



Daffodil
International
University

Maid-Servant Assigning: For a Colony Based Area

By

Md. Shakib Kazi
152-35-1230

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 “**Assigning Maid-Servant: For a Colony Based Area**”, submitted by **Md. Shakib Kazi, 152-35-1230** 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 Efata**, 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.

shakib

Md. Shakib Kazi
ID: 152-35-1230
Batch : 17th
Department of Software Engineering
Faculty of Science & Information Technology
Daffodil International University

Certified by:



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

Acknowledgement

After all, I was successful as I successfully replaced the final semester. And so I'm pleased with my Almighty. From the beginning of my university career, I learned a lot about the knowledge about software engineering and core computer science about my course teachers. Moreover, they teach our ethics, ethics, and modesty.

As well, I am very grateful to my parents that I was always supported by them. They always inspire me to my opinion priorities and as well.

My supervisor to allow me to work with this project. **Mr. Iftekharul Alam Efat** too much appreciated. He always supports me to make this project successful. Especially I am very grateful to my brother, who always admires my professionalism.

Abstract

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
Abstract	v
Chapter 1	1
1.1 Project Overview	2
1.2 Project Purpose.....	2
1.2.1 Background.....	2
1.2.2 Benefits & Beneficiaries.....	2
1.2.3 Goals	3
1.3 Stakeholders	3
1.4 Proposed System Model	4
1.5 Project Schedule	4
1.5.1 Gantt Chart	5
1.5.2 Release Plan or Milestone	6
Chapter 2.....	7
2.1 Functional Requirements	8
2.1.1 Login for Apartment owner	9
2.1.2 Apartment owner post for maid-servant.....	9
2.1.3 Apartment owner confirm assigning the maid-servant.....	9
2.1.4 Apartment owner view previous history	9
2.1.5 Apartment owner can view recent post	9
2.1.6 Apartment owner can view their profile.....	10
2.1.7 Apartment owner can edit their profile	10
2.1.8 Apartment owner can view maid-servant profile.....	10
2.1.9 Login for Admin	10
2.1.10 Registration apartment owner	10
2.1.11 Admin Create Maid-servant profile	11
2.1.12 Admin View History	11

2.1.13 Admin View Recent Post	11
2.1.14 Admin Delete Apartment Owner and Maid-Servant profile.....	11
2.1.15 Admin Edit Apartment Owner and Maid-Servant Profile	11
2.1.16 Send some maid-Servant profile	12
2.2 Data Requirements	12
2.3 Performance Requirements	12
2.3.1 Speed and Latency Requirements	12
2.3.2 Precision or Accuracy Requirements	13
2.3.3 Capacity Requirements.....	13
2.4 Dependability Requirements	13
2.4.1 Reliability Requirements	14
2.4.2 Availability Requirements	14
2.4.3 Robustness or Fault-Tolerance Requirements	14
2.4.4 Safety-Critical Requirements.....	14
2.5 Maintainability & Supportability Requirements	14
2.5.1 Maintenance Requirements	15
2.5.2 Supportability Requirements	15
2.5.3 Adaptability Requirements	15
2.5.4 Scalability or Extensibility Requirements	16
2.6 Security Requirements	16
2.6.1 Access Requirements	16
2.6.2 Integrity Requirements	16
2.6.3 Privacy Requirements	16
2.7 Usability and Human-Interaction Requirements	16
2.7.1 Ease of Use Requirements	17
2.7.2 Personalization and Internationalization Requirements	17
2.7.3 Understandability and Politeness Requirements.....	17
2.7.4 Accessibility Requirements	17
2.7.5 User Documentation Requirements	17
2.7.6 Training Requirements	17
2.8 Look and Feel Requirements.....	17
2.8.1 Appearance Requirements	17

2.8.2 Style Requirements	18
2.9 Operational and Environmental Requirements.....	18
2.9.1 Expected Physical Requirements	18
2.9.2 Requirements for Interfacing with Adjacent Systems	18
2.9.3 Release Requirements	18
2.10 Legal Requirements	18
2.10.1 Compliance Requirements.....	18
2.10.2 Standards Requirements	18
Chapter 3.....	19
3.1 Use Case Diagram	20
Use Case Description.....	22
3.1.1 Login for apartment owner	22
3.1.2 Apartment owner post for maid-servant.....	22
3.1.3 Apartment Owner confirm assigning the maid servant	23
3.1.4 Apartment Owner sees previous history.....	23
3.1.5 Apartment Owner sees recent post.....	24
3.1.6 Apartment Owner can view profile.....	24
3.1.7 Apartment Owner can edit profile.....	25
3.1.8 Apartment Owner can visit maid-servant profile.....	25
3.1.9 Login for Admin	26
3.1.10 Apartment Owner registration	26
3.1.11 Admin create maid-servant profile.....	27
3.1.12 Admin can sees previous history.....	27
3.1.13 Admin can sees recent post.....	28
3.1.14 Admin can delete profile	28
3.1.15 Admin Can Edit Profile	29
3.1.16 Admin send assign maid-servant list.....	29
3.2 Activity Diagram.....	30
3.2.1 Login for Apartment Owner	30
3.2.2 Apartment owner post for maid-servant.....	31
3.2.3 Apartment Owner confirm maid-servant.....	32
3. 2.4Apartment Owner Sees Previous History	33

3.2.5 Apartment Owner Sees Recent Post.....	34
3.2.6Apartment Owner Can View Profile.....	35
3.2.7 Apartment Owner Can Edit his Profile	36
3.2.8 Apartment Owner Can Visit Maid-Servant Profile.....	37
3.2.9 Login for Admin	38
3.2.10 Admin Create apartment owner Registration.....	39
3.2.11 Admin Create maid-servant profile.....	40
3.2.12 Admin Sees Previous History	41
3.2.13 Admin Sees Recent Post.....	42
3.2.14 Admin can edit and delete apartment owners & maid-servant profile	43
3.2.15 Admin send maid-servant list to apartment owner.....	44
3.3 Sequence Diagrams	45
3.3.1 Login for apartment owner	45
3.3.2 Apartment owner post for maid-servant.....	46
3.3.3 Apartment Owner confirm a maid-servant	46
3.3.4 Apartment Owner sees his previous history	47
3.3.5 Apartment Owner sees their recent post.....	47
3.3.6 Apartment Owner Can View Profile.....	48
3.3.7 Apartment Owner edits their profile	48
3.3.8Apartment Owner Can Visit maid-servant profile.....	49
3.3.9 Login for Admin	49
3.3.10 Admin registration for Apartment Owner	50
3.3.11Admin create maid-servant.....	50
3.3.12 Admin view Apartment Owner’s post history	51
3.3.13 Admin view Apartment Owner’s recent post	51
3.3.14Admin edits Apartment Owner profile.....	52
3.3.15 Admin deletes Apartment Owner profile	52
3.3.16Admin send maid-servant list	53
Chapter 4.....	54
4.1 Development tools and technology	55
4.1.1 User Interface Technology	55
4.1.2 Implemented tools and platform	56

4.2 Class Diagram	56
4.3 Database Design Diagram.....	57
Chapter 5.....	59
5.1 Testing Features	59
5.1.1 Features to be tested	60
5.2 Testing Strategy.....	61
5.2.1 Test approach.....	61
5.2.2 Pass / Fail Criteria	62
5.3 Testing Schedule	63
5.4 Trace Ability Matrix	63
5.5 Testing Environment	64
5.6 Test Cases	65
5.6.1 Log In.....	65
5.6.2 Post for maid-servant	66
5.6.3 Confirm maid-servant	67
5.6.4 See previous history	68
5.6.5 See recent post	69
5.6.6 View profile.....	70
5.6.7 Edit Profile.....	71
5.6.8 View Maid-Servant Profile.....	72
5.6.9 Login for Admin	73
5.6.10 Create apartment owner profile.....	74
5.6.11 Create maid-servant profile	75
5.6.12 Admin view previous history.....	76
5.6.13 Admin view recent post.....	77
5.6.14 Admin edits profile	78
5.6.15 Admin deletes Profile.....	79
5.6.16 Send maid-servant profile list to apartment owner	80
Chapter 6.....	82
6.1 Apartment owner login page	83
6.2 Home page (Apartment owner)	84
6.3 Apartment owner post for maid-servant.....	85

6.4 Apartment Owner sees recent post.....	86
6.5 Apartment Owner sees recent post details.....	87
6.6 Apartment Owner sees previous history	88
6.7 Apartment Owner confirm assigning the maid servant.....	89
6.8 Apartment Owner can view their profile.....	90
6.9 Admin login page.....	90
6.10 Apartment owner registration	91
6.11 Admin create maid-servant profile.....	92
6.12 Maid-servant profile.....	93
6.13 Apartment owner list & add apartment owner	94
6.14 Admin view recent post.....	95
6.15 Admin view recent post details.....	95
Chapter 7.....	96
7.1 GitHub Link	97
7.2 Project Summary	97
7.3 Limitations	97
7.4 Obstacles and Achievements.....	97
7.5 Future Scope.....	98
7.6 References	98

List of Figures

Figure 1.1: Proposed system model.....	15
Figure 1.2: Gantt chart.....	16
Figure 3.1: Use Case diagram for Apartment owner	31
Figure 3.2: Use Case diagram for System Maintainers (Admin).....	32
Figure 3.3: Apartment Owner Login.....	41
Figure 3.4: Apartment Owner Post for Maid-Servant.....	42
Figure 3.5: Confirm maid-servant.....	43

Figure 3.6: See previous history.....	44
Figure 3.7: See recent post.....	45
Figure 3.8: View profile.....	46
Figure 3.9: Edit/update apartment owner’s profile information.....	47
Figure 3.10: Visit maid-servant profile.....	48
Figure 3.11: Admin login.....	49
Figure 3.12: Create apartment owner registration.....	50
Figure 3.13: Create maid-servant profile.....	51
Figure 3.14: Admin sees previous history.....	52
Figure 3.15: Admin sees recent post.....	53
Figure 3.16: Can Edit/Delete apartment owners & maid-servant Profile information.....	54
Figure 3.17: admin send some maid-servant to apartment owner.....	55
Figure 3.18: Sequence Diagram for Apartment owner Login.....	56
Figure 3.19: Sequence Diagram, Post for Maid-Servant.....	57
Figure 3.20: Sequence Diagram for Apartment owner confirms maid-servant.....	57
Figure 3.21: Sequence Diagram for Apartment owner sees his/her post History.....	58
Figure 3.22: Sequence Diagram for Apartment owner sees his/her recent post.....	58
Figure 3.23: Sequence Diagram for Apartment owner view his/her Profile.....	59
Figure 3.24: Sequence Diagram for Apartment owner edits their profile.....	59
Figure 3.25: Sequence Diagram for Apartment owner Visit maid-servant profile.....	60
Figure 3.26: Sequence Diagram for Admin Login.....	60
Figure 3.27: Sequence Diagram, registration for Apartment Owner.....	61
Figure 3.28: Sequence Diagram, registration for maid-servant.....	61
Figure 3.29: Sequence Diagram, Admin view Apartment Owner’s.....	62
Figure 3.30: Sequence Diagram, Admin view Apartment Owner’s recent post history.....	62

Figure 3.31: Sequence Diagram, Admin edits Apartment Owner profile.....	63
Figure 3.32: Sequence Diagram, Admin deletes Apartment Owner profile.....	63
Figure 3.33: Sequence Diagram for assigning maid-servant.....	64
Figure 4.1: Class Diagram.....	68
Figure 4.2: Database Diagram.....	69
Figure 6.1: Apartment owner login page.....	94
Figure 6.2: Home page of apartment owner.....	95
Figure 6.3: Apartment owner post for maid-servant.....	96
Figure 6.4: Apartment owner sees recent post.....	97
Figure 6.5: Apartment owner sees recent post details.....	98
Figure 6.6: Apartment owner sees previous history.....	99
Figure 6.7: Apartment owner confirm assigning the maid servant.....	100
Figure 6.8: Apartment owner view their profile.....	101
Figure 6.9: Admin login page.....	101
Figure 6.10: Apartment owner registration form.....	102
Figure 6.11: Admin create maid-servant profile.....	103
Figure 6.12: Maid-servant profile.....	104
Figure 6.13: Apartment owner list & add apartment owner.....	105
Figure 6.14: Admin view recent post.....	106
Figure 6.15: Admin view recent post details.....	

Chapter 1

Introduction

1.1 Project Overview

During the revolutionary people they cannot manage time to perform all their work. They spend a lot of time for traveling, eating, shopping, exercising, sports and other activities. Because of this, sometimes they cannot manage time to do their required work. Here the good parts of technology that can provide a good and easier lifestyle for people.

Technology is not just a source of entertainment and communication but it also plays an important role in resolving various issues of our life. Think about the people daily lifestyle, they usually have to do different types of work.

In the daily life of people they have to do some work by himself, and for the some task they have to depend on someone else such as search and recruit a Maid-Servant. Now, finding a maid-maid is not easy. Sometimes your requirements do not match the slaves and sometimes the slave-friendly needs do not match your needs. Also, Maid-Servant is not easily found. There is also a lot of time to find maid-Servant.

That's why I am 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. This process would be done within a few seconds.

1.2 Project Purpose

The main purpose of this project named “**Maid-Servant Assigning for a Colony Based Area**” 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 I am going to develop such a project.

1.2.1 Background

We 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, we 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

Our 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. Our 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 our system are:

- Apartment Owner
- Maid-Servant
- Maintainers(Admin)

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.

Maid-Servant: Maid-Servant to be assigned to apartment owner. When a maid-servant assigning request come to admin then admin select some maid-servant for sending to apartment owner.

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.

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.

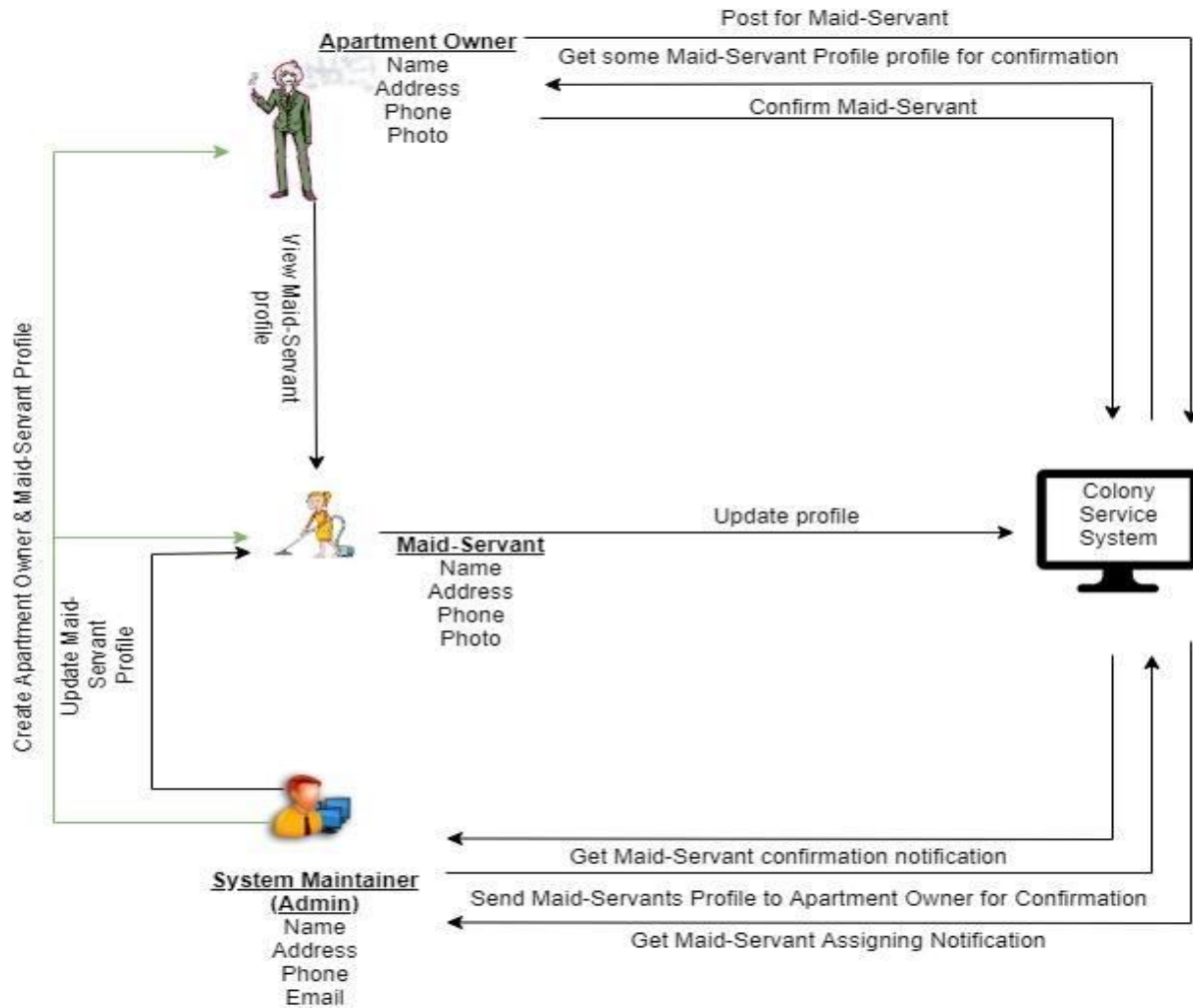


Figure 1.1: Proposed system model

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
Planning	Ideas	█													
	Problem definition	█													
	Proposal planning		█												
Requirements	Requirement specification			█	█	█									
	Requirement analysis					█	█								
QA-1	Quality assurance						█	█							
System design	Sketching							█	█	█					
	Design specification							█	█	█					
	Database design								█	█					
Implementation	Post for maid-servant									█	█	█			
QA-2	Test cases										█	█			
Testing	Unit testing									█	█				
	Black box testing										█	█	█	█	█
Delivery	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 month & First-half(Third month)	1.5
Requirement analysis	Third month	1
Sketching	Fourth month & First-half(Fifth month)	1.5
Design specification	Fourth month & First-half(Fifth month)	1.5
Database design	Second-half (Fourth month) & First-half(Fifth month)	1
Post for maid-servant	Fifth month & First-half(Sixth month)	1.5
Quality assurance	Second-half (Third month) & First-half(Fourth month)	1
Test case	Second-half (Fifth month) & First-half(Sixth month)	1
Unit testing	Fifth month	1
Black-box testing	Second-half (Fifth month), Sixth month & First-half (Seventh 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

