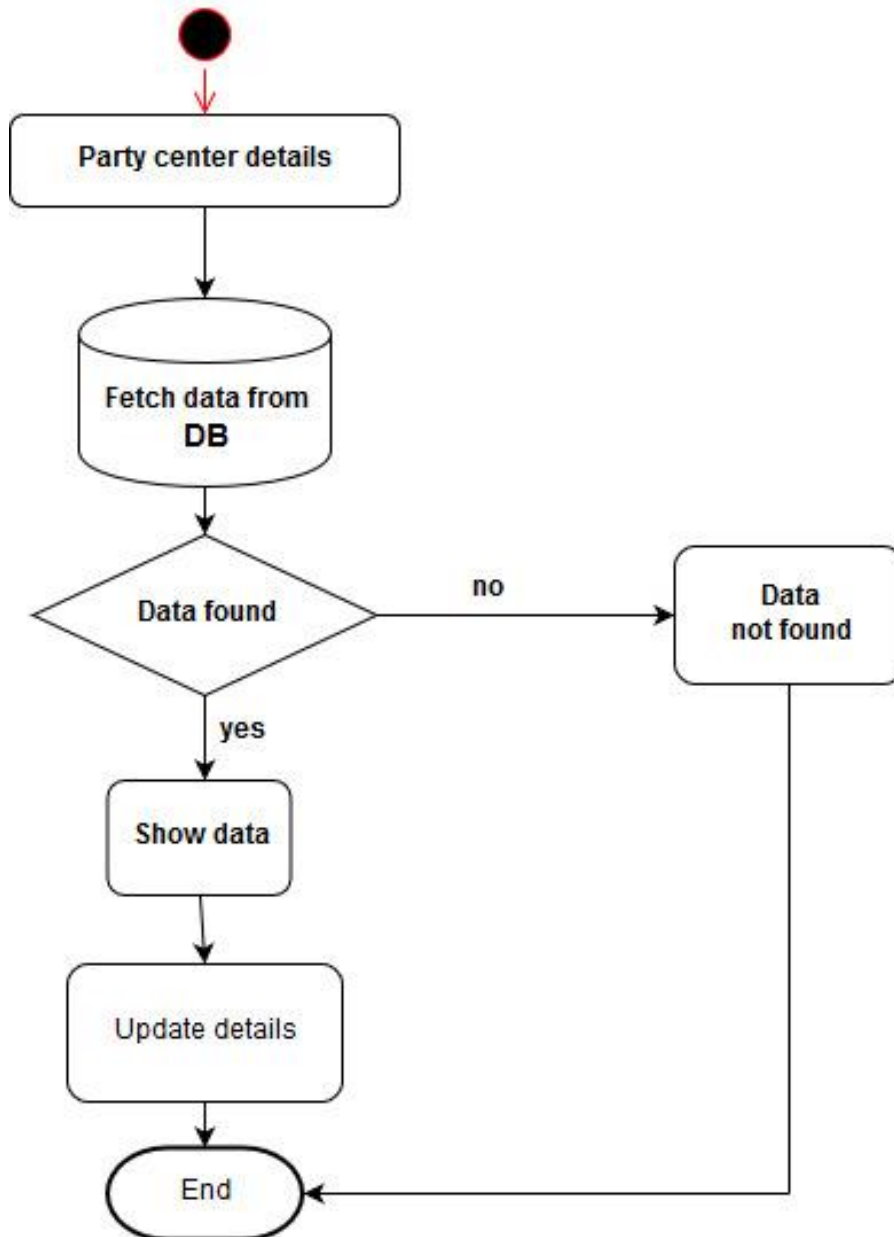


Figure 3.18: Admin can update party center details

### 3.3 Sequence Diagrams

Mainly sequence diagrams understand us how the data will be followed in any application. Now we are going to show some sequence diagrams.

#### 3.3.1 Login for Apartment Owner

Apartment owner login the system using the mail and password that provided by admin.

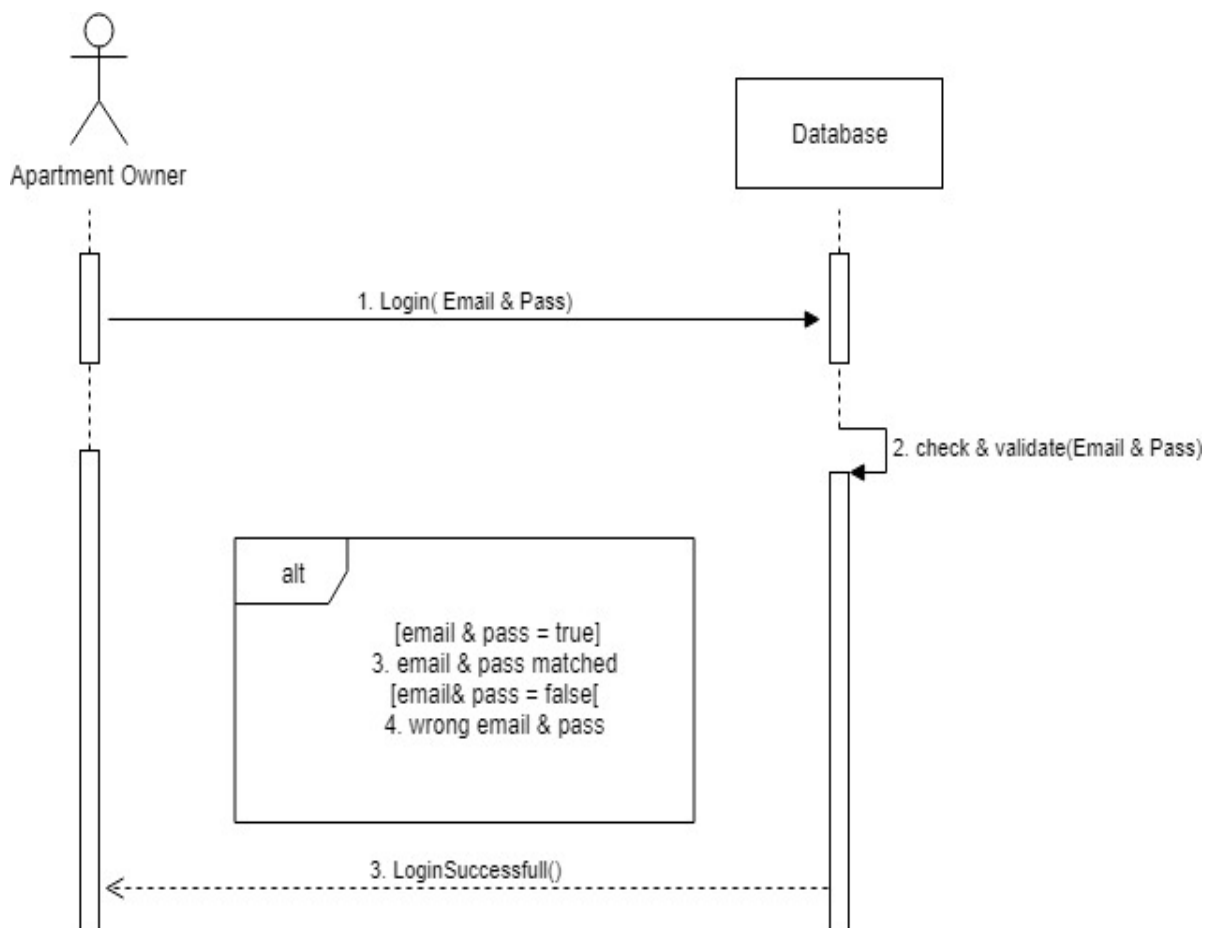


Figure 3.19: Sequence Diagram for Apartment owner Login

### 3.3.2 Apartment Owner can view their profile

Apartment owner can able to view their own profile.

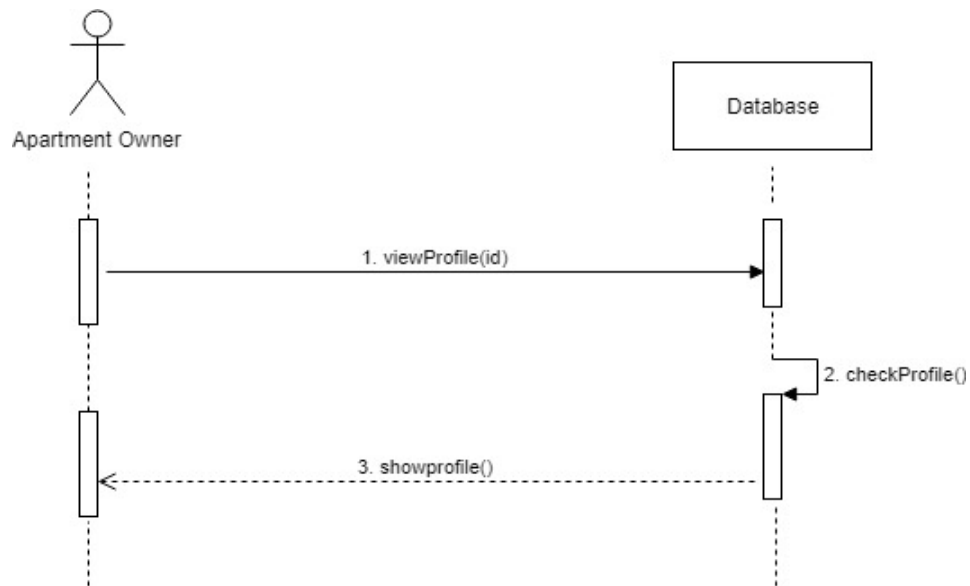


Figure 3.20: Sequence Diagram for Apartment owner view his/her Profile

### 3.3.3 Apartment Owner sees his previous post history

Apartment owner can see their previous, if there have some previous data.

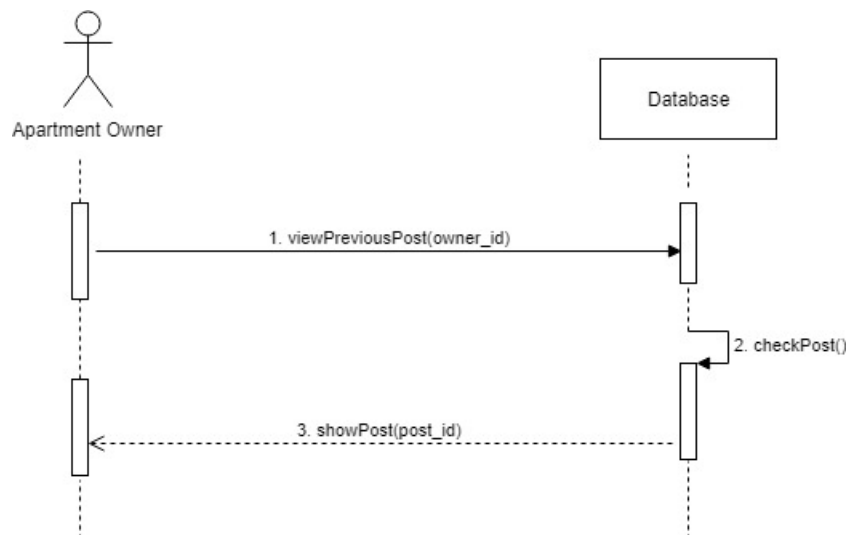


Figure 3.21: Sequence Diagram for Apartment owner sees his/her post History

### 3.3.4 Apartment Owner sees their recent post

Apartment owner can also view their recent post, if they perform any post before.