Priority Chart

Important\Urgent	Yes	No
Yes	High priority	Medium priority
No	Low priority	Ignore

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

FR16	Yes	Yes	High
-------------	------------	------------	-------------

2.1.1 Login for Apartment owner

FR01	Login for Apartment owner		
Description	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.		
Stakeholders	Apartment owner	Priority	High

2.1.2 Apartment owner post for maid-servant

FR02	Post for maid-servant		
Description	After successfully login, apartment owner can post for looking Maid-Servant. Apartment Owner must be fill up require input field with his requirements, then submit the post.		
Stakeholders	Apartment owner	Priority	High

2.1.3 Apartment owner confirm assigning the maid-servant

FR03	Confirm assigning the maid-servant		
Description	After submit the maid-servant assigning request, the request goes to the admin and admin send some maid-servant list to apartment owner and finally apartment owner choose and confirm a maid-servant.		
Stakeholders	Apartment owner, Maid-servant	Priority	High

2.1.4 Apartment owner view previous history

FR04	Apartment owner view previous history		
Description	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.		
Stakeholders	Apartment owner	Priority	Medium

2.1.5 Apartment owner can view recent post

FR05	Apartment owner view recent post		
Description	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.		
Stakeholders	Apartment owner	Priority	High

2.1.6 Apartment owner can view their profile

FR06	Apartment owner can view their own profile		
Description	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		
Stakeholders	Apartment owner	Priority	High

2.1.7 Apartment owner can edit their profile

FR07	Apartment owner edit their profile		
Description	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.		
Stakeholders	Apartment owner	Priority	Medium

2.1.8 Apartment owner can view maid-servant profile

FR08	Apartment owner can view maid-servant profile		
Description	Apartment Owner can see only the Maid-Servant Profile who is assigned for his work. After assigning the Maid-Servant for apartment owner, The apartment owner only can see the Maid-Servant profile who is assigned for his work.		
Stakeholders	Apartment owner, Maid-servant	Priority	High

2.1.9 Login for Admin

FR09	Login for Admin		
Description	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.		
Stakeholders	Admin	Priority	High

2.1.10 Registration apartment owner

FR10	Registration apartment owner		
Description	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.		
Stakeholders	Admin	Priority	High

2.1.11 Admin Create Maid-servant profile

FR11	Create Maid-servant profile		
Description	Admin can create the Maid-servant profile with Valid require information like Name, Phone number, Address etc.		
Stakeholders	Admin	Priority	High

2.1.12 Admin View History

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

2.1.13 Admin View Recent Post

FR13	Admin View Recent Post		
Description	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.		
Stakeholders	Admin	Priority	High

2.1.14 Admin Delete Apartment Owner and Maid-Servant profile

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

2.1.15 Admin Edit Apartment Owner and Maid-Servant Profile

FR15	Edit Profile		
Description	Admin also can able to edit the apartment owner and Maid-Servant profile		
Stakeholders	Admin	Priority	Medium

2.1.16 Send some maid-Servant profile

FR16	Confirm Maid-Servant		
Description	Admin getting Maid-Servant assigning request notification from apartment owner, then admin check the schedule and apartment owner requirements and other info and send two or three Maid-Servant profile to apartment owner for final confirmation. Apartment owner see those profile and confirm appropriate Maid-Servant.		
Stakeholders	Apartment owner, Admin	Priority	High

2.2 Data Requirements

Data requirements prescribed scope and level of detail required to achieve the model objectives. Gathering system data is to determine what data is required for building the model. 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:

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

2.3 Performance Requirements

Incorrectly defined performance specifications can lead to disputes between clients and suppliers.

2.3.1 Speed and Latency Requirements

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

RT1	View and post result should be show less than 3 second		
Description	When the Apartment Owner or Admin want to see any information then the view result must be show less than 3 second.		
Stakeholders	Apartment Owner, Admin	Priority	Medium

2.3.2 Precision or Accuracy Requirements

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

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

2.3.3 Capacity Requirements

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

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

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

2.4.2 Availability Requirements

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

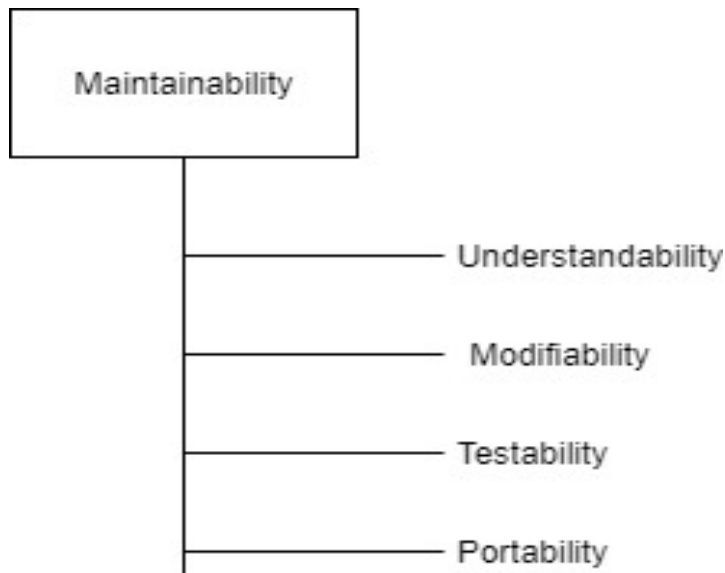
RFT1	Ensure that system can handle all user access or any external tolerance.		
Description	The system can cope with error during execution. Also ensure that the system continue properly even of the failure of some of its components		
Stakeholders	Apartment Owner, Admin	Priority	High

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 base 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.

ARI	In optional field the word "optional" must appear into the field		
Description	If there is no optional word appeared into the field that means this field is mandatory for input.		
Stakeholders	Apartment Owner, Admin	Priority	High

2.8.2 Style Requirements

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

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

I have use case diagram. And there are two actors. Each actor plays a different role. And those who are already directed to draw in this use case. This diagram will clarify our system in brief.

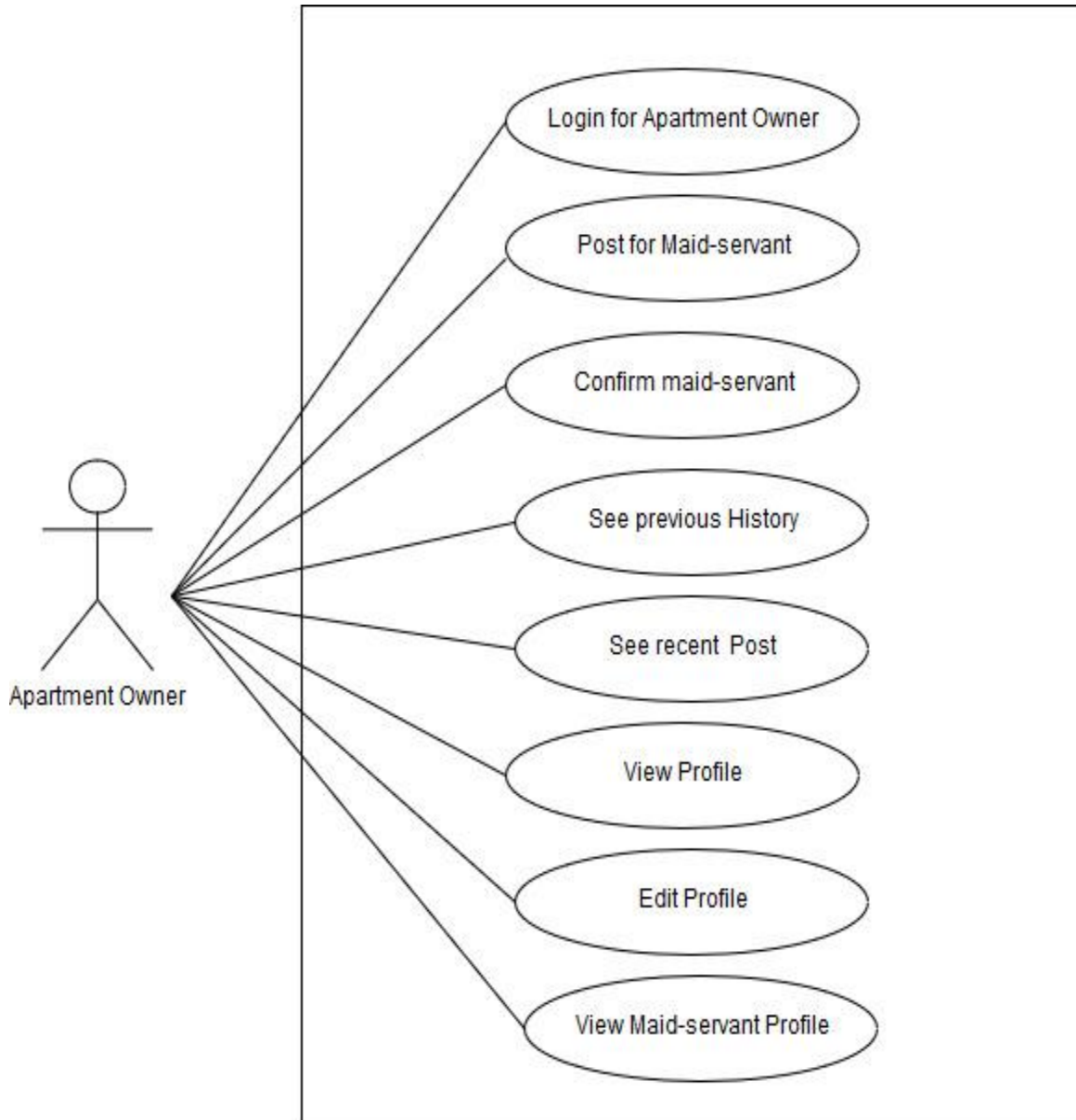


Figure 3.1: Use Case diagram for Apartment owner

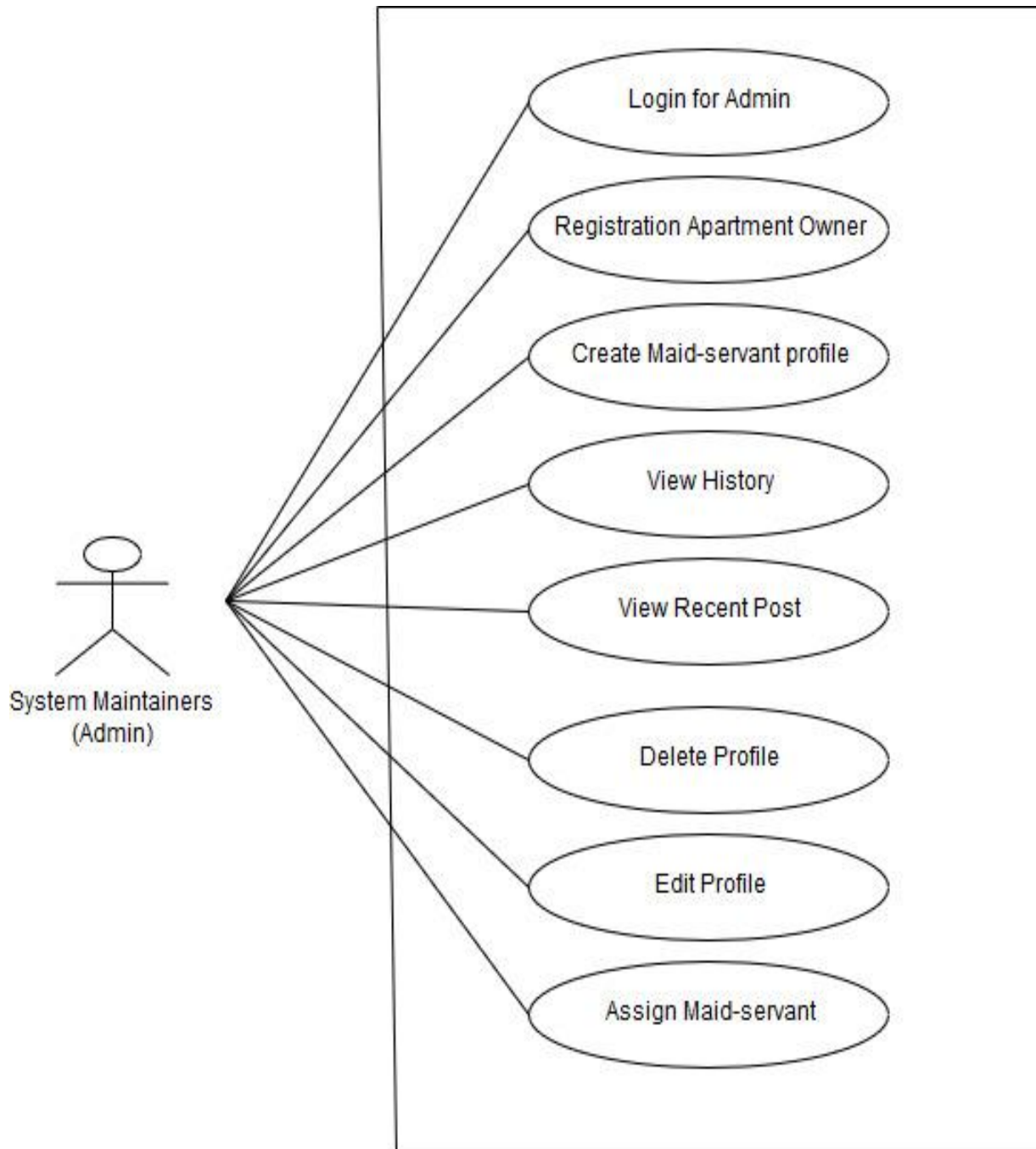


Figure 3.2: Use Case diagram for System Maintainers (Admin)

Use Case Description

3.1.1 Login for apartment owner

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

3.1.2 Apartment owner post for maid-servant

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

3.1.3 Apartment Owner confirm assigning the maid servant

Use Case	Apartment Owner confirm assigning the maid servant	
Goal	Apartment owner choose and confirm a maid-servant	
Preconditions	<ul style="list-style-type: none"> User type must be Apartment Owner Apartment Owner have a maid-servant assigning request post 	
Success End Condition	A maid-servant is assigned against apartment owner post	
Failed End Condition	A maid-servant did not assign against apartment owner post	
Primary Actors: Secondary Actors:	Apartment Owner Maid-Servant	
Trigger	A maid-servant is assigned	
Description / Main Success Scenario	Step	Action
	1	After getting maid-servant list, apartment owner view those maid-servant profile.
	2	Apartment owner choose and confirm a maid-servant
Alternative Flows	Step	Branching Action
	2a	A maid-servant did not assign
Quality Requirements	Step	Requirements
	1	N/A

3.1.4 Apartment Owner sees previous history

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

3.1.5 Apartment Owner sees recent post

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

3.1.6 Apartment Owner can view profile

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

3.1.7 Apartment Owner can edit profile

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

3.1.8 Apartment Owner can visit maid-servant profile

Use Case	Apartment Owner can visit maid-servant profile	
Goal	Apartment Owner can see the maid-servant profile	
Preconditions	<ul style="list-style-type: none"> • User type must be Apartment Owner • Apartment Owner have a post for maid-servant 	
Success End Condition	Apartment owner can see the maid-servant profile	
Failed End Condition	Apartment owner can't see the maid-servant profile	
Primary Actors: Secondary Actors:	Apartment Owner Maid-Servant	
Trigger	Maid-Servant profile appear to apartment owner	
Description / Main Success Scenario	Step	Action
	1	Maid-Servant data comes from their profile
	2	Maid-Servant profile appear to apartment owner
Alternative Flows	Step	Branching Action
	2a	Apartment can't able to see the maid-servant profile
Quality Requirements	Step	Requirements
	1	N/A

3.1.9 Login for Admin

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

3.1.10 Apartment Owner registration

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

3.1.11 Admin create maid-servant profile

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

3.1.12 Admin can sees previous history

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

3.1.13 Admin can sees recent post

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

3.1.14 Admin can delete profile

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

3.1.15 Admin Can Edit Profile

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

3.1.16 Admin send assign maid-servant list

Use Case	Admin send assign maid-servant list	
Goal	After getting the maid-servant assigning request, admin send some maid-servant profile to apartment owner	
Preconditions	<ul style="list-style-type: none"> User type must be Admin Have a maid-servant assigning request by apartment owner 	
Success End Condition	Apartment owner getting some maidservant profile list	
Failed End Condition	Apartment owner is not getting any maid-servant profile list	
Primary Actors: Secondary Actors:	Admin Apartment Owner, Maid-Servant	
Trigger	Some maid-servant profile is appear to apartment owner	
Description / Main Success Scenario	Step	Action
	1	Firstly apartment owner post for maid-servant, then admin get this notification
	2	Then admin send some maid-servant profile list to apartment owner
Alternative Flows	Step	Branching Action
	2a	Apartment owner is not getting any maid-servant profile list
Quality Requirements	Step	Requirements
	1	N/A

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

Firstly apartment owner need to verify themselves by login into the system. Apartment owner login the system using the email and password that provided by admin.

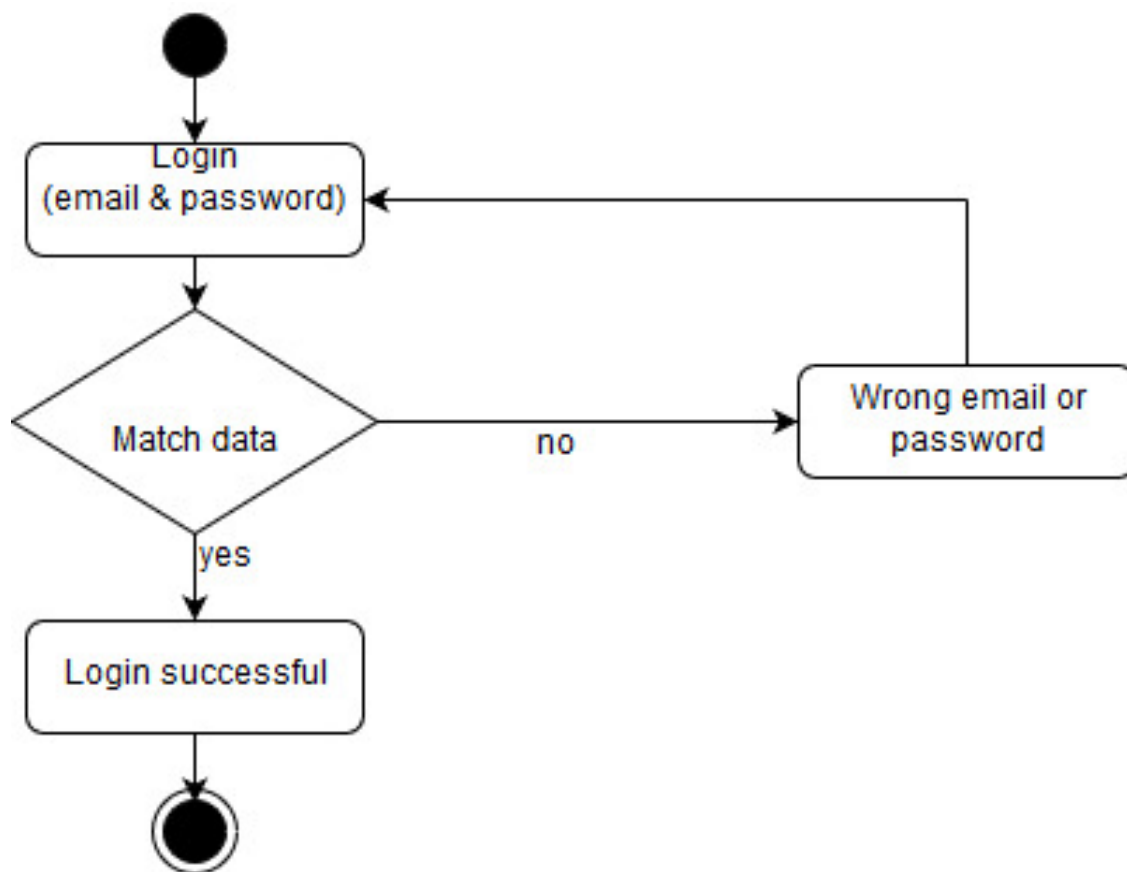


Figure 3.3: Apartment Owner Login

3.2.2 Apartment owner post for maid-servant

After login successfully, the apartment owner enters into the system and can able to post for Maid-Servant. For assign, a maid-servant apartment owner must be Fill-up some require field, then submit the post.

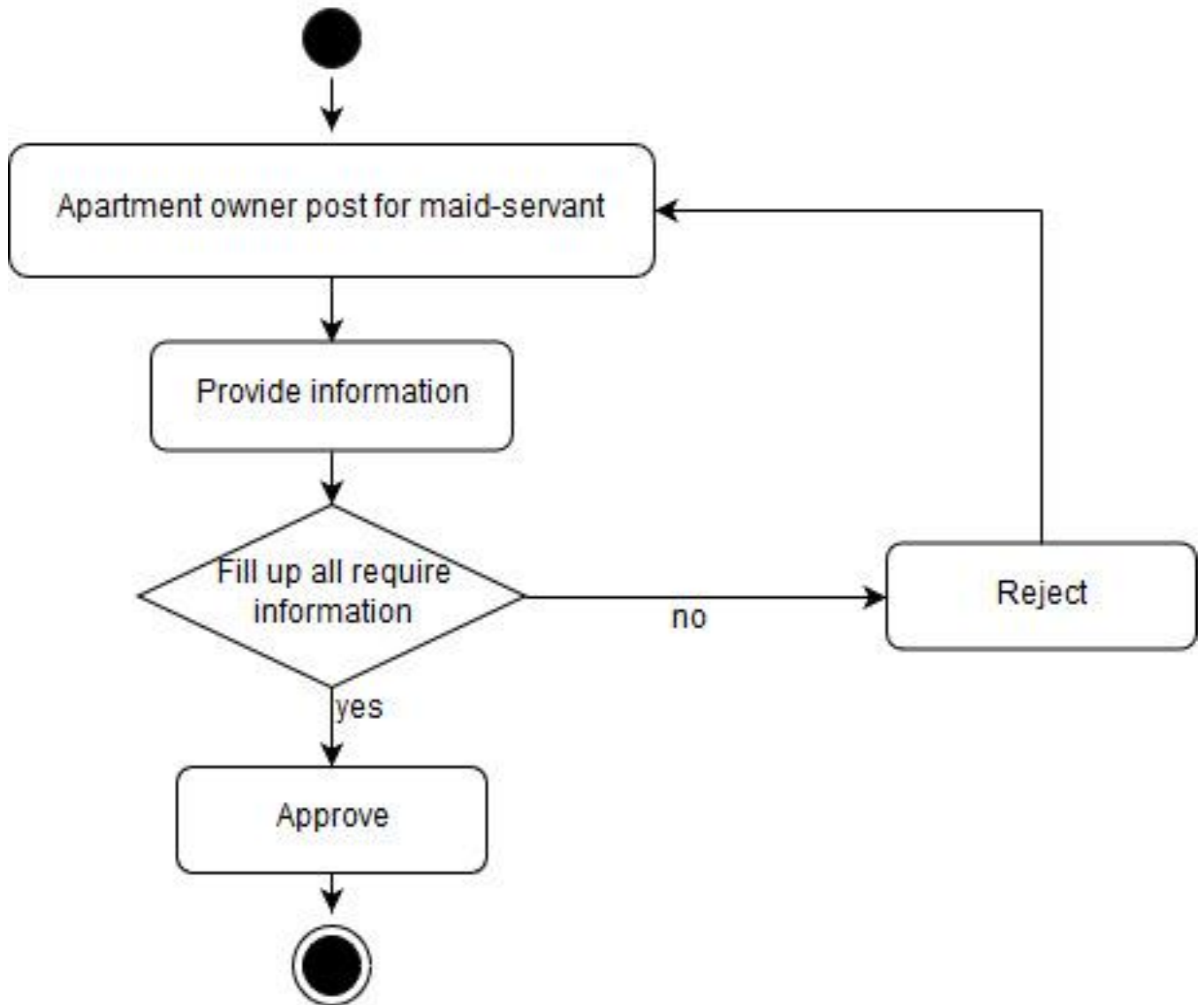


Figure 3.4: Apartment Owner Post for Maid-Servant

3.2.3 Apartment Owner confirm maid-servant

After submit the maid-servant assigning request, apartment owner get back some maid-servant list and choose the maid-servant from the list for final confirmation.

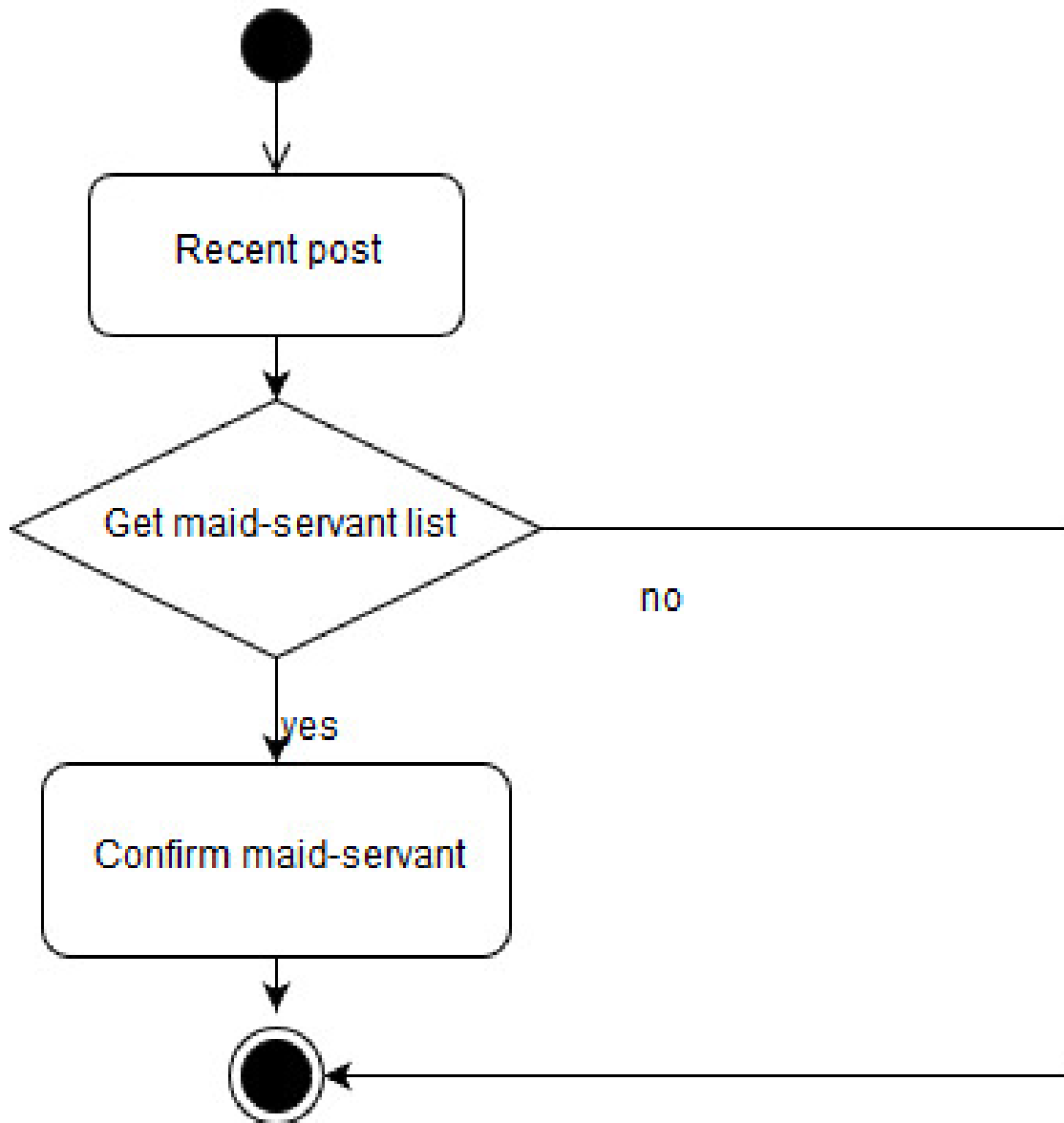


Figure 3.5: Confirm maid-servant

3. 2.4Apartment 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.