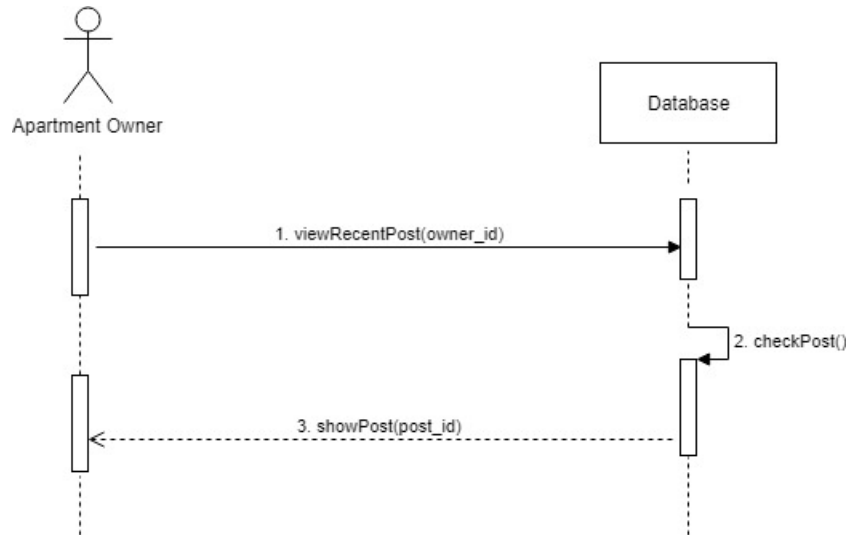


Figure 3.22: Sequence Diagram for Apartment owner sees his/her recent post

### 3.3.5 Apartment Owner edits their profile

When it will be needed to edit the profile then apartment owner can edit their own profile.

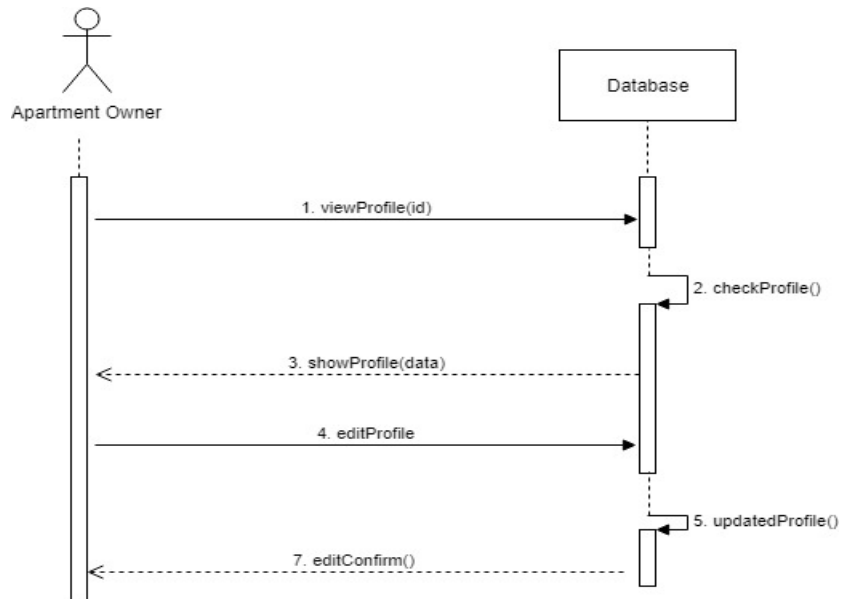


Figure 3.23: Sequence Diagram for Apartment owner edits their profile

### 3.3.6 Apartment Owner post for laundry service

Apartment owner enter into the system and can post for laundry service.

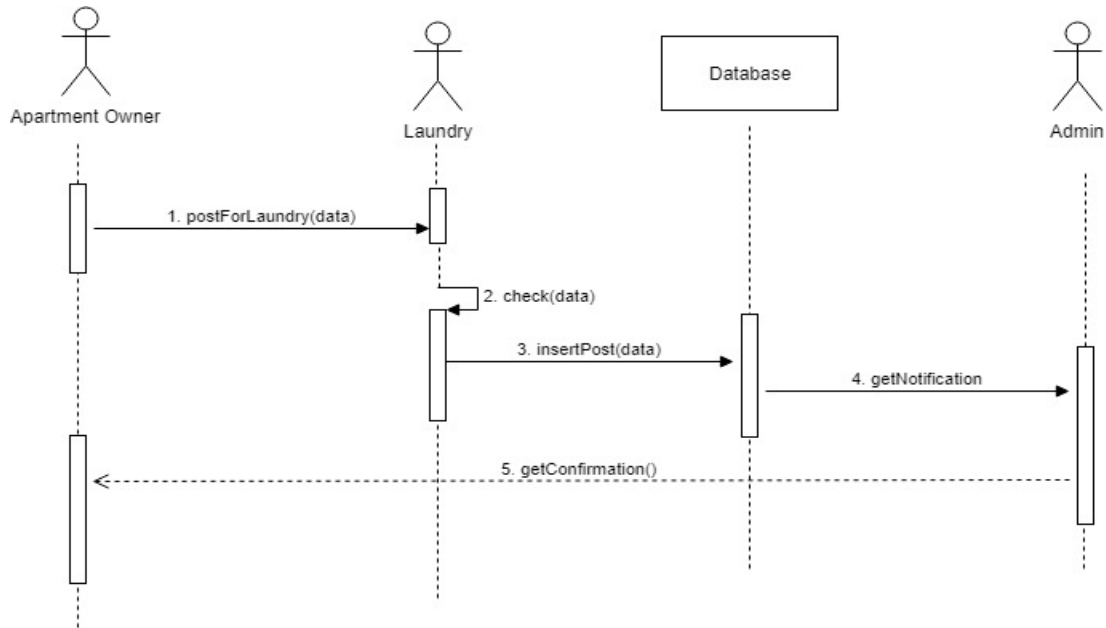


Figure 3.24: Sequence Diagram for Apartment owner post for laundry service

### 3.3.7 Apartment Owner post for booking party center

Apartment owner enters into the system and can post for booking party center

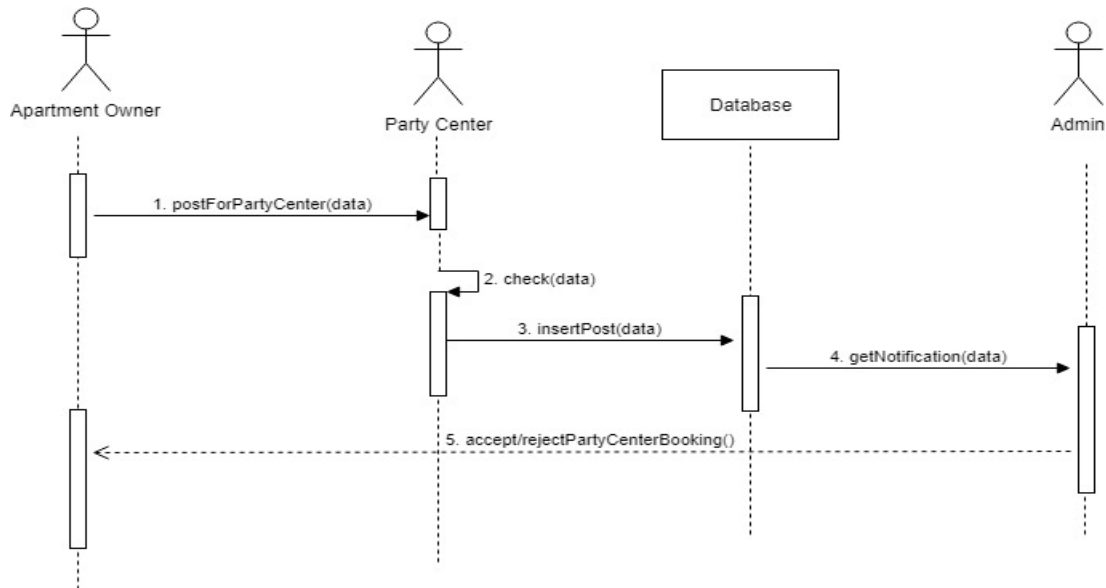


Figure 3.25: Sequence Diagram for Apartment owner post for party center booking

### 3.3.8 See Laundry service details

Apartment owner can view laundry service details.

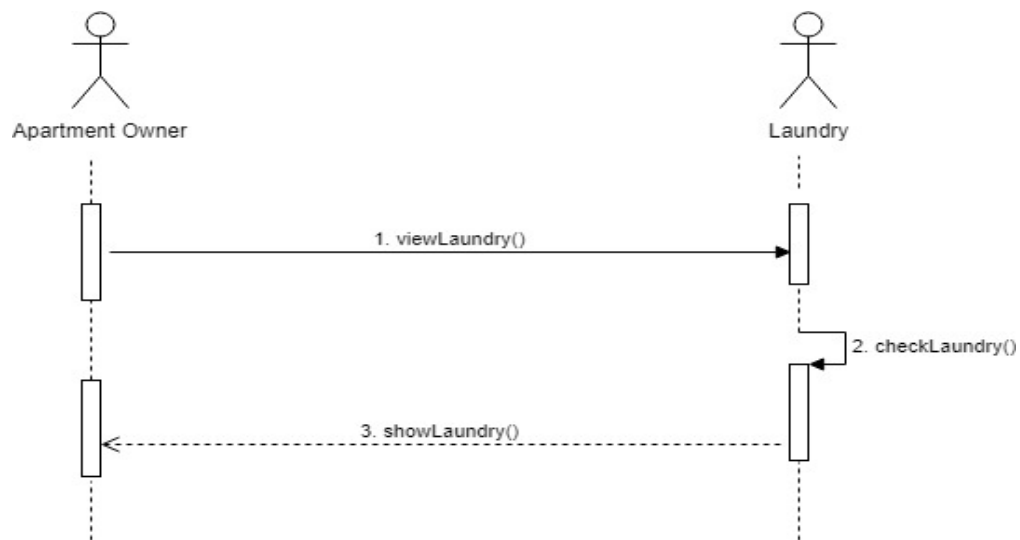


Figure 3.26: Sequence Diagram for Apartment owner see laundry service details

### 3.3.9 See Party Center details

Apartment owner can view party center booking details.

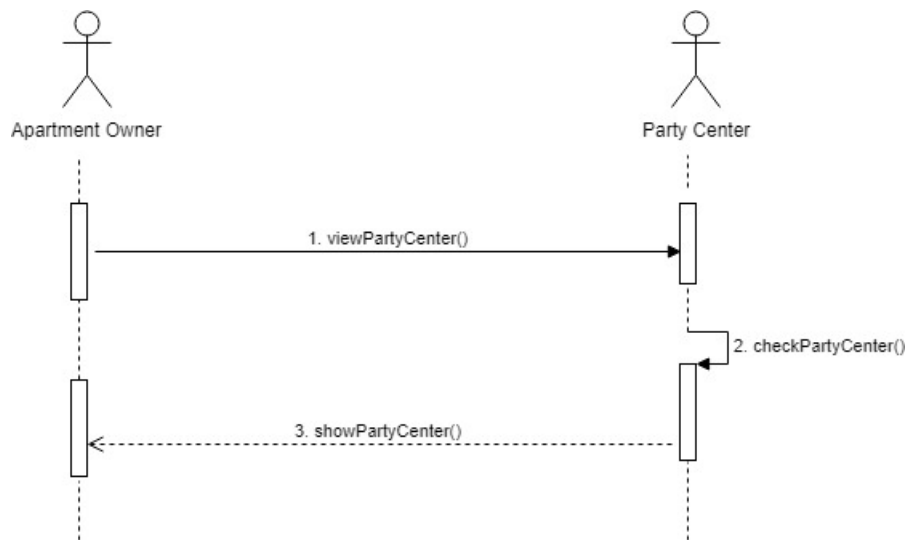


Figure 3.27: Sequence Diagram for Apartment owner see laundry service details

### 3.3.10 Login for Admin

Admin also needs to authenticate by login into the system, using email and password.

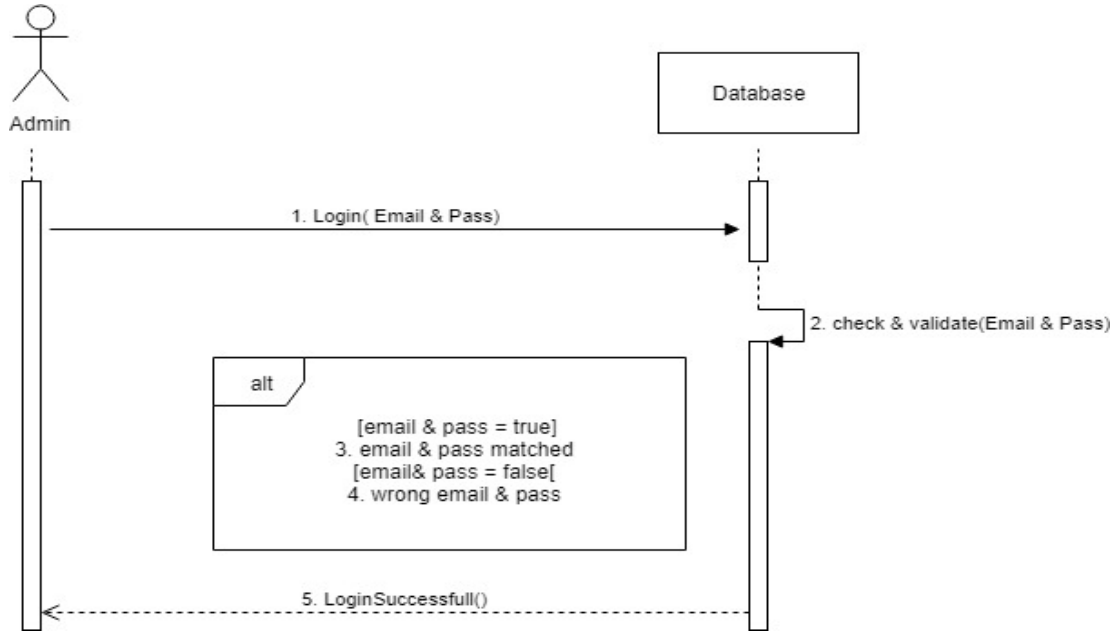


Figure 3.28: Sequence Diagram for Admin Login

### 3.3.11 Admin registration for Apartment Owner

Admin created apartment owner profile and provide the profile to apartment owner, and using this profile apartment owner entry the system.

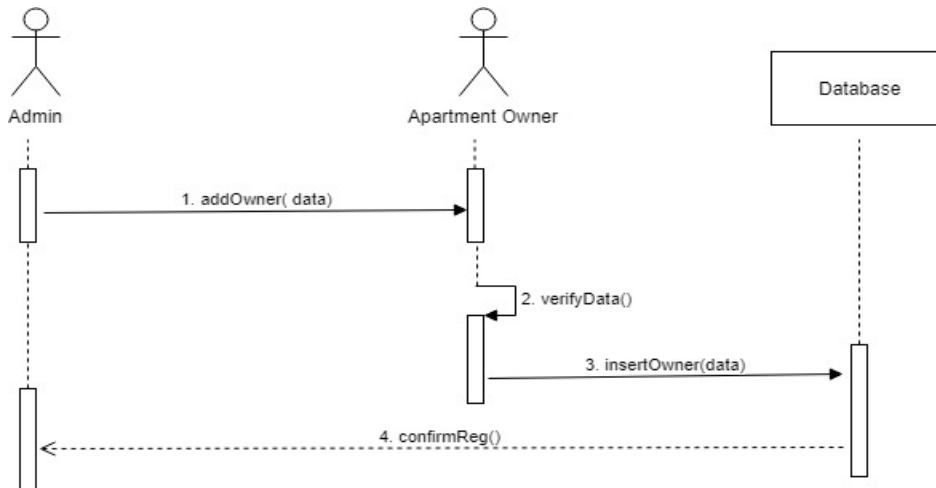


Figure 3.29: Sequence Diagram, registration for Apartment Owner

### 3.3.12 Admin view Apartment Owner's post history

The all previous history data of apartment owner is visible to admin.

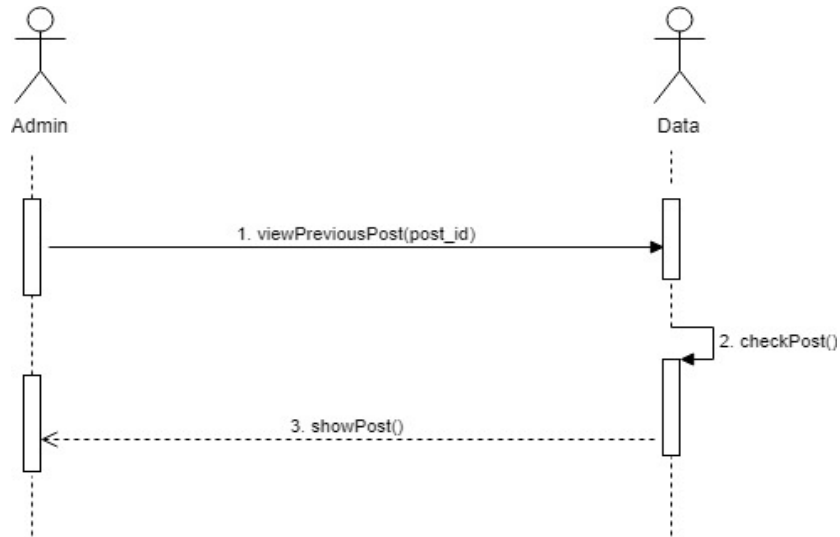


Figure 3.30: Sequence Diagram, Admin view Apartment Owner's post history

### 3.3.13 Admin view Apartment Owner's recent post

The recent post of apartment owner is also visible to admin

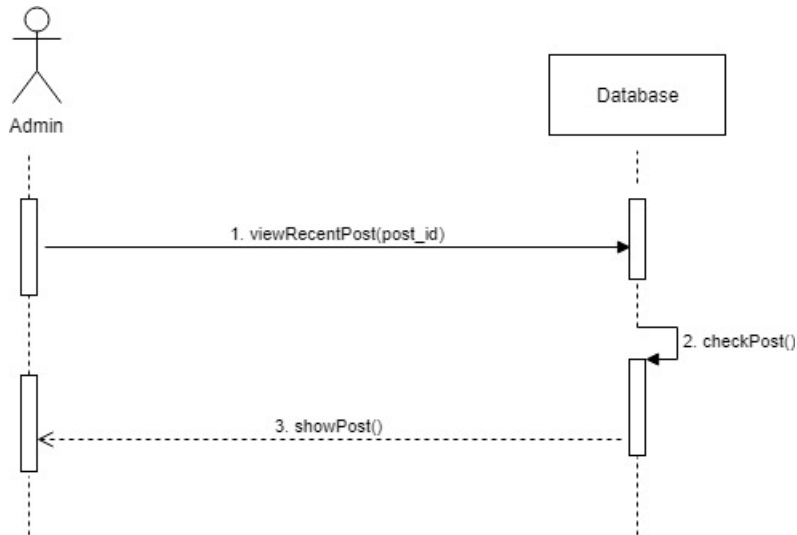


Figure 3.31: Sequence Diagram, Admin view Apartment Owner's recent post



### 3.3.14 Admin edits Apartment Owner/maid-servant profile

Admin can edit the apartment owner profile.

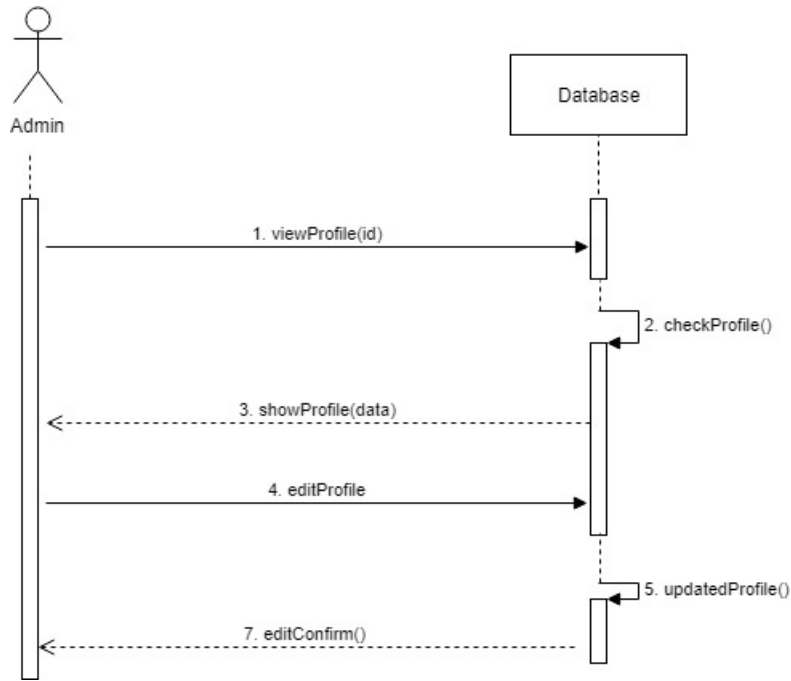


Figure 3.32: Sequence Diagram, Admin edits Apartment Owner profile

### 3.3.15 Admin deletes Apartment Owner/maid-servant profile

Admin also can delete the apartment owner and maid-servant profile

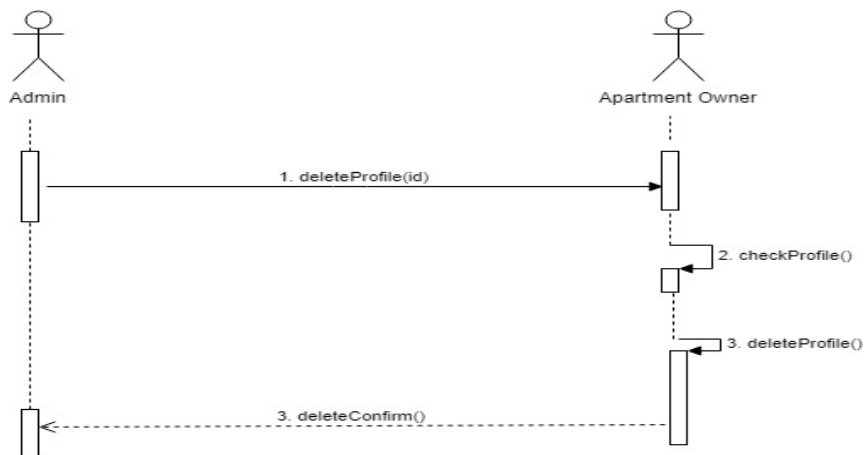


Figure 3.33: Sequence Diagram, Admin deletes Apartment Owner profile

# **Chapter 4**

## System Design Specification

## 4.1 Development tools and technology

To develop software some tools are needed to be used. And there are too many tools for develop software. For software developer there are so many tools and technique to make the development phase easy. In our project the below tools we are going to use.

### 4.1.1 User Interface Technology

A good user interface can help to keep the user into the system. The main point of UI in a system is how much easy is to use and how simple it is. UI linked the user and system. UI refer to the structure of content, Action Button, Images etc.

#### 4.1.1.1 CSS

There are many technologies to displaying the HTML elements. Html elements can appear in many ways. Is goods practice to use external CSS. Internal and Inline CSS not for a project and it also not a good practice.

#### 4.1.1.2 Bootstrap

In our project we used bootstrap for front-end design. Bootstrap is a free and open-source front-end Web framework. This framework is developed by CSS and JavaScript. It's browser friendly framework. For using bootstrap we also able to use some JavaScript facilities.

#### 4.1.1.3 JQuery

jQuery is a JavaScript library. It is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating. In our project we use a very few JQuery function.

#### 4.1.1.4 Programming Language

Programming language is the only way to make any system. Any system can be developed with any language. But we are going to use PHP programming language for back-end side in our project. PHP is stand for Hypertext Preprocessor and it's a general purpose programming language.

## 4.1.2 Implemented tools and platform

So like the tools and technologies also some platform is needed to develop software. Our project we use some platform to build the system. We are using bellows tools and platform.

### 4.1.2.1 Editor

A source-code editor is a text editor program designed specifically for editing source code. Source-code editors are a fundamental programming tool. For developing our system we use Sublime Text code editor. Sublime Text have is variety use of software area. And it supports cross platform.

### 4.1.2.2 Web Server

A web server is a system that delivers content or services to end users over the internet. In our project we uses Apache server. Apache is free and open source software to use. Develop under the auspice of the Apache Software Foundation. It supports Cross platform. It has many feature and variety of use.

### 4.1.2.3 Database Server

Database server is the term used to refer to the back-end system of a database application using client/server architecture. Perform tasks such as data analysis, storage, data manipulation, archiving, and other non-user specific tasks.