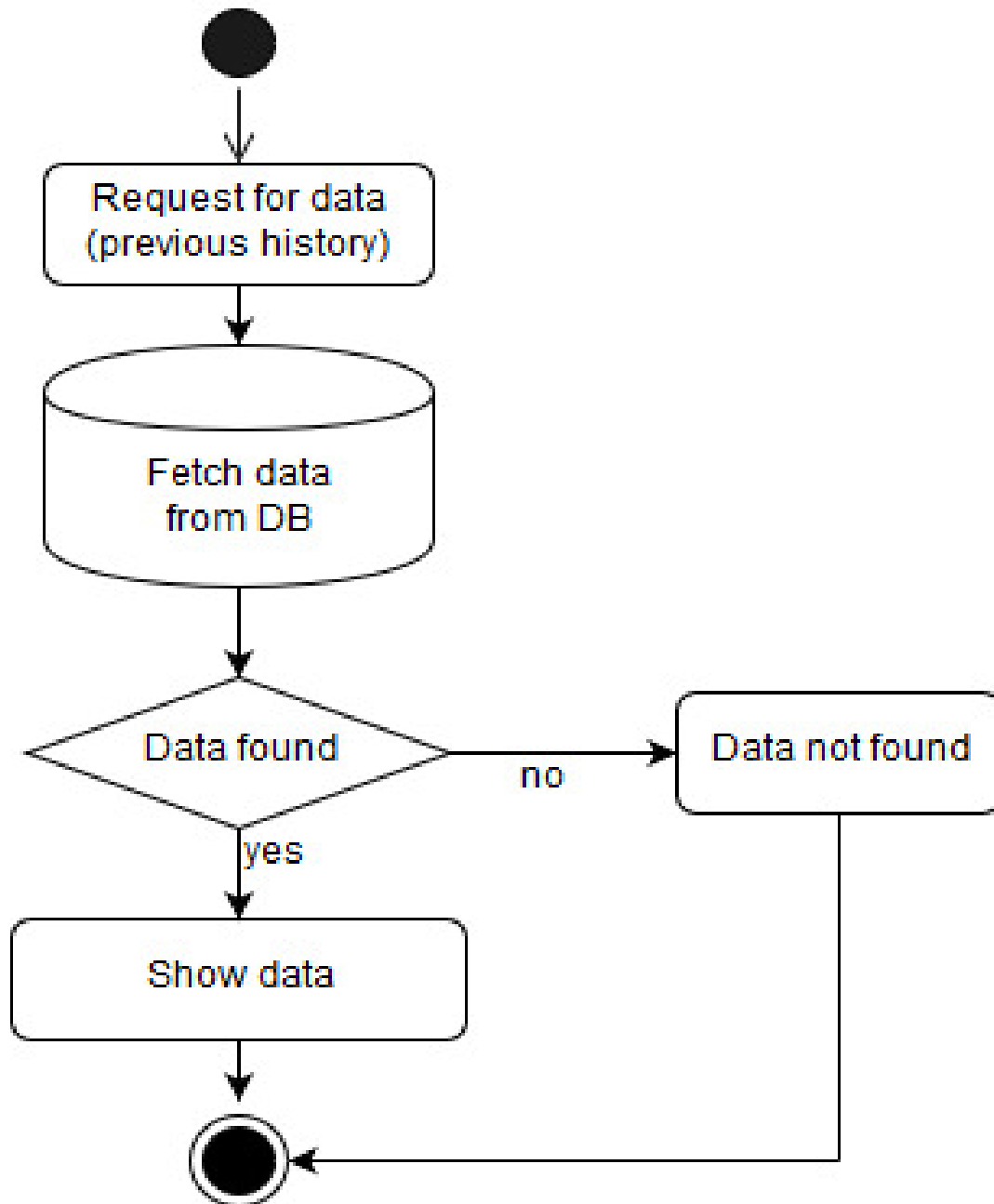


Figure 3.6: See previous history

3.2.5 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 .

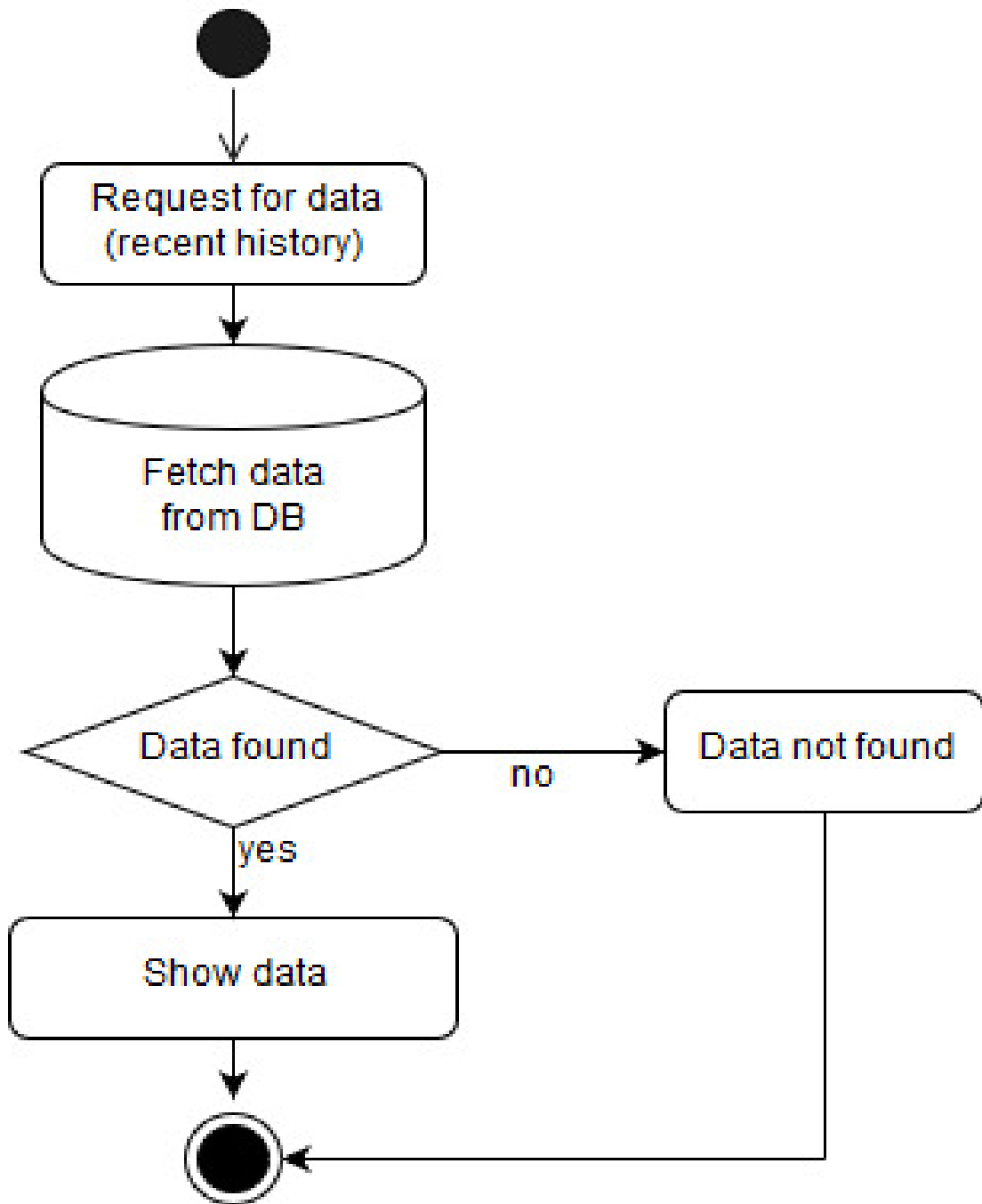


Figure 3.7: See recent post

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

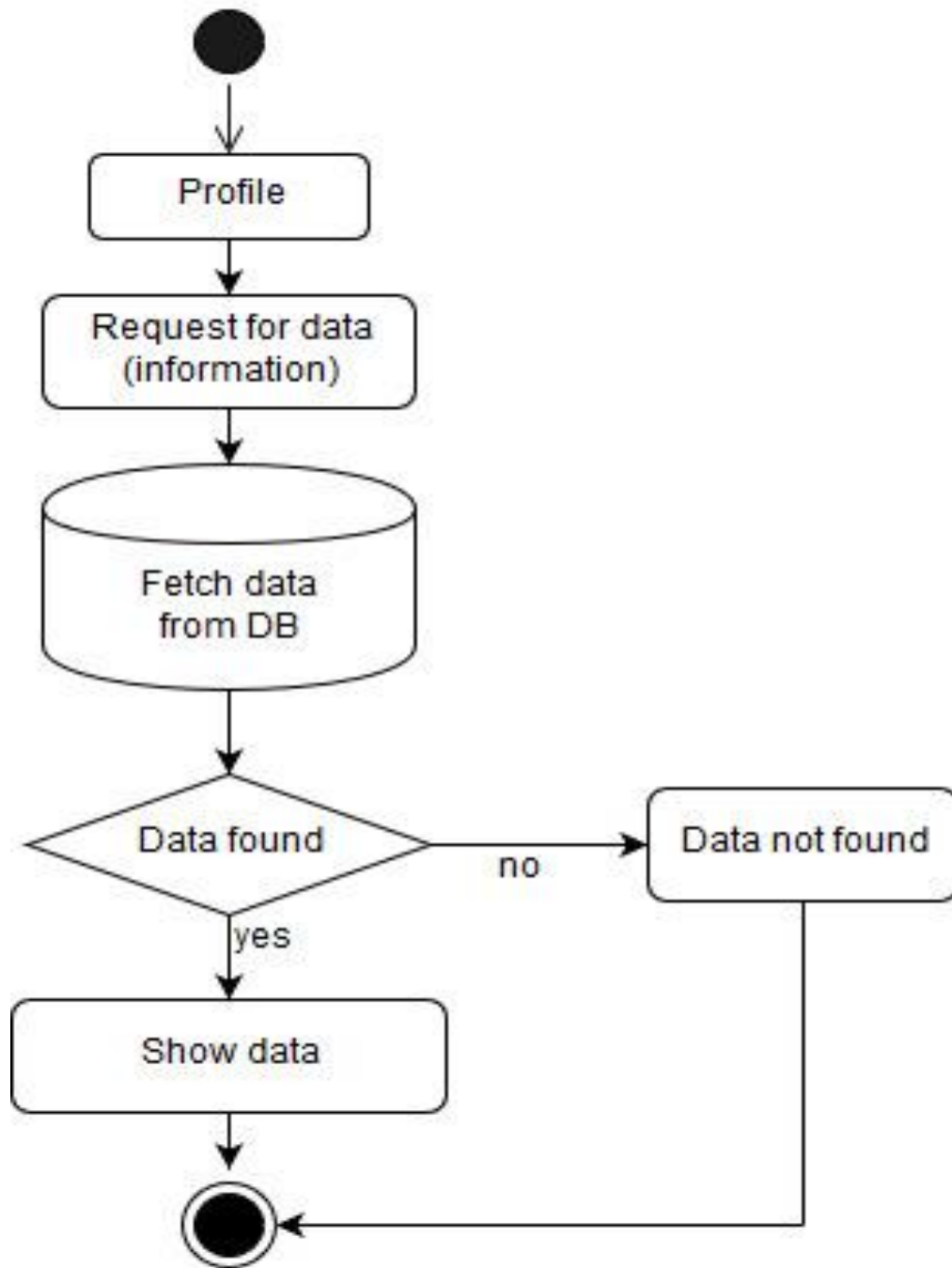


Figure 3.8: View profile

3.2.7 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

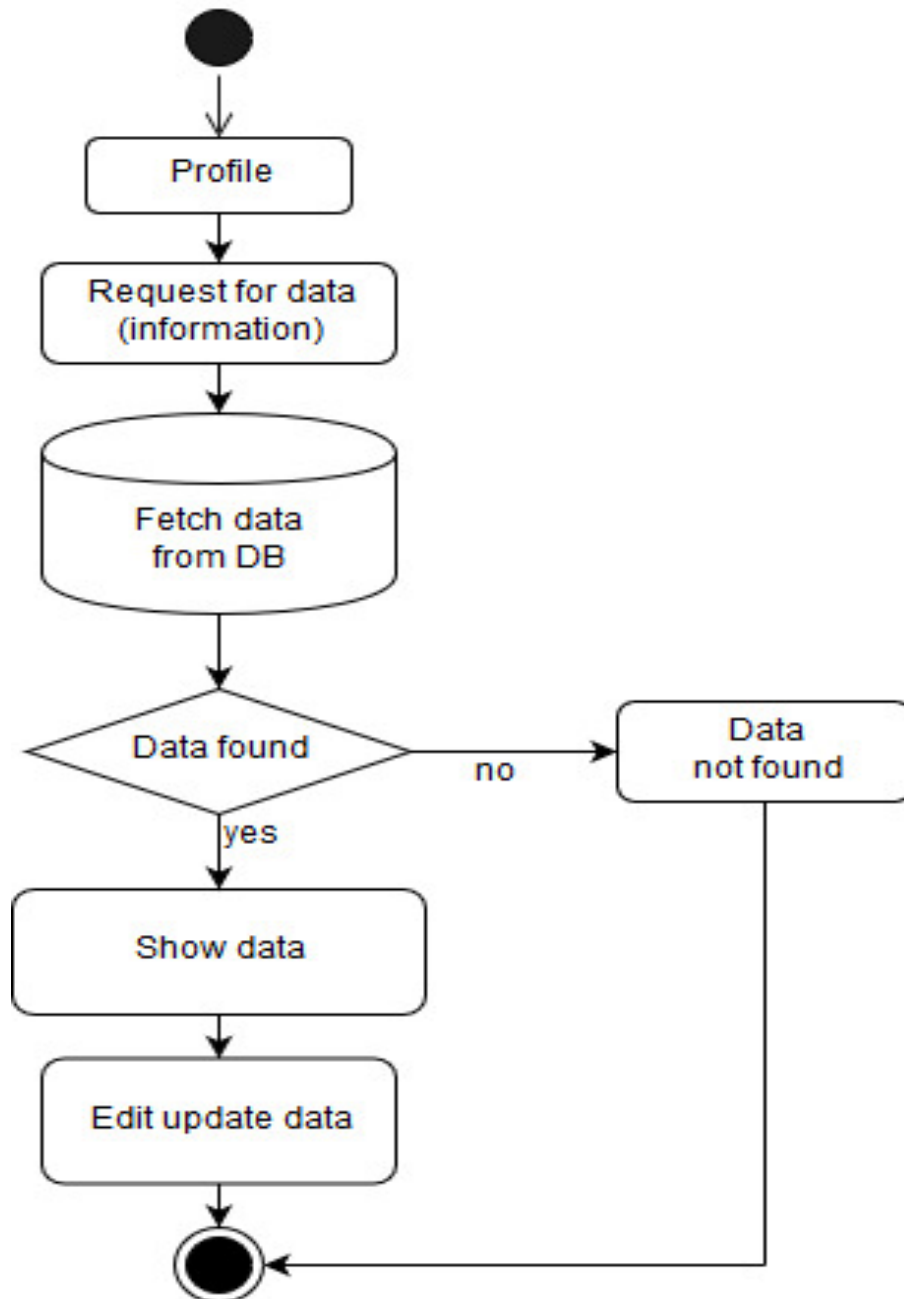


Figure 3.9: Edit/update apartment owner's profile information

3.2.8 Apartment Owner Can Visit Maid-Servant Profile

When a maid-servant is assigned to any apartment owner then the apartment owner can view that maid-servant profile.