We have followed Relational Database Management System. We use MySQL server. MySQL is an open source Relation Database Management System.

## 4.2 Class Diagram

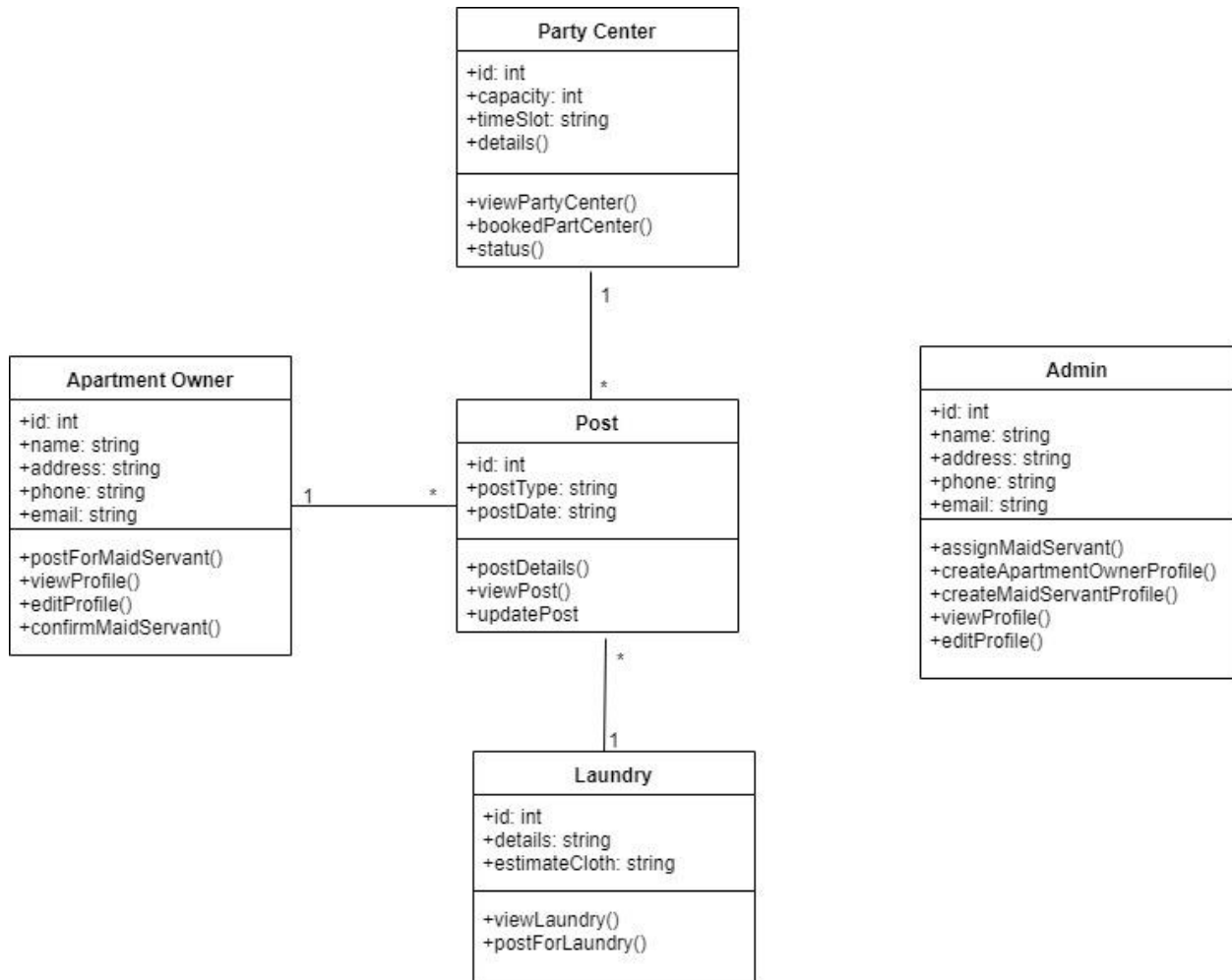


Figure 4.1: Class Diagram

## 4.3 Database Design Diagram

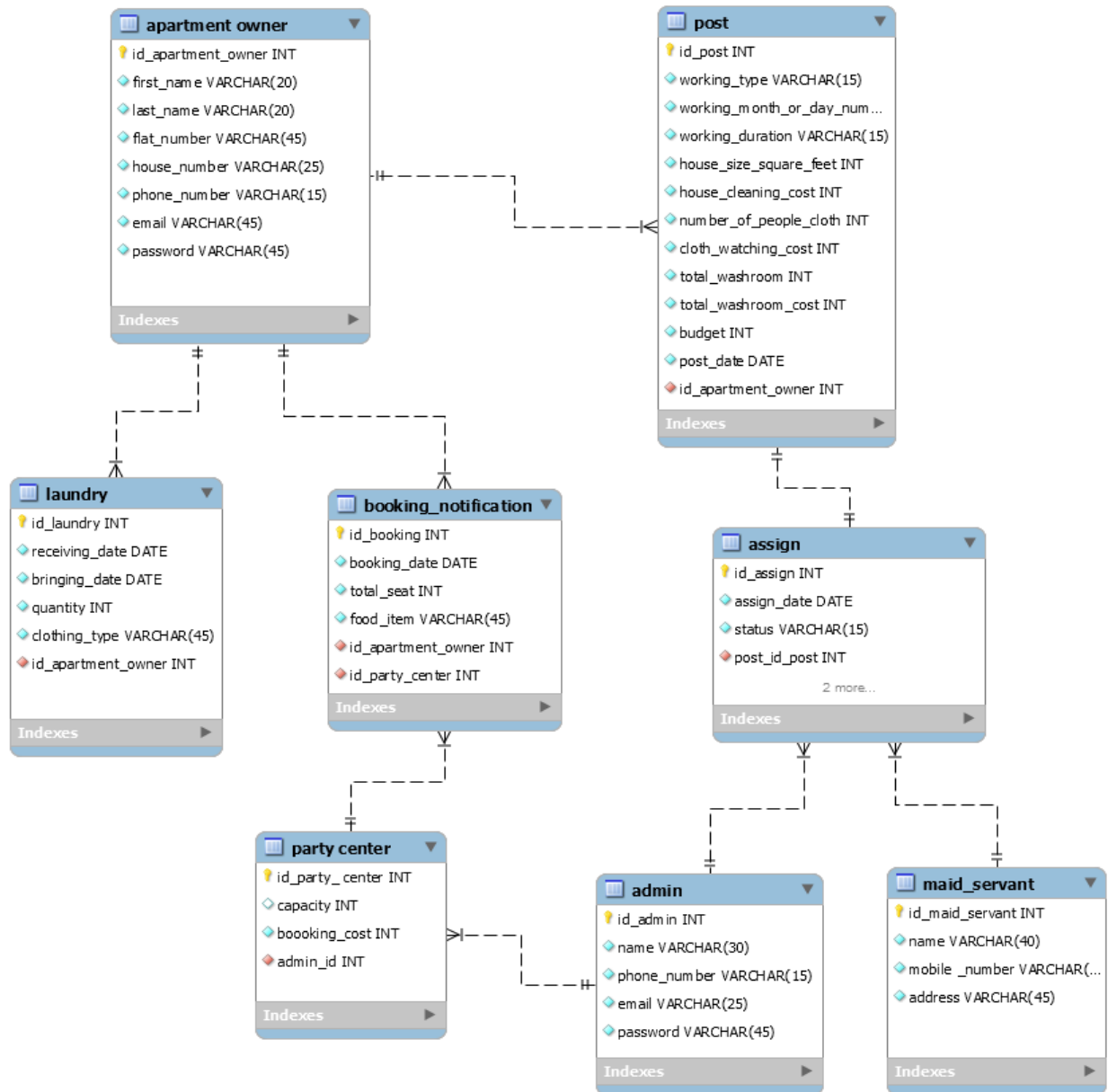


Figure 4.2: Database Diagram

# **Chapter 5**

## **System Test**

## 5.1 Testing Features

Feature Testing is refer to making changes in software system to add new features or to make modifications existing features. A test set is need to writing for test the feature and function of the system. All feature and function are not same. Different function have different goal. Feature test makes an application more useful, intuitive, reliable, secured, scalable and effective.

### 5.1.1 Feature to be tested

#### User (Apartment Owner) Part

Features	Priority	Description
Login	3	User must be authenticated by login
Logout	2	Session must be destroyed after logout
Post for Laundry Service	2	Laundry Service post data must be inserted properly
See Laundry Service Details	1	Apartment owner can able to view the Laundry Service Details
See recent post	3	Apartment Owner can able to see his recent post
Edit Post	3	Apartment Owner can edit there Recent Post
Booking Party Center	2	Apartment Owner can ablebook the party center
See Party Center Details	1	Apartment owner can able to view the Party Center Details



## Admin Part

Features	Priority	Description
Login	3	Admin must be authenticated by login
Logout	2	Session must be destroyed after logout
Create Apartment Owner Profile	3	Admin can create the Apartment Owner profile
Confirm Party Center booking	3	Admin confirm Party Center booking request
See recent post	2	Admin can see Apartment Owner recent post
Create Maid-Servant Profile	3	Admin can create the maid-servant profile
Edit Profile	2	Admin can edit maid-servant profile
Delete Profile	2	Admin can delete both apartment owner and maid-servant profile

**Here, 1 = Low Priority, 2 = Medium Priority, 3 = High Priority**

## 5.2 Testing Strategy

Test Strategy is a plan for defining the testing approach. It's a guideline followed to achieve the test objective and execution of test types. Test strategy work with test objective, test environment, test approach, automation tools and strategy, and risk analysis. It's created based on development design documents. It defined main goals and helps to achieve the goals.

### 5.2.1 Test approach

Test approach means implementation of test strategy. It defined all test plans and test design. Test approach is created by the individual tester according to the module or application means his own views or approaches for that module.

There are two Test approach.

- **Preventative approach:** In preventative approach, tests are designed before the software development.
- **Reactive approach:** In Reactive approach, tests are designed after software development.

#### 5.2.1.1 Black Box Testing

Black box testing is also known as behavioral testing. Black box testing is a testing method that performed without touch of the internal function. Black box testing is manly based on software requirements and specifications. Black box testing can be both of functional or non-functional. In Black box Testing we just focus on inputs and output of the software system

#### 5.2.1.2 Equivalent Class Partitioning

Equivalent Class Partitioning is also a black box technique. In Equivalent Class Partitioning inputs of the software are divided into group. This technique is to eliminate the set of input data and reduce the number of redundant test cases by eliminating those that generate the same output. Equivalent Class Partitioning used large pool of test cases individually. But we apply this technique where in the input field.

#### 5.2.1.3 Boundary Value Analysis

Boundary value analysis based on testing the boundary value of valid and invalid partitions. It's also a black box technique. So this is like Start- End, Lower- Upper, Maximum-Minimum values. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition.

#### 5.2.1.4 White Box Testing

White box testing strategy worked with the internal logic and structure of the code. White box testing is also called as glass, structural or open box. The code structure is known and understood by the tester in white box testing. This type of testing, the code is visible to the tester, and tester must have to know the function.

**Classified of White box testing into some levels:**

- **Unit Testing**
- **Integration Testing**
- **System Testing**

#### 5.2.2 Pass / Fail Criteria

Tester will set the pass and fail criteria. They prepare the pass fail criteria based on item if an item worked then it pass and if an item doesn't work then it's failed. This is not the place to define the detailed pass criteria for each feature, but to describe the process and overall standards for evaluating the test results.

Now I will give the pass / fail criteria below:

- If system crash then it will not be considered as pass case.
- If any items pass 100% times, then it will be considered as pass criteria only.
- If data can't be displayed to the application properly, then it is also to be considered as fail criteria.

### 5.3 Testing Schedule

<b>Test Phase</b>	<b>Time</b>
Testing plan create	1 week
Test specification	2 week
Unit testing	During development time
Component test	1 week
Integration testing	1 week
Validating use cases	1 week
Testing user interfaces	1 week
Load testing	1 week
Performance testing	1 week
Release to production	1 week

## 5.4 Trace Ability Matrix

### Apartment Owner Part

Project Manager			Business Analyst Lead		
QA Lead			Target Implementation Date		
BR#	Category /Functionality/ Activity	Requirement Description	Use Case Reference	Test Case Reference	Comments
BR-1	Functional	Login	Use-case 3.1.1	Test-case 5.6.1	
BR-2	Functional	Post for laundry service	Use-case 3.1.2	Test-case 5.6.2	
BR-3	Functional	View laundry service details	Use-case 3.1.3	Test-case 5.6.4	
BR-4	Functional	Booking party center	Use-case 3.1.4	Test-case 5.6.3	
BR-5	Functional	View party center details	Use-case 3.1.5	Test-case 5.6.5	
BR-6	Functional	Sees previous history	Use-case 3.1.6	Test-case 5.6.7	
BR-7	Functional	Sees recent post	Use-case 3.1.7	Test-case 5.6.8	
BR-8	Functional	View profile	Use-case 3.1.8	Test-case 5.6.9	
BR-9	Functional	Edit profile	Use-case 3.1.9	Test-case 5.6.10	
BR-10	Functional	Get notification confirmation	Use-case 3.1.10	Test-case 5.6.6	

## Admin Part

Project Manager			Business Analyst Lead		
QA Lead			Target Implementation Date		
BR#	Category /Functionality/ Activity	Requirement Description	Use Case Reference	Test Case Reference	Comments
BR-1	Functional	Login	Use-case 3.1.11	Test-case 5.6.11	
BR-2	Functional	Apartment owner registration	Use-case 3.1.12	Test-case 5.6.14	
BR-3	Functional	Party center booking confirmation	Use-case 3.1.13	Test-case 5.6.12	
BR-4	Functional	View history	Use-case 3.1.14	Test-case 5.6.15	
BR-5	Functional	View recent post	Use-case 3.1.15	Test-case 5.6.16	
BR-6	Functional	Delete profile	Use-case 3.1.16	Test-case 5.6.18	
BR-7	Functional	Update's party center details	Use-case 3.1.17	Test-case 5.6.13	
BR-8	Functional	Update's laundry service details	Use-case 3.1.18	Test-case 5.6.13	
BR-9	Functional	Edit profile of apartment owner & maid-servant	Use-case 3.1.19	Test-case 5.6.17	

## 5.5 Testing Environment

Testing environment means prepare the environment that support to execution the test with software, hardware and network configures. Test bed or test environment is configured as per the need of the application under test.

For build the environment of testing, some key area need to setup:

- Test data
- Database server
- Client's operating system
- Front end running environment

- Browser
- System and application
- Network

People are involved with test environment setup:

- System Admins
- Developers
- Testers

## 5.6 Test Cases

Test case means set some rules and regulations or conditions by which it can be determined whether a system can be able to meet the works or requirements under test cases correctly. Test cases process can help to find the problems in the requirements or design of an application. In test case there have some elements. Such as:

- Test Case ID
- Test Scenario
- Test Case Description
- Prerequisite
- Test Data
- Expected Result
- Actual Result
- Etc

## 5.6.1 Log In

<b>Test case #1</b>		<b>Test case name:</b> Log in			
<b>System:</b> Apartment colony service system		<b>Subsystem:</b>			
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019			
<b>Executed by:</b>		<b>Executed date:</b>			
<b>Short description:</b> The registered users need to login to the application. And before that our application will check the authentication and authorization.					
<b>Pre-conditions:</b>					
<ul style="list-style-type: none"> <li>• Users are always redirected to the login page if they have not authenticated by our application.</li> <li>• Assume that, the email is <a href="mailto:rohomot236@gmail.com">rohomot236@gmail.com</a> and password is “123456”</li> </ul>					
Step	Email	Password	Response	Pass/Fail	Comment
1	roho@gmail.com	123411	Invalid	Fail	
2	pakhi@gmail.com		Password can't be blank	Fail	
3		123456	Email can't be blank	Fail	
4	<a href="mailto:piro236@gmail.com">piro236@gmail.com</a>	123456	Successfully login into the application	Pass	
<b>Post-conditions:</b> Apartment owner will successfully login to the application.					

### 5.6.2 Apartment Owner Post for Laundry Service

<b>Test case #2</b>		<b>Test case name:</b> Post for Laundry Service		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b>		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment Owner provided all require information for laundry service post then submit the post.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• Users must be authenticated.</li> <li>• User type must be Apartment Owner.</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	All required fields are not filled	Filled up require fields	Fail	
2	Filled up all required fields	Post is Successful	Pass	
<b>Post-conditions:</b> Post for Laundry Service is Successful.				

### 5.6.3 Apartment Owner Booking Party Center

<b>Test case #3</b>		<b>Test case name:</b> Booking Party Center		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b>		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> For booking party center firstly apartment owner need to provided all require information then submit the post.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• Users must be authenticated.</li> <li>• User type must be Apartment Owner.</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	All required fields are not filled	Filled up require fields	Fail	
2	Filled up all required fields	Post is Successful	Pass	
<b>Post-conditions:</b> Post for Laundry Service is Successful.				



### 5.6.4 Apartment owner view laundry service details

<b>Test case #4</b>		<b>Test case name:</b> View laundry service details		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b>		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment can able to view laundry service details				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• <b>Users must be authenticated.</b></li> <li>• <b>User type must be Apartment Owner.</b></li> </ul>				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Laundry service details data is shown properly.	All require data is appear.	Pass	
<b>Post-conditions:</b> Laundry service information data is showing to apartment owner				

### 5.6.5 Apartment owner view party center booking details

<b>Test case #5</b>		<b>Test case name:</b> View party center booking details		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b>		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 15-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment can able to view party center booking details				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• <b>Users must be authenticated.</b></li> <li>• <b>User type must be Apartment Owner.</b></li> </ul>				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Party center booking details data is shown properly.	All require data is appear.	Pass	
<b>Post-conditions:</b> Party center booking information data is showing to apartment owner				

## 5.6.6 Get party center booking confirmation message

<b>Test case #6</b>		<b>Test case name:</b> Get party center booking confirmation message		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 10-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> After post part center booking request, admin confirm or reject the party center booking request. And apartment owner get that confirmation message				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be authenticated</li> <li>• Apartment Owner have a party center booking request</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	Admin is not taking any action	No action has been occurred then.	Fail	
2	Admin accept the part center booking request.	Apartment owner get the confirming notification	Pass	
<b>Post-conditions:</b> Apartment owner get the confirmation message of part center				

## 5.6.7 See previous history

<b>Test case #7</b>		<b>Test case name:</b> See previous history		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 20-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment owner can see their previous history if there have some previous activities. Previous history data appear to apartment owner.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment owner have some previous activities</li> </ul>				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	All required information is not showing.	Require data not shown.	Fail	
2	Previous history data is shown properly.	All require data is appear.	Pass	
<b>Post-conditions:</b> Apartment Owner can able to see his previous history				

## 5.6.8 See recent post

<b>Test case #8</b>		<b>Test case name:</b> See recent post		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 20-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment owner also can see his recent post. Recent post data appear to apartment owner if he has a recent post.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment owner have some previous activities</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	All required information is not showing.	Require data not shown.	Fail	
2	Previous history data is shown properly.	All require data is appear.	Pass	
<b>Post-conditions:</b> Recent post data displayed to apartment owner.				

## 5.6.9 View own profile

<b>Test case #9</b>		<b>Test case name:</b> View own profile		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Every apartment owner has their own profile, and they can view their own profile.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment owner have a profile</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	Profile is not showing.	There is no profile to appear.	Fail	
2	Profile is showing	Profile is appear	Pass	
<b>Post-conditions:</b> Apartment Owner can able to view his own profile				

## 5.6.10 Edit Profile

<b>Test case #10</b>		<b>Test case name:</b> Edit profile		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment owner go to their profile and they can edit their own profile.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Apartment Owner</li> <li>• Apartment Owner must be logged in</li> <li>• Apartment Owner have a profile</li> </ul>				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	All required fields are not filled yet.	Fields have not been completed yet.	Fail	
2	All input filed is filled up by the apartment owner.	The fields have been completed and the post also.	Pass	
<b>Post-conditions:</b> Apartment Owner edited their own profile				

### 5.6.11 Login for Admin

<b>Test case #11</b>		<b>Test case name:</b> Log in for Admin			
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A			
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 15-Mar-2019			
<b>Executed by:</b>		<b>Executed date:</b>			
<b>Short description:</b> The admin need to login for enters into the application. And before the login our application will check the authentication and authorization.					
<b>Pre-conditions:</b>					
<ul style="list-style-type: none"> <li>User type must be Admin</li> <li>Assume that, the email is <a href="mailto:akash10@gmail.com">akash10@gmail.com</a> and password is “123456”</li> </ul>					
Step	Email	Password	Response	Pass/Fail	Comment
1	akash@gmail.com	123411	Wrong email and password	Fail	
2	akash10@gmail.com		Wrong email and password	Fail	
3		123456	Email can't be blank	Fail	
4	<a href="mailto:akash10@gmail.com">akash10@gmail.com</a>	123456	Successfully login into the application	Pass	
<b>Post-conditions:</b> Admin will successfully login into the application.					

## 5.6.12 Confirm or Reject party center booking request

<b>Test case #12</b>		<b>Test case name:</b> Confirm or Reject party center booking request		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 20-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> When apartment owner post for part center booking then that booking request comes to admin, and admin confirm or reject the party center booking request.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be authenticated</li> <li>• Admin have a party center booking request</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	Admin is not taking any action	No action has been occurred then.	Fail	
2	Admin accept the part center booking request.	Apartment owner get the confirming notification	Pass	
<b>Post-conditions:</b> Apartment owner booked the party center after the confirmation				



### 5.6.13 Update party center and laundry service details

<b>Test case #13</b>		<b>Test case name:</b> Update party center and laundry service details		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 20-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> When it needed admin can update the details information of party center pr laundry service				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be authenticated</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	Admin is not updating the details	No action has been occurred then.	Fail	
2	Details information is updated	The new updated information is displayed	Pass	
<b>Post-conditions:</b> Updated information show to the apartment owner				

## 5.6.14 Registration apartment owner profile

<b>Test case #14</b>		<b>Test case name:</b> Registration apartment owner profile		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 01-Mar-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> The admin create apartment owner profile and give it to the apartment owner. And apartment owner enter the system by using this profile.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be authenticated.</li> <li>• Admin have apartment owner information</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	All required fields are not filled yet.	Fields have not been completed yet.	Fail	
2	All input filed is filled up by the apartment owner.	Adding apartment owner is successful	Pass	
<b>Post-conditions:</b> An apartment owner profile is created.				

## 5.6.15 Admin view previous history

<b>Test case #15</b>		<b>Test case name:</b> Admin view previous history		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Admin can view all type of previous history data. Previous history data appear to apartment owner.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be logged in</li> <li>• Apartment owner have some previous activities record</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	Click on “Previous History” button	Empty history. There is no previous history data	Fail	This is early stage. So here, no apartment owner activities is recorded
2	Click on “Previous History” button	Show previous history data	Pass	
<b>Post-conditions:</b> Admin can able to see all type of previous history				

## 5.6.16 Admin view recent post

<b>Test case #16</b>		<b>Test case name:</b> Admin view recent post		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 10-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Admin can view recent post that is performed by apartment owner.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be logged in</li> <li>• Apartment owner have some recent post</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	Click on “Recent Post” button	Empty post. There is no previous history data	Fail	This is early stage. So here, apartment owner haven’t perform any post yet
2	Click on “Recent Post” button	Show all recent post	Pass	
<b>Post-conditions:</b> Admin can able to see all recent post				

## 5.6.17 Admin Edit Profile

<b>Test case #17</b>		<b>Test case name:</b> View profile		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment owner go to their profile and they can edit their profile.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be logged in</li> <li>• Apartment Owner have a profile</li> </ul>				
Step	Action	Response	Pass/Fail	Comment
1	All required fields are not filled yet.	Fields have not been completed yet.	Fail	
2	All input filed is filled up properly.	The fields have been filled-up and profile is updated.	Pass	
<b>Post-conditions:</b> Apartment Owner profile is updated				

## 5.6.18 Admin Delete Profile

<b>Test case #18</b>		<b>Test case name:</b> Delete profile		
<b>System:</b> Apartment colony service system		<b>Subsystem:</b> N/A		
<b>Designed by:</b> Kabir Hossain		<b>Designed date:</b> 05-Apr-2019		
<b>Executed by:</b>		<b>Executed date:</b>		
<b>Short description:</b> Apartment owner go to their profile delete the profile.				
<b>Pre-conditions:</b>				
<ul style="list-style-type: none"> <li>• User type must be Admin</li> <li>• Admin must be logged in</li> <li>• Apartment Owner have a profile</li> </ul>				
<b>Step</b>	<b>Action</b>	<b>Response</b>	<b>Pass/Fail</b>	<b>Comment</b>
1	Go to profile and hit the “Delete” button	The profile is deleted	Pass	
<b>Post-conditions:</b> Apartment Owner or Maid-Servant profile is deleted				

# **Chapter 6**

## User Manual

## 6.1 Apartment owner login page

First of all, the apartment owner will need to login to our application to use this application. Apartment owner login the system using the email and password that provided by admin. Now I will provide the screenshot of login page below.