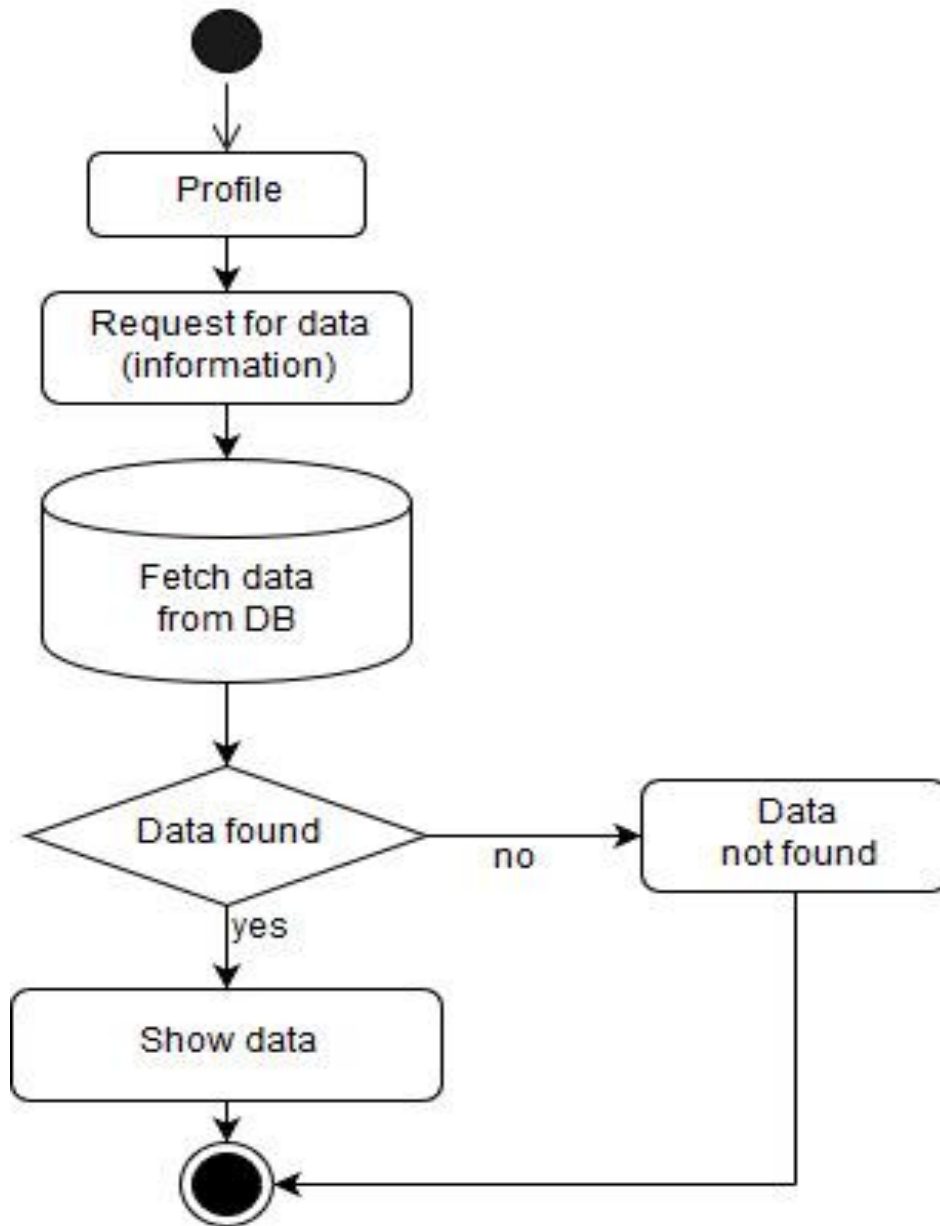


Figure 3.10: Visit maid-servant profile

3.2.9 Login for Admin

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

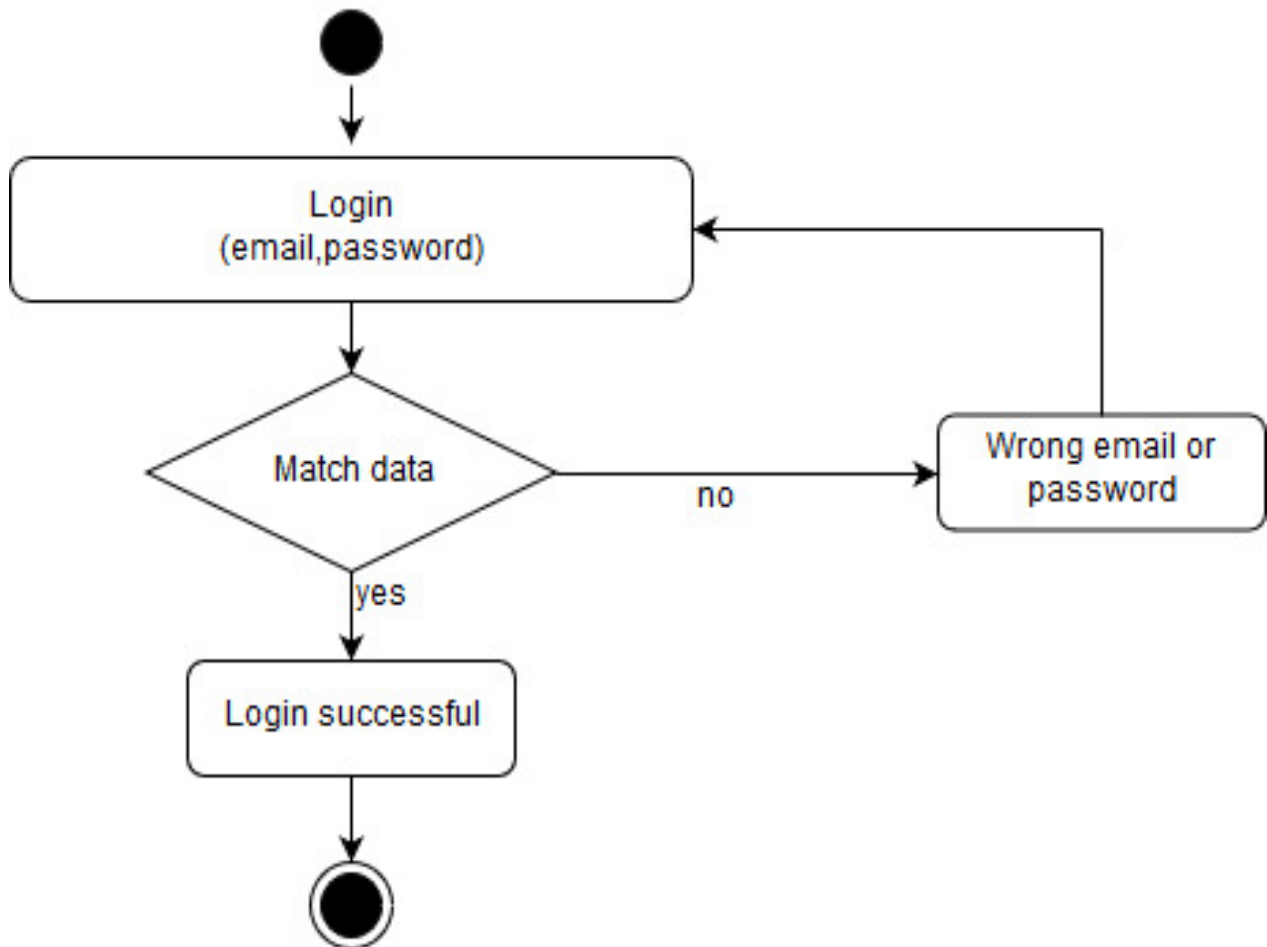


Figure 3.11: Admin login

3.2.10 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.

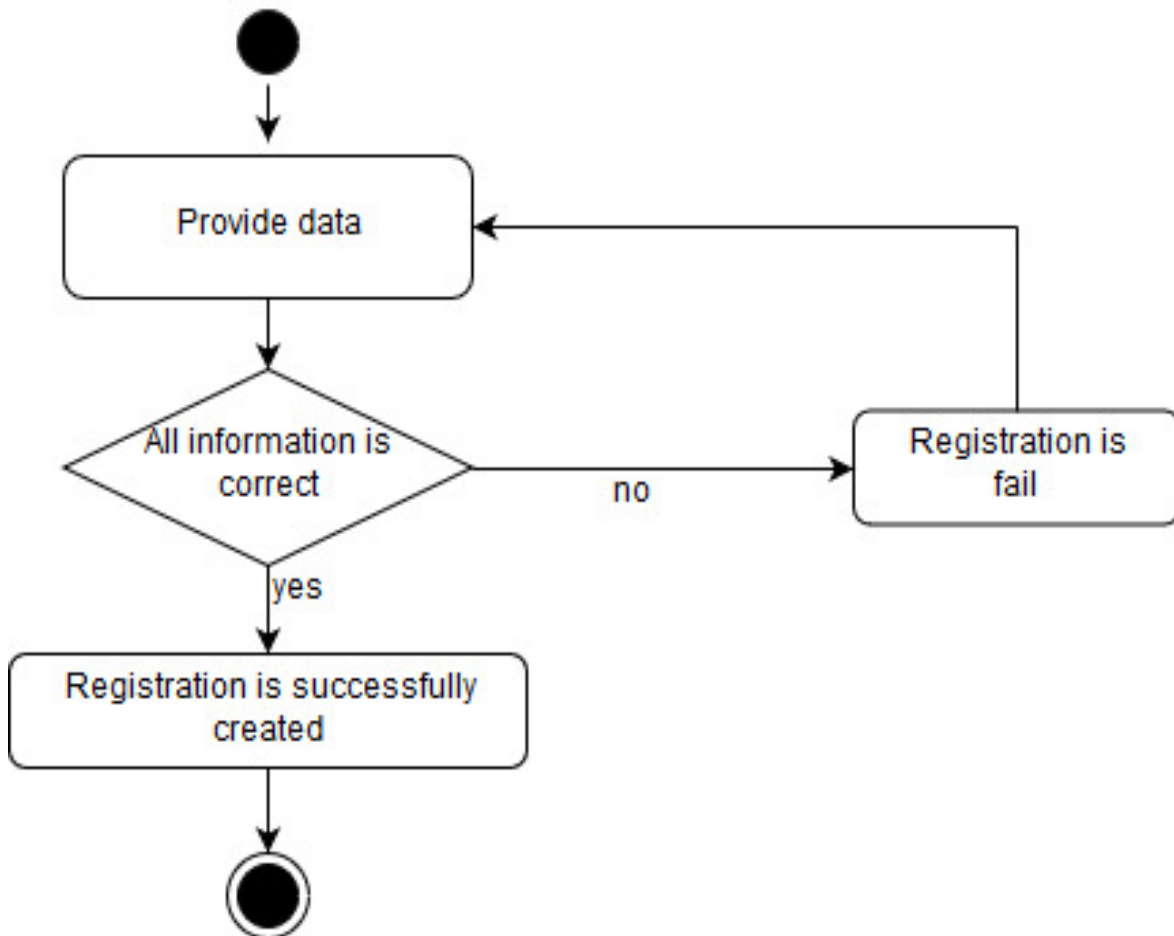


Figure 3.12: Create apartment owner registration

3.2.11 Admin Create maid-servant profile

Admin also created maid-servant profile with the information such as Name, Address, Phone, age, etc.

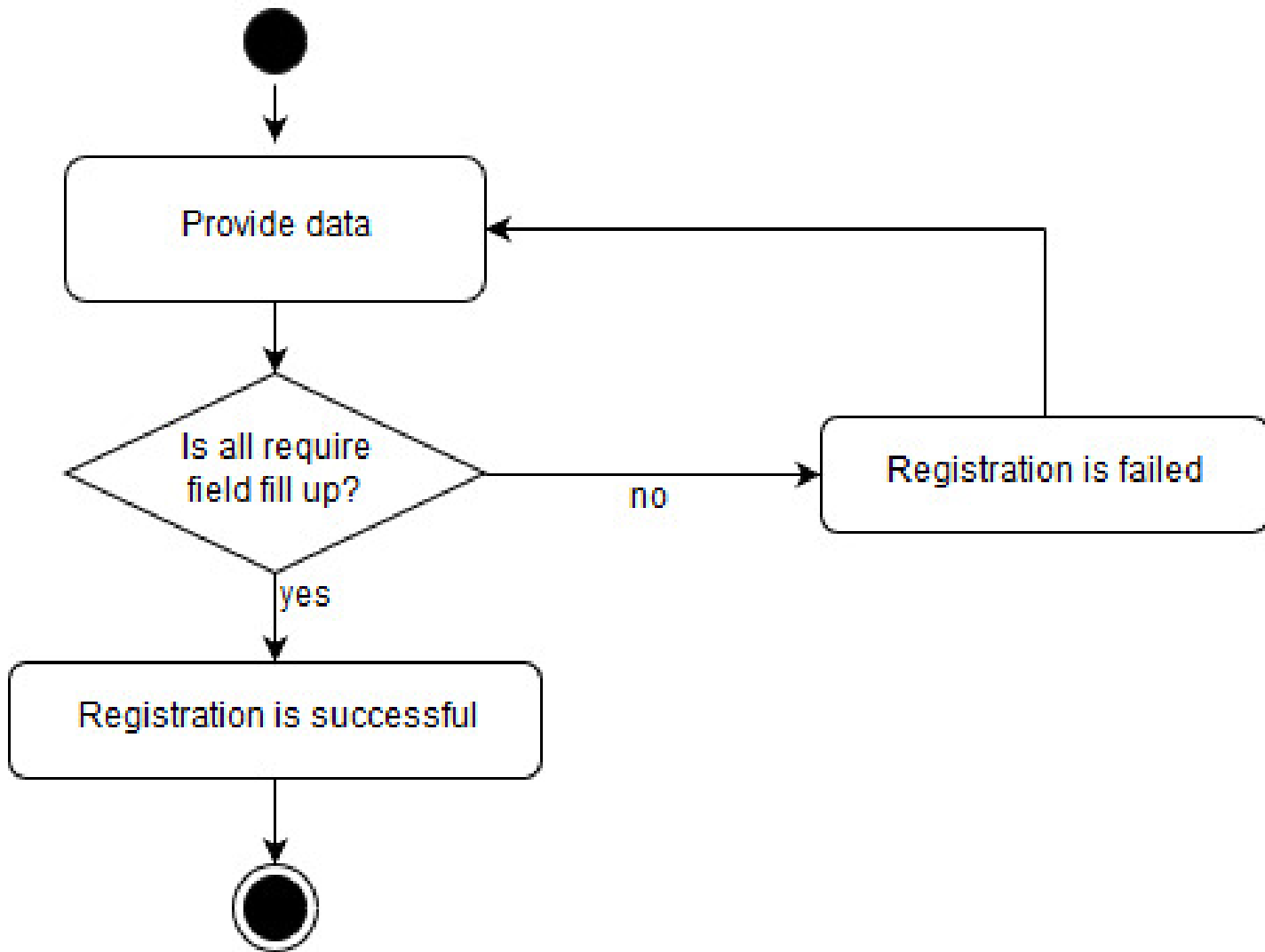


Figure 3.13: Create maid-servant profile

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.

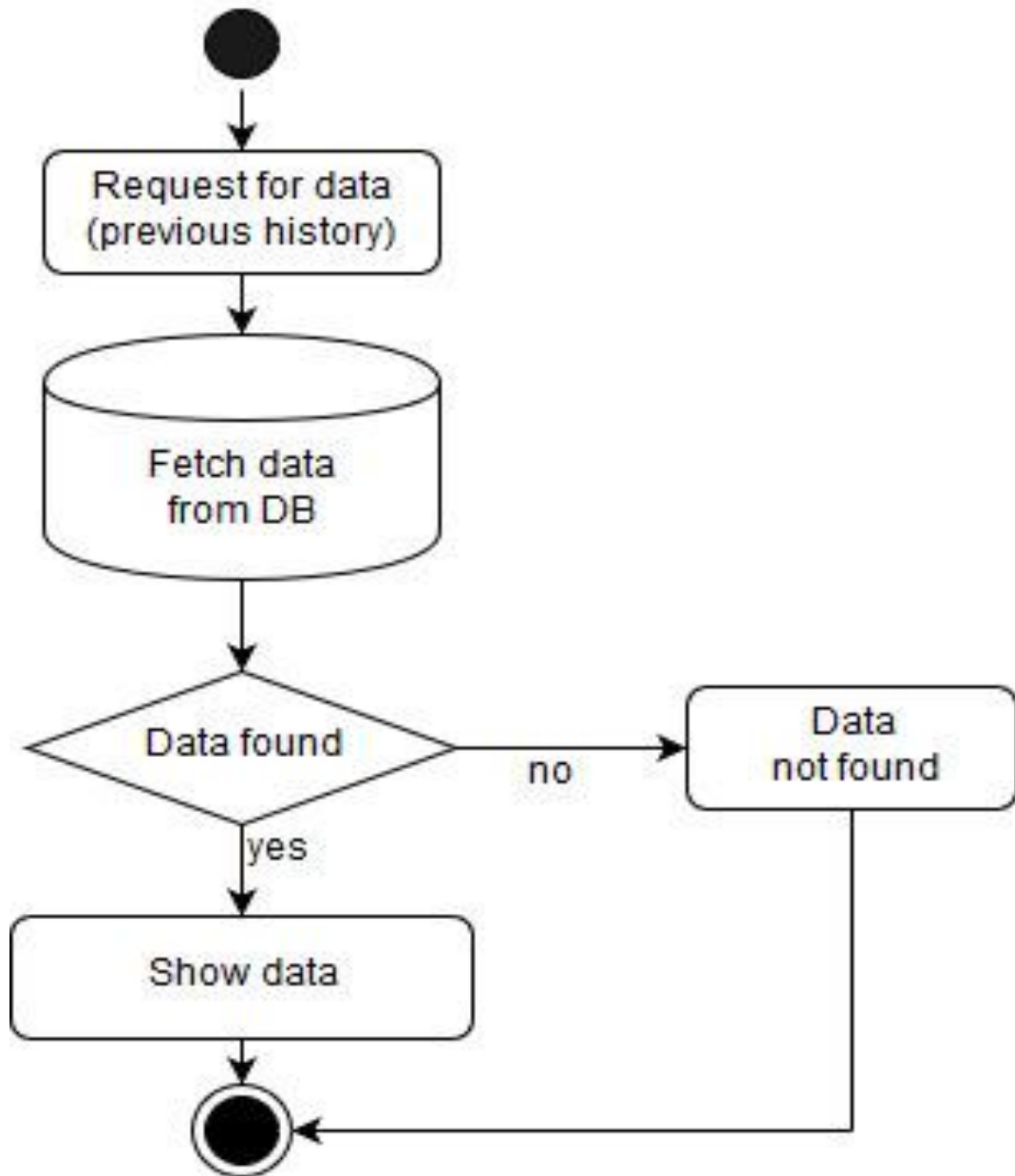


Figure 3.14: Admin sees previous history

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.