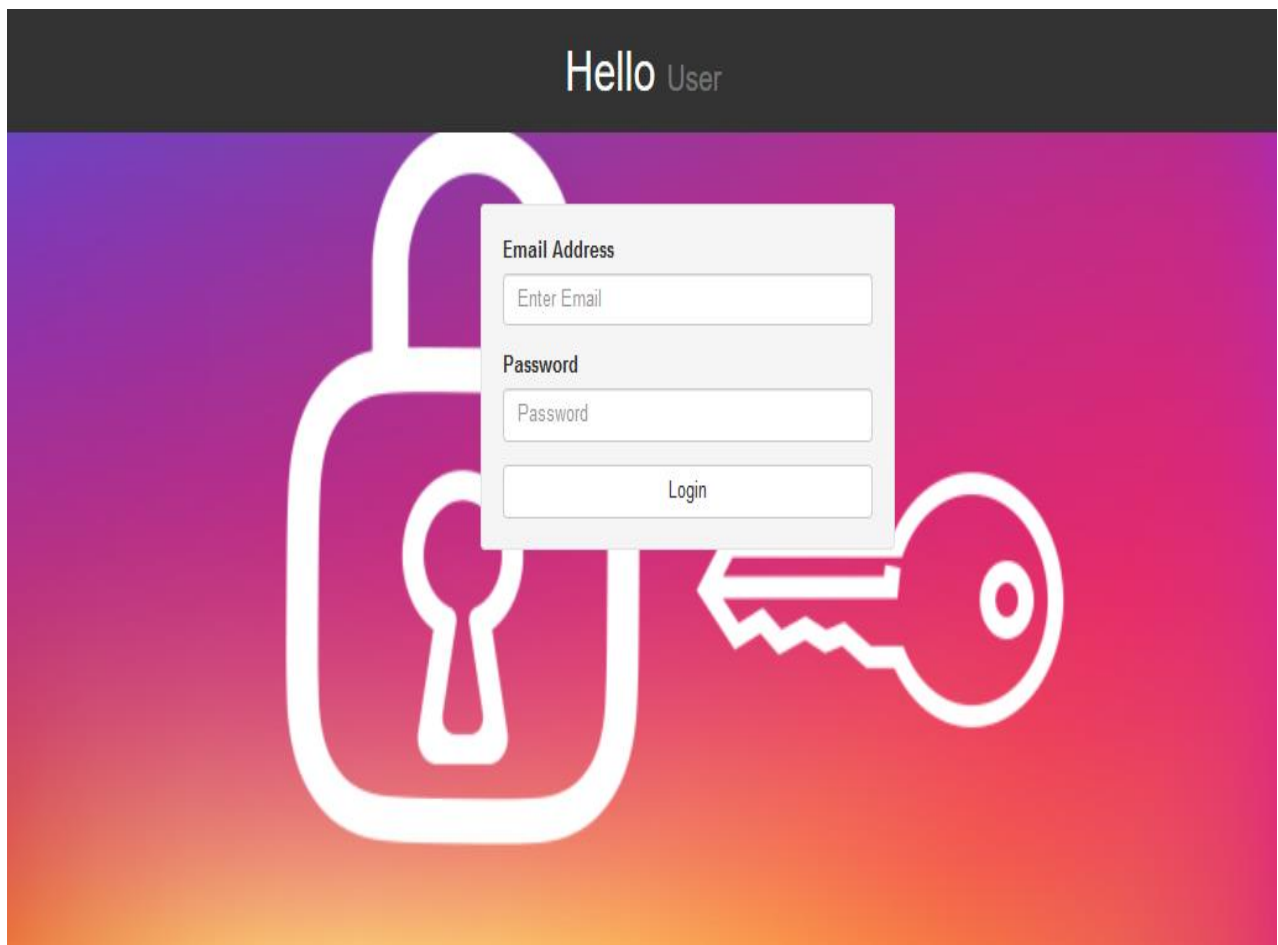


Figure 6.1: Login page



## 6.2 Home page (Apartment owner)

After successfully logging in, the apartment owner enters the system and is able to see the main page. Now I will provide the screenshot of login page below.

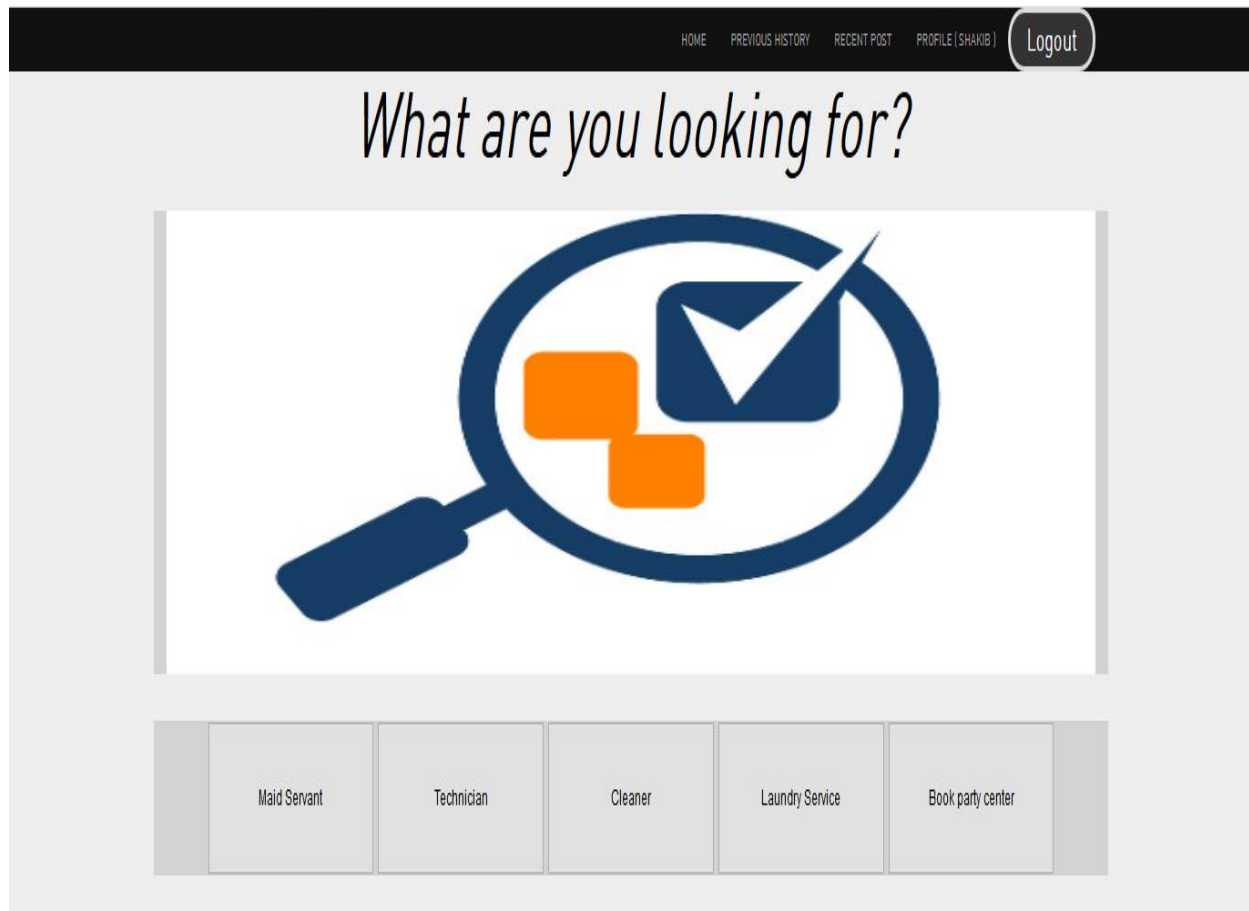
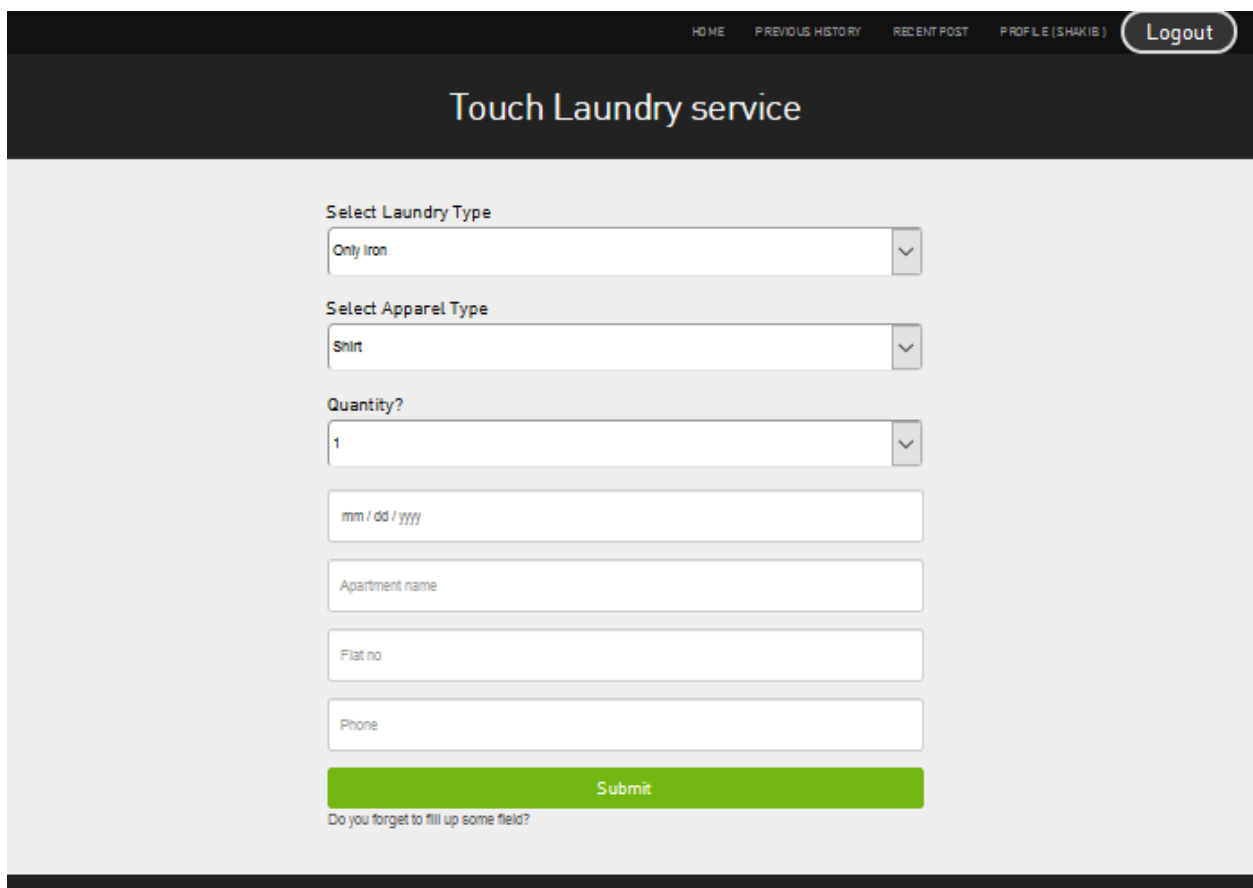


Figure 6.2: Home page of apartment owner

## 6.3 Apartment owner post for laundry service

The Apartment owner can post for laundry service when they have need. For post the laundry service they have to provide some information. Now I will provide the screenshot of the page below.

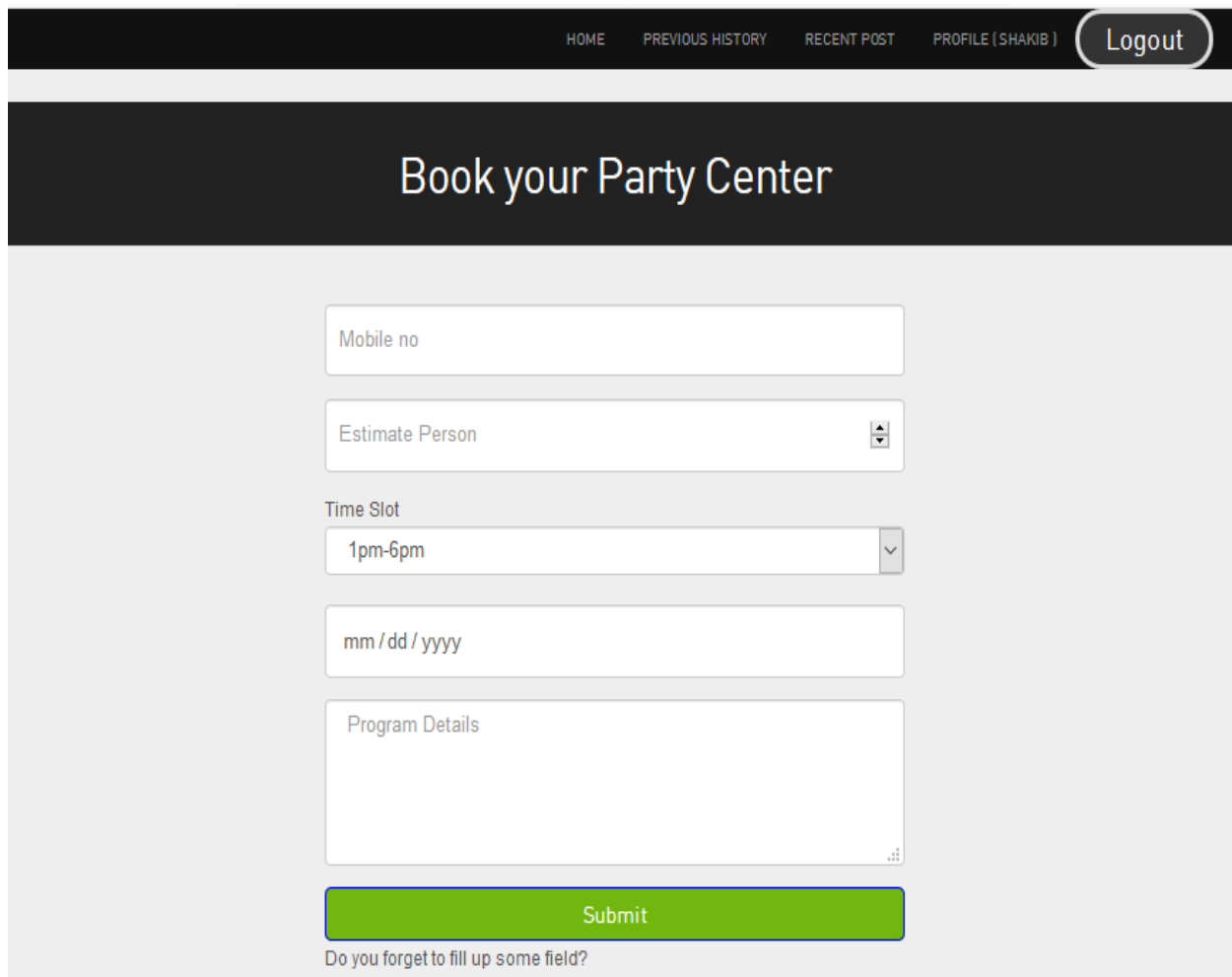


The screenshot shows a web interface for posting a laundry service request. At the top, there is a navigation bar with links for HOME, PREVIOUS HISTORY, RECENT POST, and PROFILE (SHAKIB), along with a Logout button. The main heading is "Touch Laundry service". The form consists of several input fields: "Select Laundry Type" (dropdown menu with "Only Iron" selected), "Select Apparel Type" (dropdown menu with "Shirt" selected), "Quantity?" (dropdown menu with "1" selected), a date field (placeholder: mm / dd / yyyy), "Apartment name", "Flat no", and "Phone". A green "Submit" button is at the bottom, with a warning message "Do you forget to fill up some field?" below it.

Figure 6.3: Apartment owner post for laundry service

## 6.4 Apartment Owner post for booking party center

Apartment owner can post for party center booking. For post the party center they have to provide some information. Now I will provide screenshots of the page below.



The screenshot shows a web interface for booking a party center. At the top, there is a navigation bar with links for HOME, PREVIOUS HISTORY, RECENT POST, and PROFILE [SHAKIB], along with a Logout button. Below this is a dark header with the text "Book your Party Center". The main content area contains a form with the following fields:

- Mobile no: A text input field.
- Estimate Person: A dropdown menu.
- Time Slot: A dropdown menu with "1pm-6pm" selected.
- mm / dd / yyyy: A date input field.
- Program Details: A large text area for entering details.
- Submit: A green button.

Below the Submit button, there is a message: "Do you forget to fill up some field?"

Figure 6.4: Apartment owner post for booking party center

## 6.5 Apartment Owner can view their profile

Apartment owner can able to see their own profile. Now I will provide the screenshot of the page below.

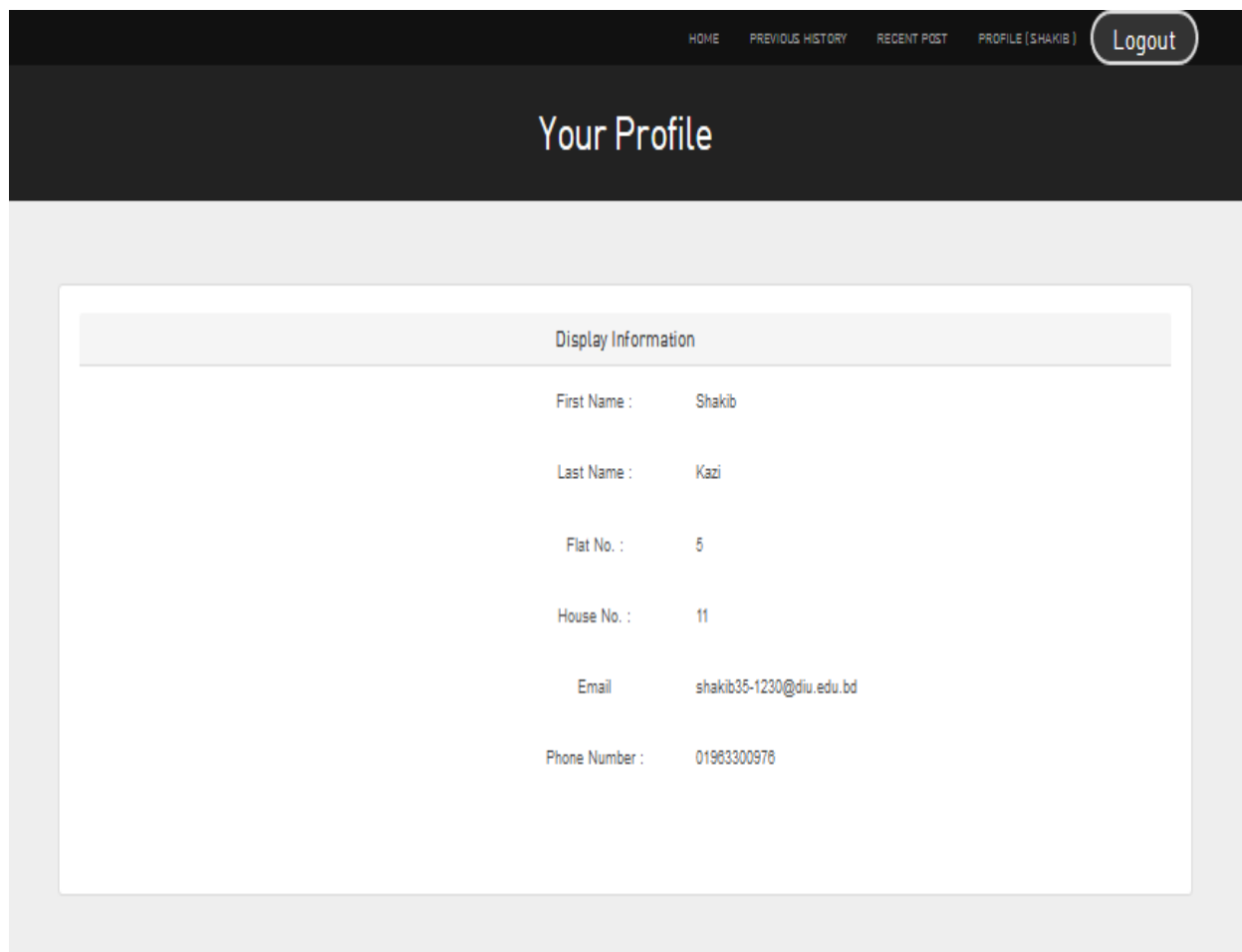
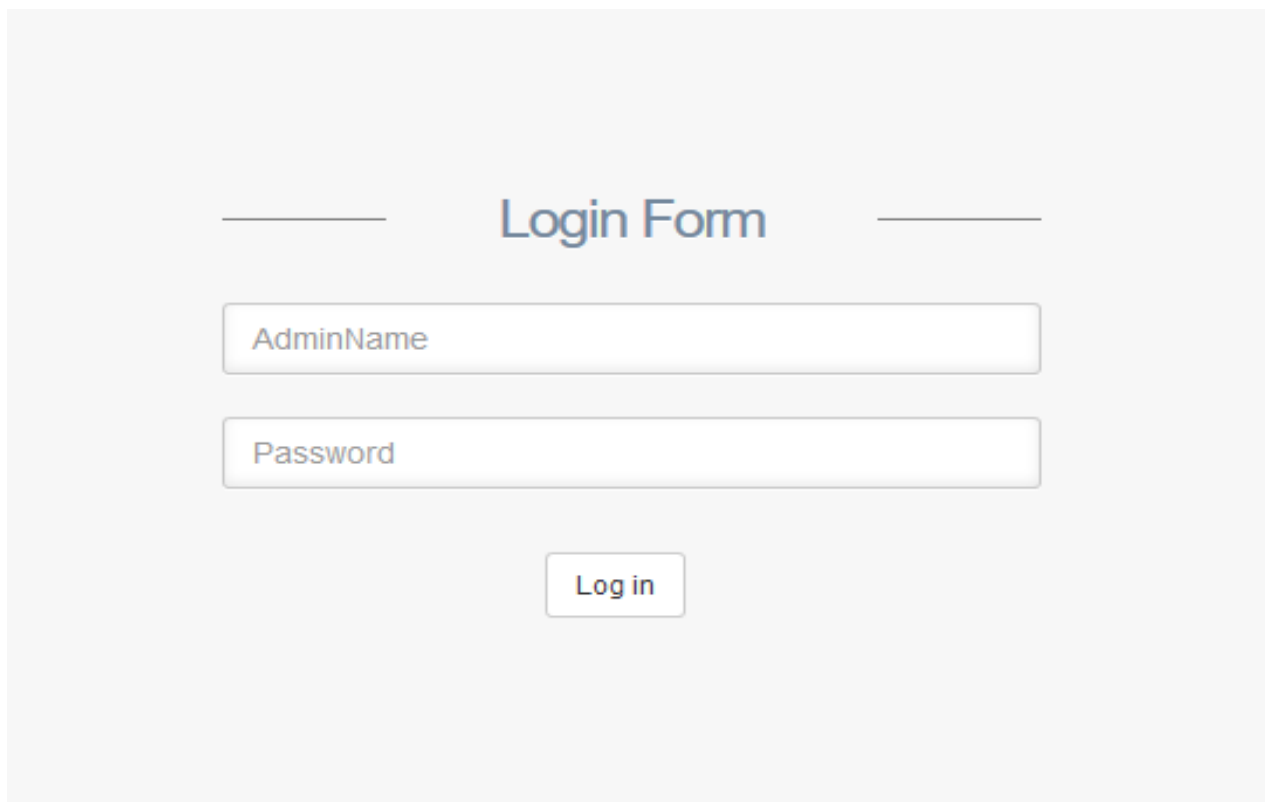


Figure 6.5: Apartment owner view his/her own profile

## 6.6 Admin login page

First of all, the administrator has to login to the application to use this application. Administrators must login to the system using email and password. Now I will provide screenshots of the page below.