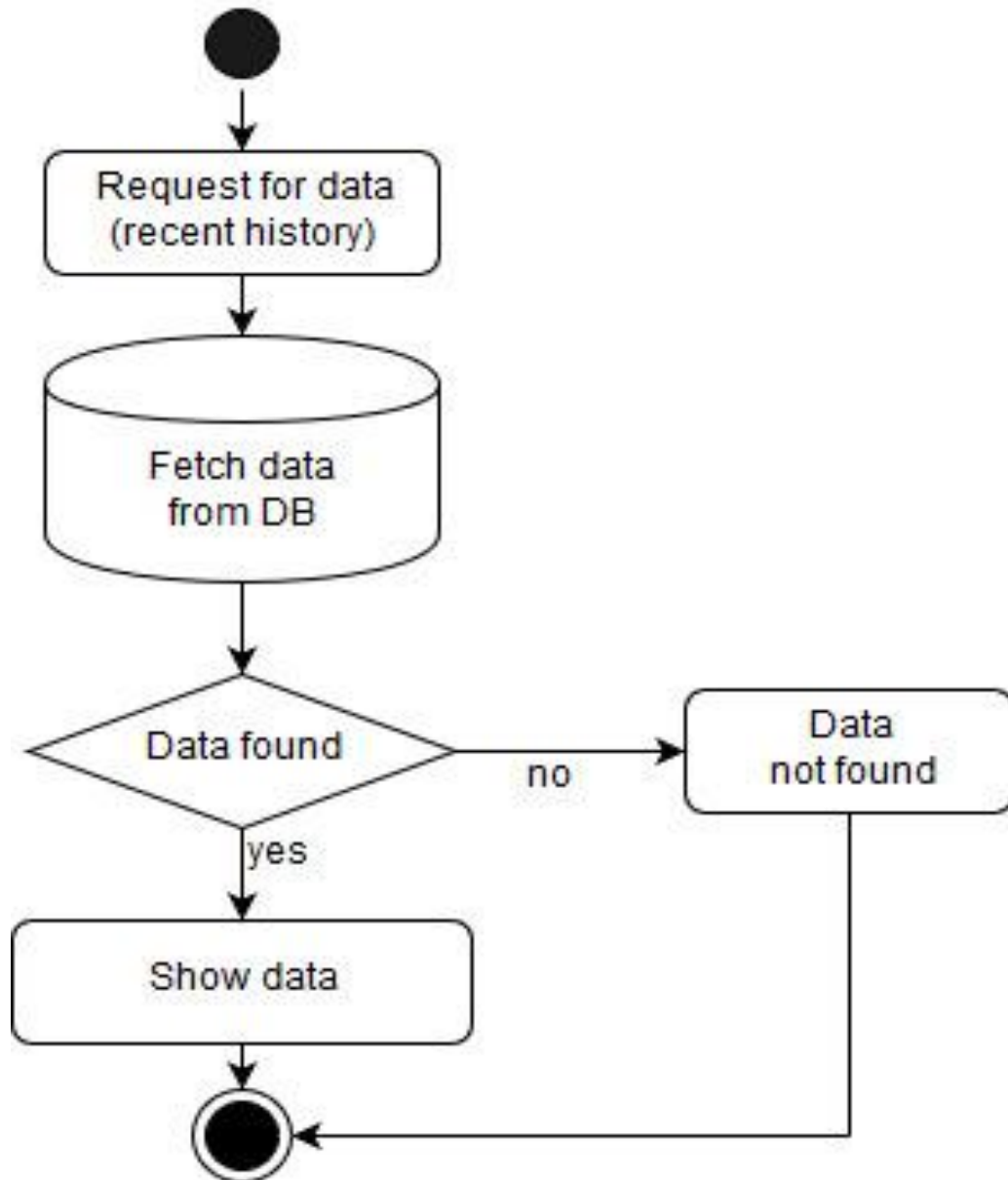


Figure 3.15: Admin sees recent post

3.2.14 Admin can edit and delete apartment owners & maid-servant 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.

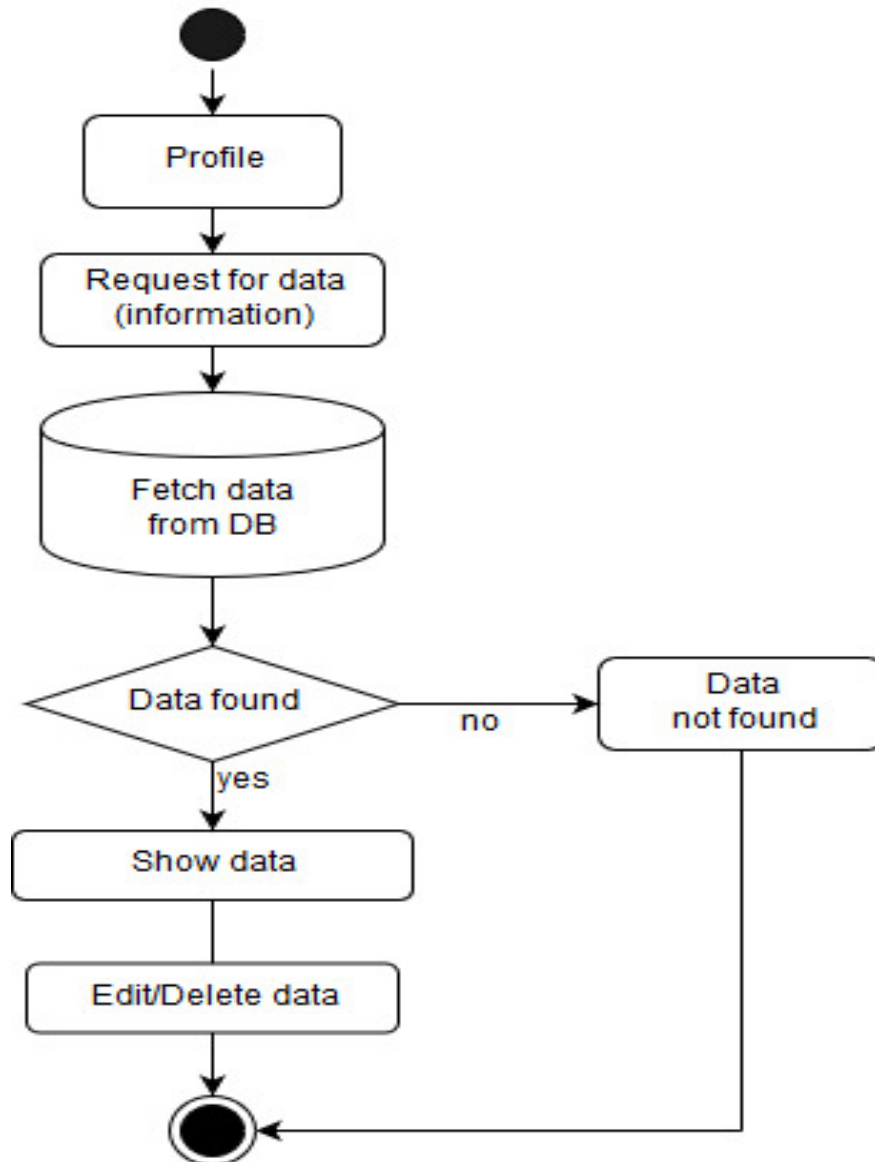


Figure 3.16: Can Edit/Delete apartment owners & maid-servant Profile information

3.2.15 Admin send maid-servant list to apartment owner

After seeing the recent post if there have any maid-servant assigning request by apartment owner, then admin send some maid-servant profile list to apartment owner for final confirmation.

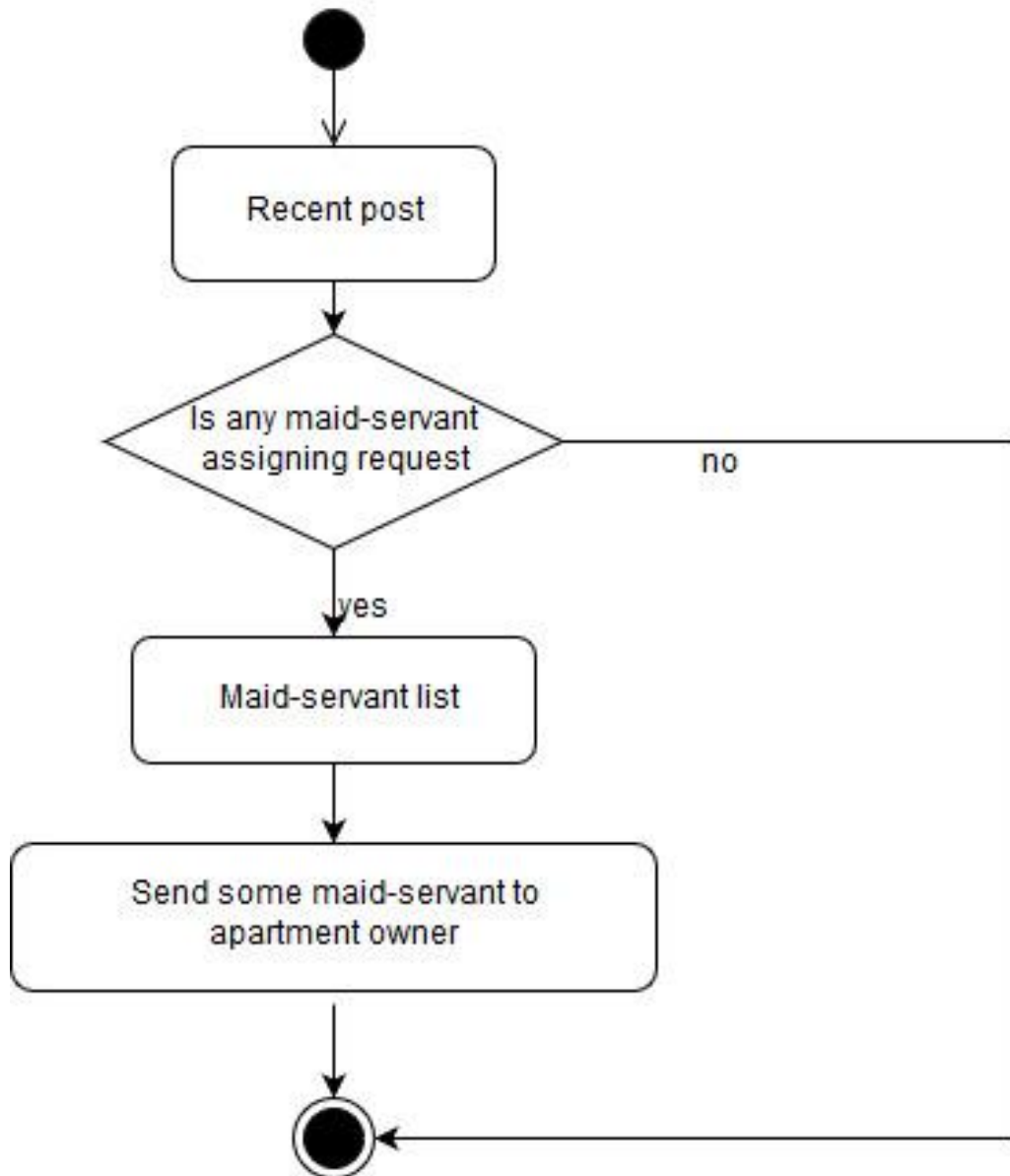


Figure 3.17: admin send some maid-servant to apartment owner

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.

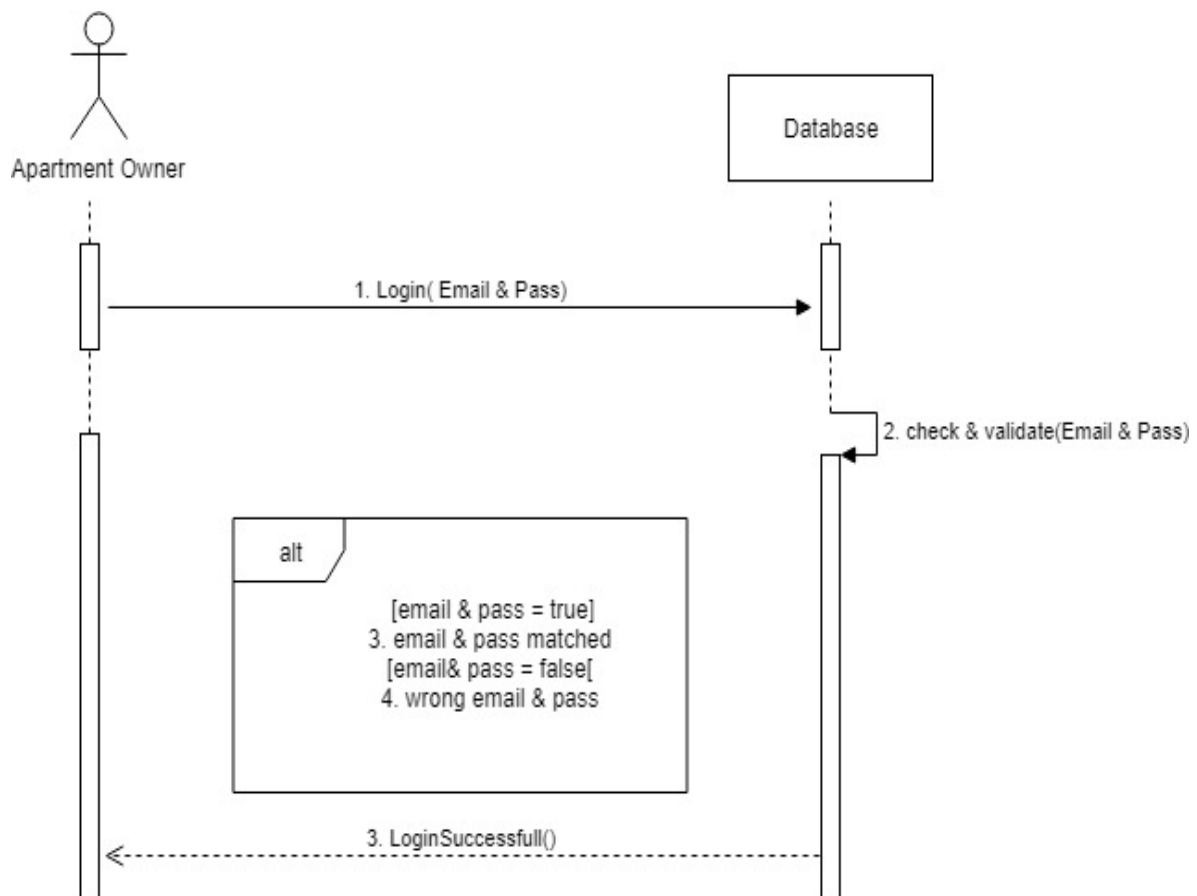


Figure 3.18: Sequence Diagram for Apartment owner Login

3.3.2 Apartment owner post for maid-servant

Apartment owner enter into the system and can post for maid-servant

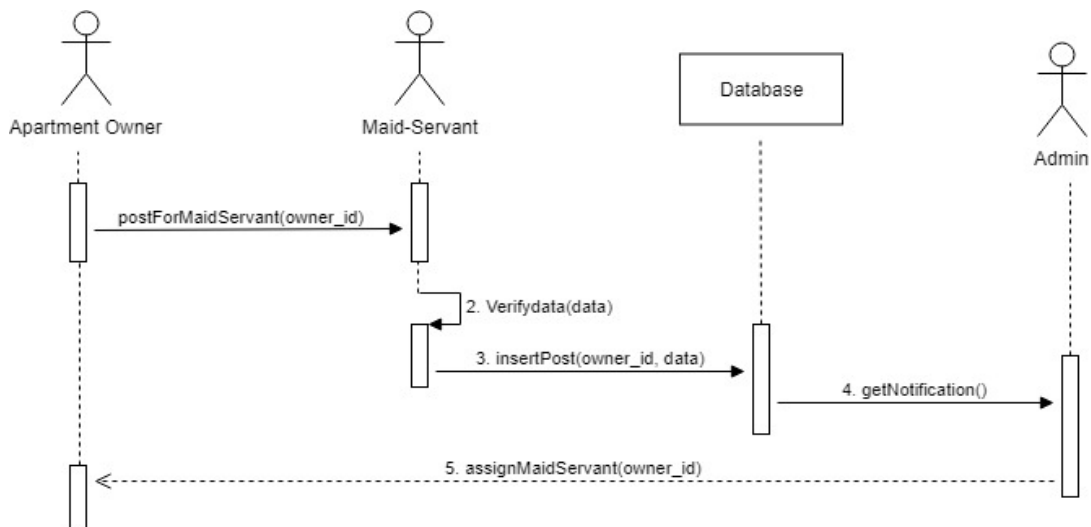


Figure 3.19: Sequence Diagram, Post for Maid-Servant

3.3.3 Apartment Owner confirm a maid-servant

After submitting the request for appointment of the maid-servant, the owner sends some of the list of maid-servant to the apartment owner and he chooses a maid-servant from the list for final confirmation.