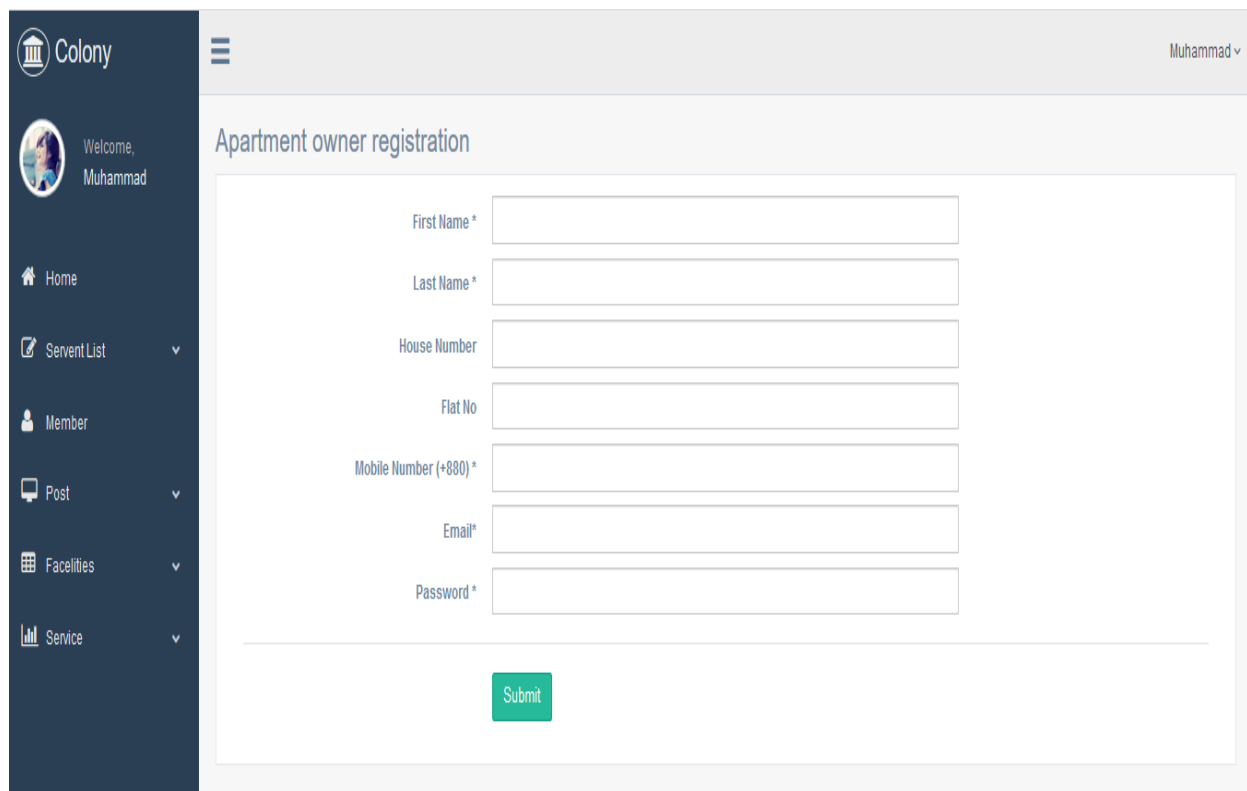


The screenshot displays a simple login interface. At the top center, the text "Login Form" is displayed in a blue font, flanked by two horizontal lines. Below this, there are two input fields: the first is labeled "AdminName" and the second is labeled "Password". Both fields are white with a light gray border. Below the input fields, there is a single button labeled "Log in" with a light gray border and a white background.

Figure 6.6: Admin login page

## 6.7 Apartment owner registration

The user must log in first to get access to our application. But users have not signed up before. Then, first of all, he has to register in our application. To register, the user has to go to the admin. After that, you'll have to provide extravagant information to the admin. The admin will provide the user with an email and password through which the user can access the system. Now I will provide user snapshots Register page below.



The screenshot shows a web application interface for apartment owner registration. On the left is a dark blue sidebar with a 'Colony' logo and a navigation menu containing: Home, ServentList, Member, Post, Facilities, and Service. The main content area has a header with a hamburger menu icon and the name 'Muhammad'. Below the header, the title 'Apartment owner registration' is displayed. The registration form consists of several input fields: 'First Name \*', 'Last Name \*', 'House Number', 'Flat No', 'Mobile Number (+880) \*', 'Email\*', and 'Password \*'. A green 'Submit' button is located at the bottom center of the form area.

Figure 6.7: Apartment owner registration form

## 6.8 Apartment owner list & add apartment owner

Admin can view apartment list & apartment owner's all information .Besides, admin can add apartment owner into the application by registration. Now I will provide the screenshot of the page below.

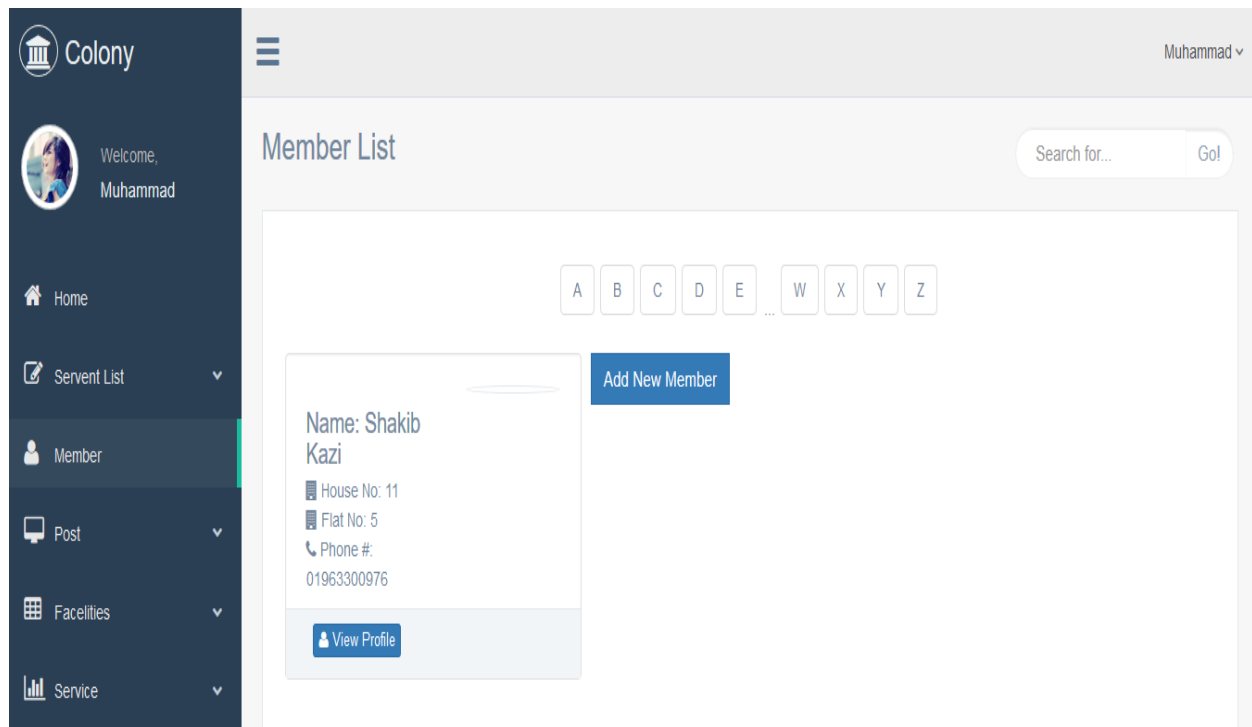


Figure 6.8: Apartment owner list & add apartment owner

# **Chapter 7**

## **Conclusion**



## 7.1 GitHub Link

## 7.2 Project Summary

I have started the development part on January. As like as any others project, you have to put yours very hard working, patience, dedication, and concentration to complete the project. There are many factors comes to when a project going to the development phase, such as fulfilling the stakeholder requirements properly, etc.

A proper requirements analysis then it helps the development phase a lot. We analysis the requirements of our project firstly then we are going to the next step design specification.

Database is an important factor of any application system. That's why in the next step we are focusing to create the database design. We have designed the database diagram having tables with the proper relationship. The admin part also can be called maintainers part plays a big role in our system. So the next part is the user interface. Making the user interface is simple as if a user can understand it easily. After that, we are checking the all thing again and move to the core functionality of the project.

It's not easy task to complete project but if it to be completed then the project is not the and actually. At the end of complete the project, you have to make sure that your project functionality works fine. For that, you have to come in the testing part, it's part of quality assurance. The responsibility of quality assurance is to find the vulnerability of the system. If any bug can be found before the system release then there is a change to fix that bug. So testing the project we have assured the quality of the project.

## 7.3 Limitations

For developing this project, I have faced some limitations. Now I will describe those in brief.

- **Payment Method:** In this application, there are not having any payment method directly. So the user can't make any payment transaction through our system. User pays the payment manually.
- **Only Web Version:** Our system is only web-based but there are many other platforms is popular such as Mobile version, iOS version, etc.

## 7.4 Obstacles and Achievements

When a project is developing then you have face many obstacles and those obstacle create challenges and you have to overcome this challenges. Without challenges you can't learn anything, challenge comes with opportunity and here your change to grave this opportunity.

By doing this project we have learned how collect the requirements and how to analysis them. Brainstorming is must for doing any project you have to do it for sure. The system analysis and database design that we learn properly by doing this project. Our supervisor helps us in every step of this project

Also there are some obstacles and achievement in our project and that is below:

- **Requirements getting from stakeholders:** There are many types of stakeholders in this system and everyone have different functionality. It would be better, if we could gather requirements from them directly.
- **Scope Change:** When we check the requirements then some features has been added and a few of feature get cutting. When it was needed we did brainstorming for that.

## 7.5 Future Scope

I learned a lot while finishing this project. To build and improve this project ,I have met some young entrepreneurs and enthusiasts. I thank all of them very much.

Besides, I am grateful to them because I have been able to complete my project by adopting their important opinions and discussions. It will help me in the future to work with similar projects.

## 7.6 References

I have gained some knowledge from some platforms. Obviously I will mention those references. For making my project successful those resources help me a lot. Not I will mention the names below.

- ❖ [www.google.com](http://www.google.com)
- ❖ [www.youtube.com](http://www.youtube.com)
- ❖ [www.github.com](http://www.github.com)
- ❖ [www.wikipedia.com](http://www.wikipedia.com)
- ❖ [www.php.net](http://www.php.net)
- ❖ [www.mysql.com](http://www.mysql.com)
- ❖ [www.mysql.com/products/workbench](http://www.mysql.com/products/workbench)
- ❖ [www.getbootstrap.com](http://www.getbootstrap.com)
- ❖ [www.w3schools.com](http://www.w3schools.com)
- ❖ [www.jquery.com](http://www.jquery.com)
- ❖ [www.sublimetext.com/3](http://www.sublimetext.com/3)
- ❖ [www.apachefriends.org](http://www.apachefriends.org)