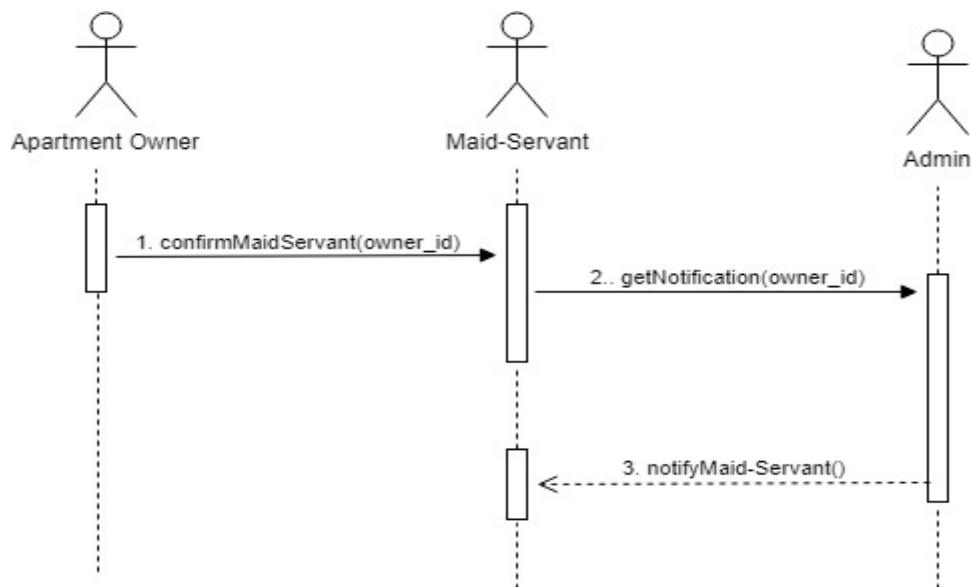


Figure 3.20: Sequence Diagram for Apartment owner confirm maid-servant

3.3.4 Apartment Owner sees his previous history

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

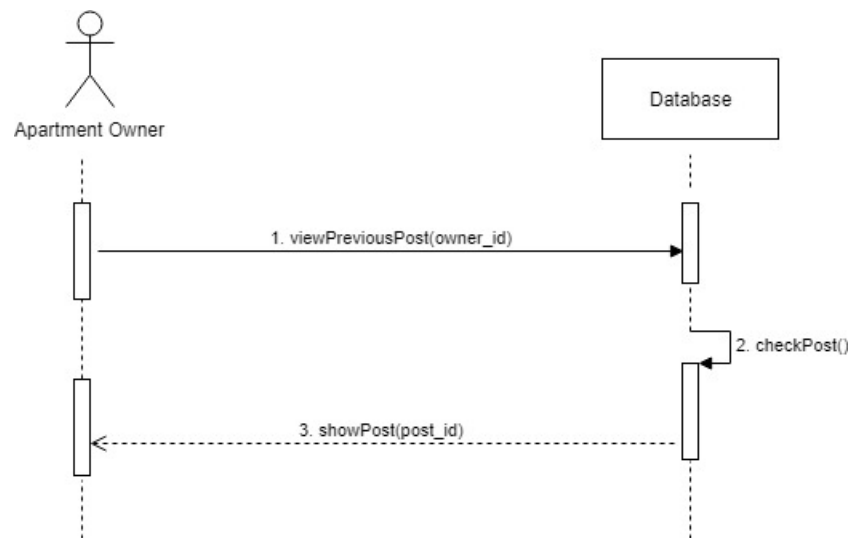


Figure 3.21: Sequence Diagram for Apartment owner sees his/her post History

3.3.5 Apartment Owner sees their recent post

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

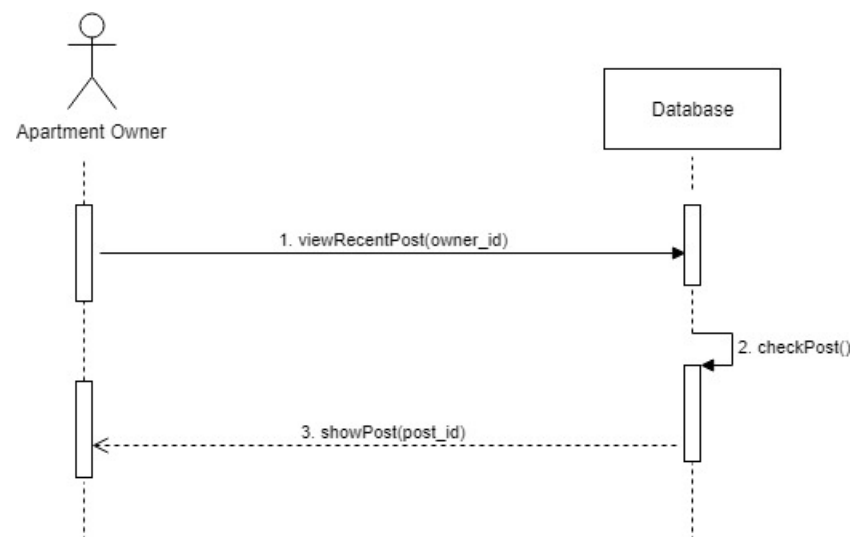


Figure 3.22: Sequence Diagram for Apartment owner sees his/her recent post

3.3.6 Apartment Owner Can View Profile

Apartment owner can able to view their own profile

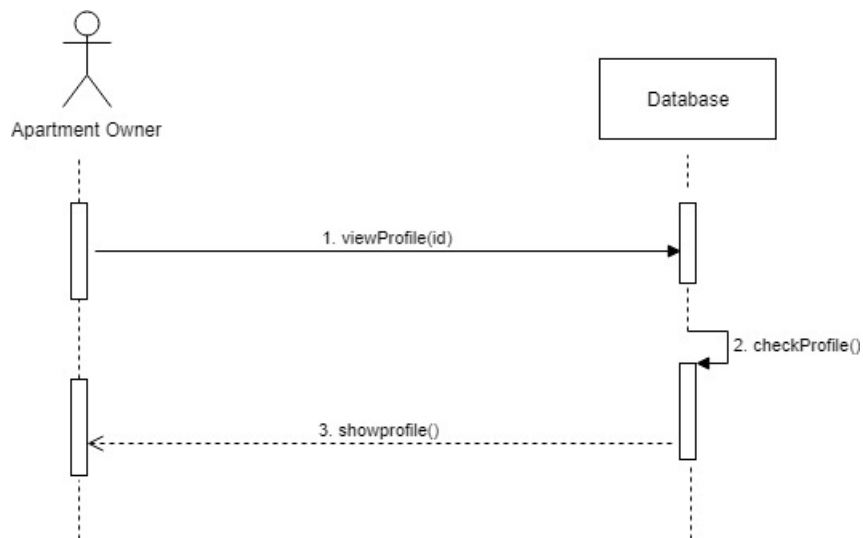


Figure 3.23: Sequence Diagram for Apartment owner view his/her Profile

3.3.7 Apartment Owner edits their profile

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

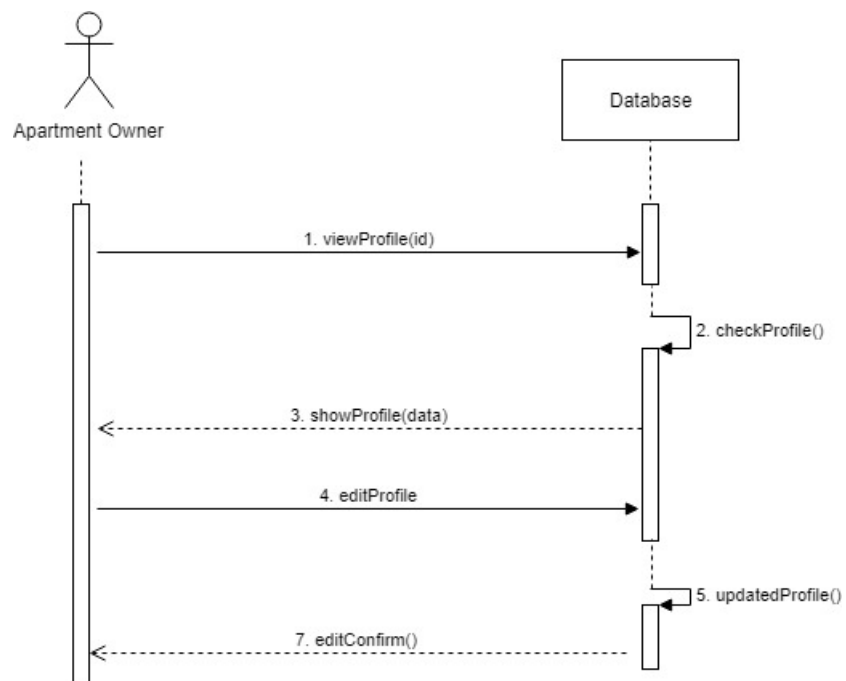


Figure 3.24: Sequence Diagram for Apartment owner edits their profile

3.3.8 Apartment Owner Can Visit maid-servant profile

The maid-servant profile who are assigning to apartment owner, and apartment owner can view only those specific maid-servant profile.

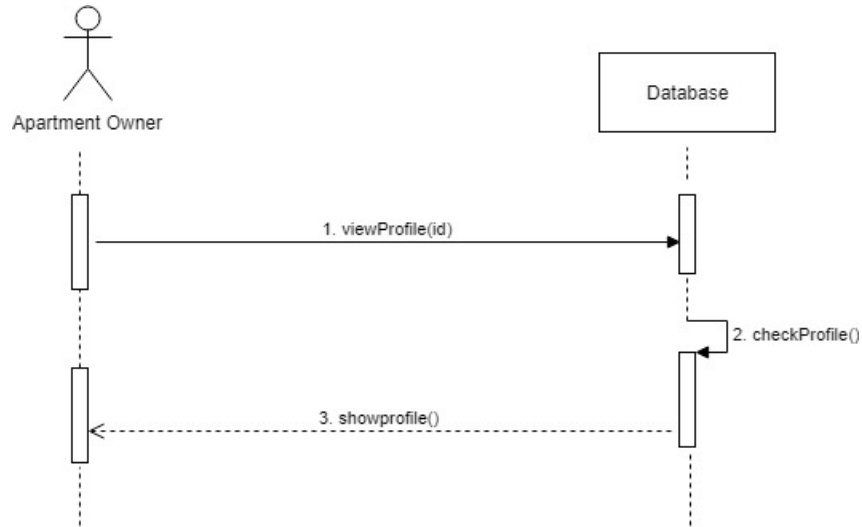


Figure 3.25: Sequence Diagram for Apartment owner Visit maid-servant profile

3.3.9 Login for Admin

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

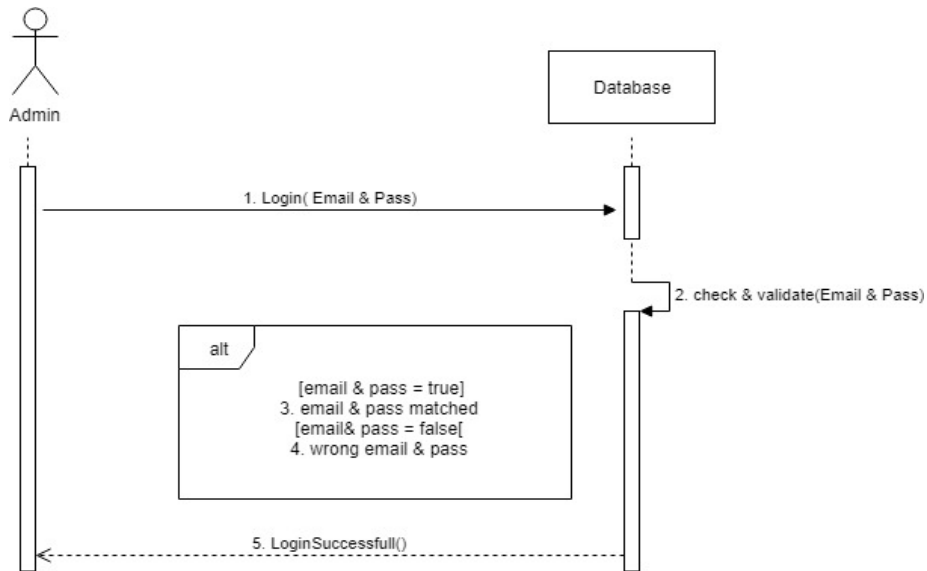


Figure 3.26: Sequence Diagram for Admin Login

3.3.10 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.

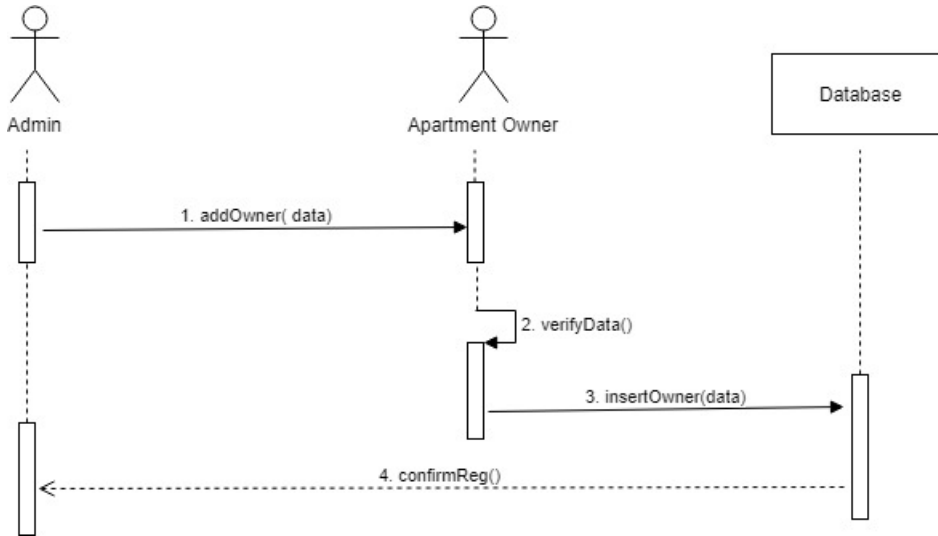


Figure 3.27: Sequence Diagram, registration for Apartment Owner

3.3.11 Admin create maid-servant

Admin also created maid-servant profile with the information

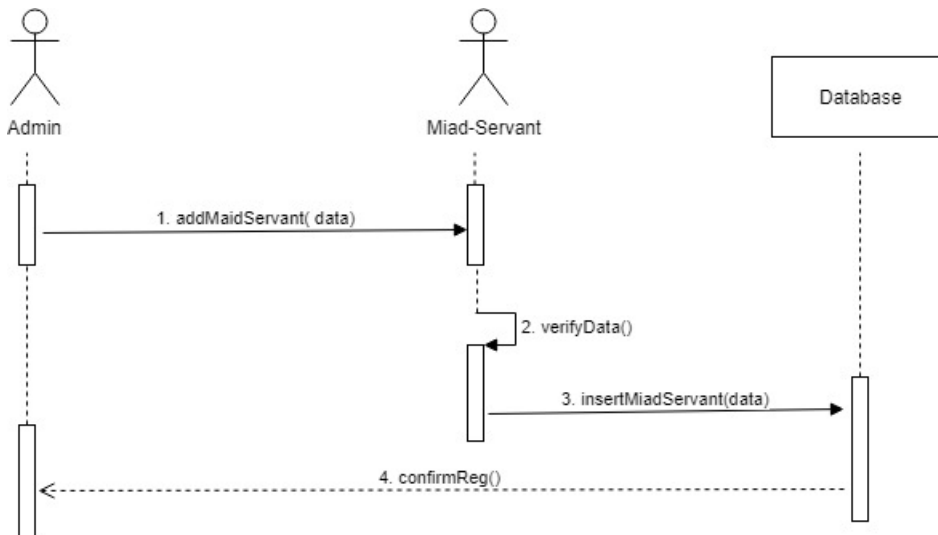


Figure 3.28: Sequence Diagram, registration for maid-servant

3.3.12 Admin view Apartment Owner's post history

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

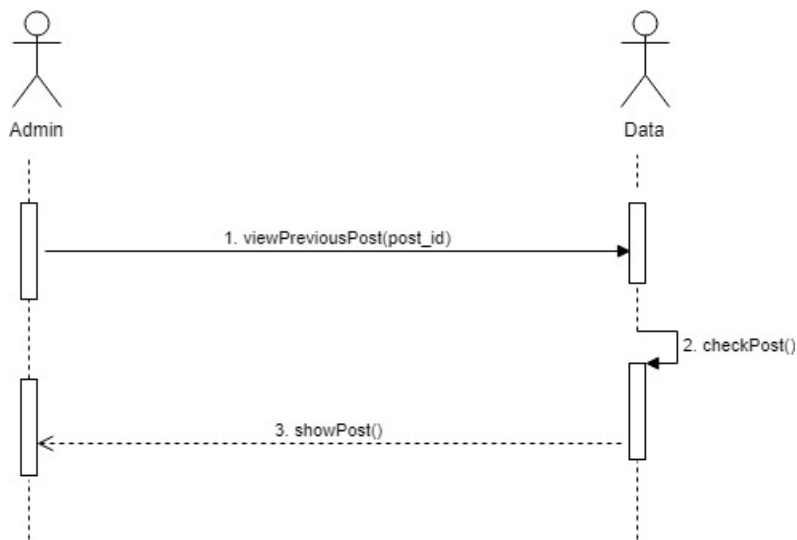


Figure 3.29: Sequence Diagram, Admin view Apartment Owner's post history

3.3.13 Admin view Apartment Owner's recent post

The recent post of apartment owner is also visible to admin

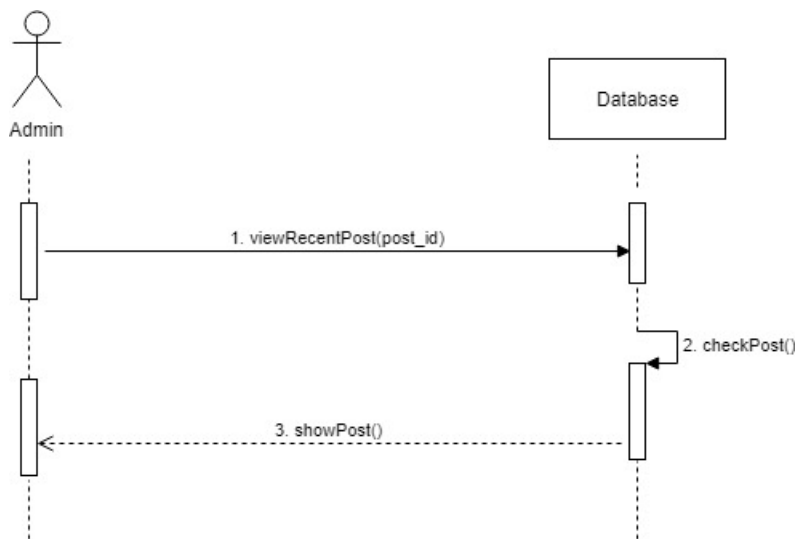


Figure 3.30: Sequence Diagram, Admin view Apartment Owner's recent post

3.3.14 Admin edits Apartment Owner profile

Admin can edit the apartment owner profile

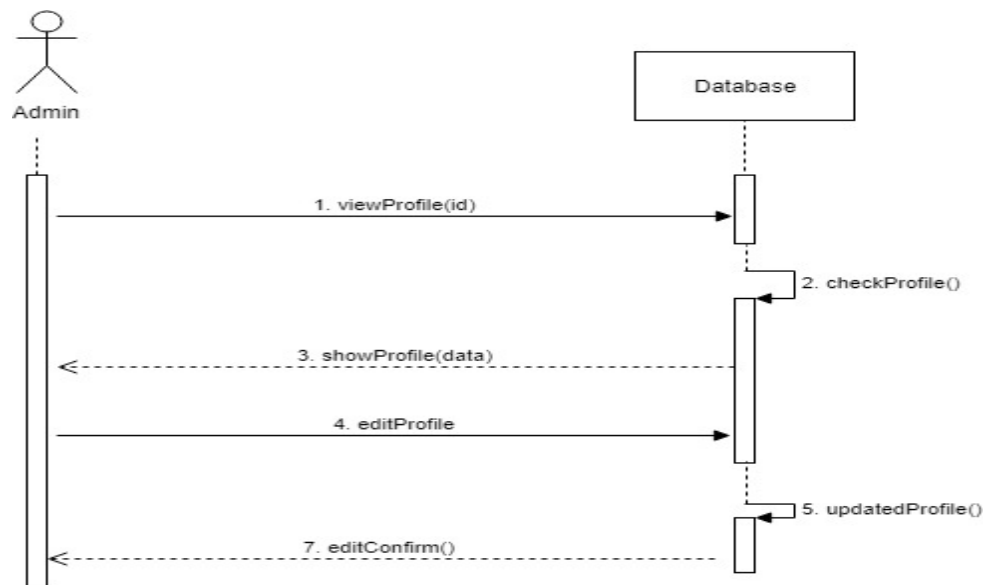


Figure 3.31: Sequence Diagram, Admin edits Apartment Owner profile

3.3.15 Admin deletes Apartment Owner profile

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

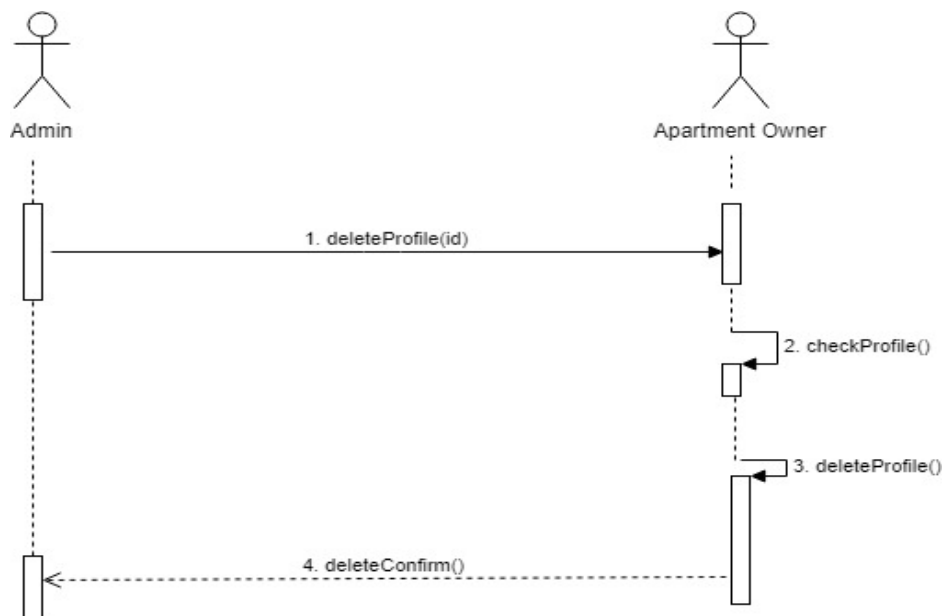


Figure 3.32: Sequence Diagram, Admin deletes Apartment Owner profile

3.3.16 Admin send maid-servant list

After getting the maid-servant assigning request, admin send some maid-servant profile list to apartment owner for confirmation

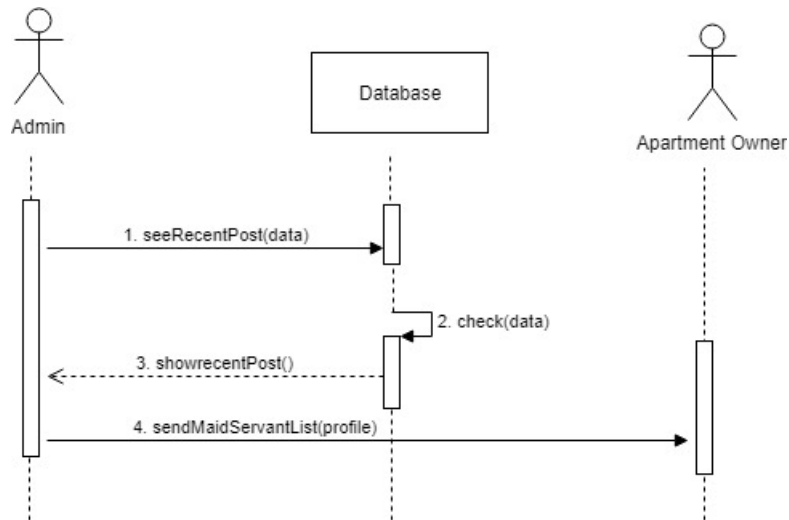


Figure 3.33: Sequence Diagram for assigning maid-servant

Chapter 4

System Design Specification

4.1 Development tools and technology

To make the software dynamic and develop, it is essential to use some tools. There are too many tools for develop any software. Different types of tools, which are used Software developers, can present the software development process very easy. In my project, some of tools I am using to develop my project. Now I'll mention the name of some tools that I used to develop for my project.

4.1.1 User Interface Technology

When any application is made, the software is brought to the user via the interface. For the success of a software application, the user interface plays an important role. Therefore, the importance of user interface is very superior into the application Users can easily access the system via the user interface. User interface includes using good image, content, action buttons, style sheets etc.

4.1.1.1 CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) is a style sheet language. HTML elements can arrive in front of us in many ways. After the end of the markup, cascading style sheets are designed. CSS is the standard way to format HTML pages and ways to make the page more elusive. There are three types of CSS style sheets to write. Which includes internal CSS, External CSS and Inline CSS .But external css is used most often? Because, using external CSS, all CSS information can be kept in different files. External CSS works better than internally and inline CSS into a project. We used external cascading style sheets when we have many HTML attributes and we can use them as required.

4.1.1.2 Bootstrap

Bootstrap is a free and open-source front-end web framework. In our project, we used bootstrap for front-end design. Bootstrap provides a set of files containing style sheets that give basic definitions. It also provides some JavaScript elements .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 library which uses JavaScript as core programming language. It is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating. It can override cascading style sheets .In my project, we use a very few JQuery function.

4.1.1.4 Programming Language

For developing an application, the only way to make a system is to use the programming language. Programming language plays a vital role into any application. Any system can be developed using the programming language. Basically, we are going to use PHP programming language for back-end side in our project. PHP is stand for Hypertext Preprocessor. Besides, PHP programming language is a general purpose programming language.

4.1.2 Implemented tools and platform

Some platform tools are needed for software development and some useful technologies. It is very important to determine which tools and platforms are required into my requirements. 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 server software or hardware dedicated to running software that is capable of satisfying the requests of the World Wide Web client. 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 server supports Cross platform. Apache is free and open source software to use. Develop under the auspice of the Apache Software Foundation. It has many feature and variety of use.

4.1.2.3 Database Server

For the development of our entire project, we have followed the relational database management system or RDBMS. We use MySql server. MySql is an open source Relation Database Management System 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.

4.2 Class Diagram

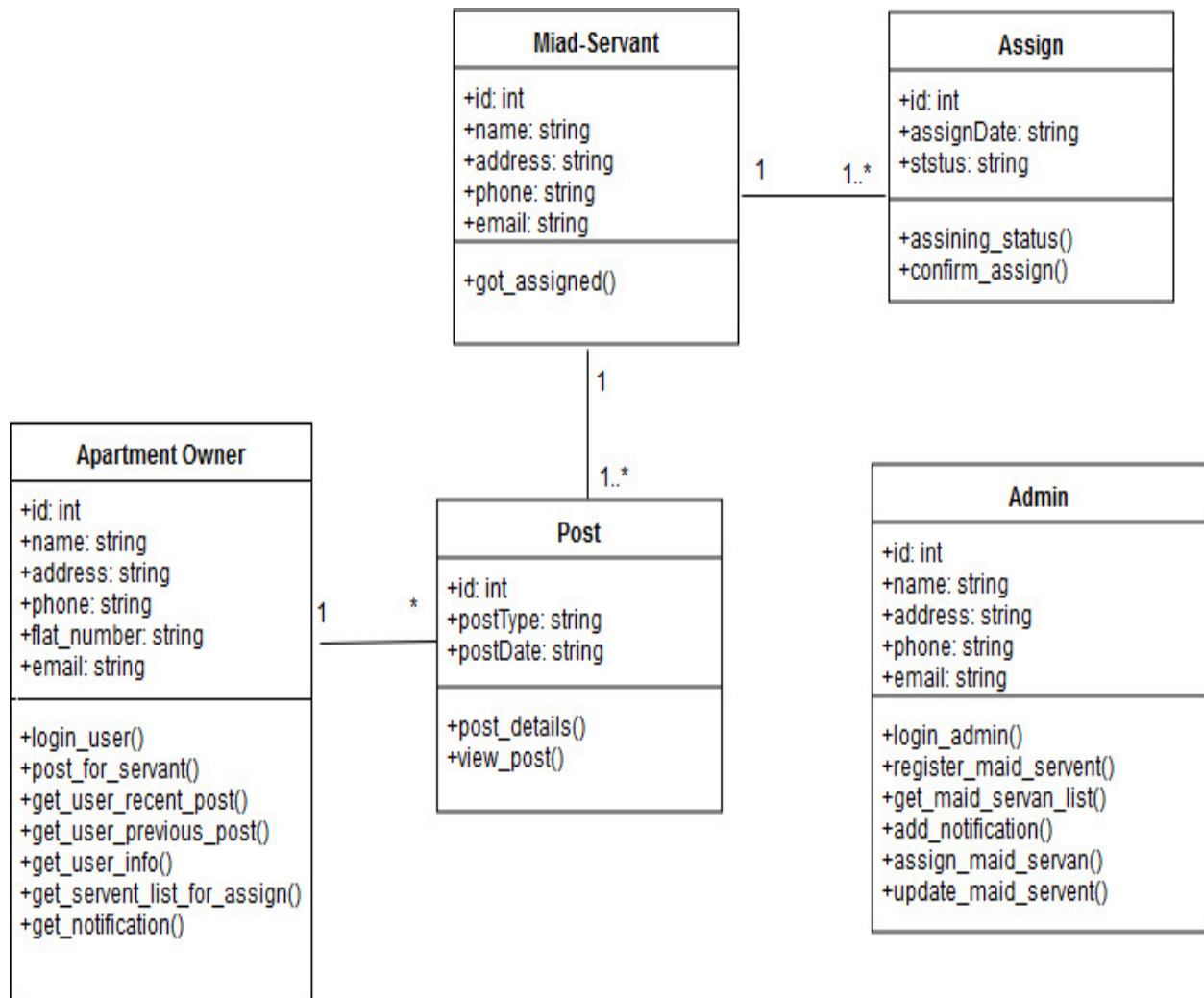


Figure 4.1: Class Diagram

4.3 Database Design Diagram

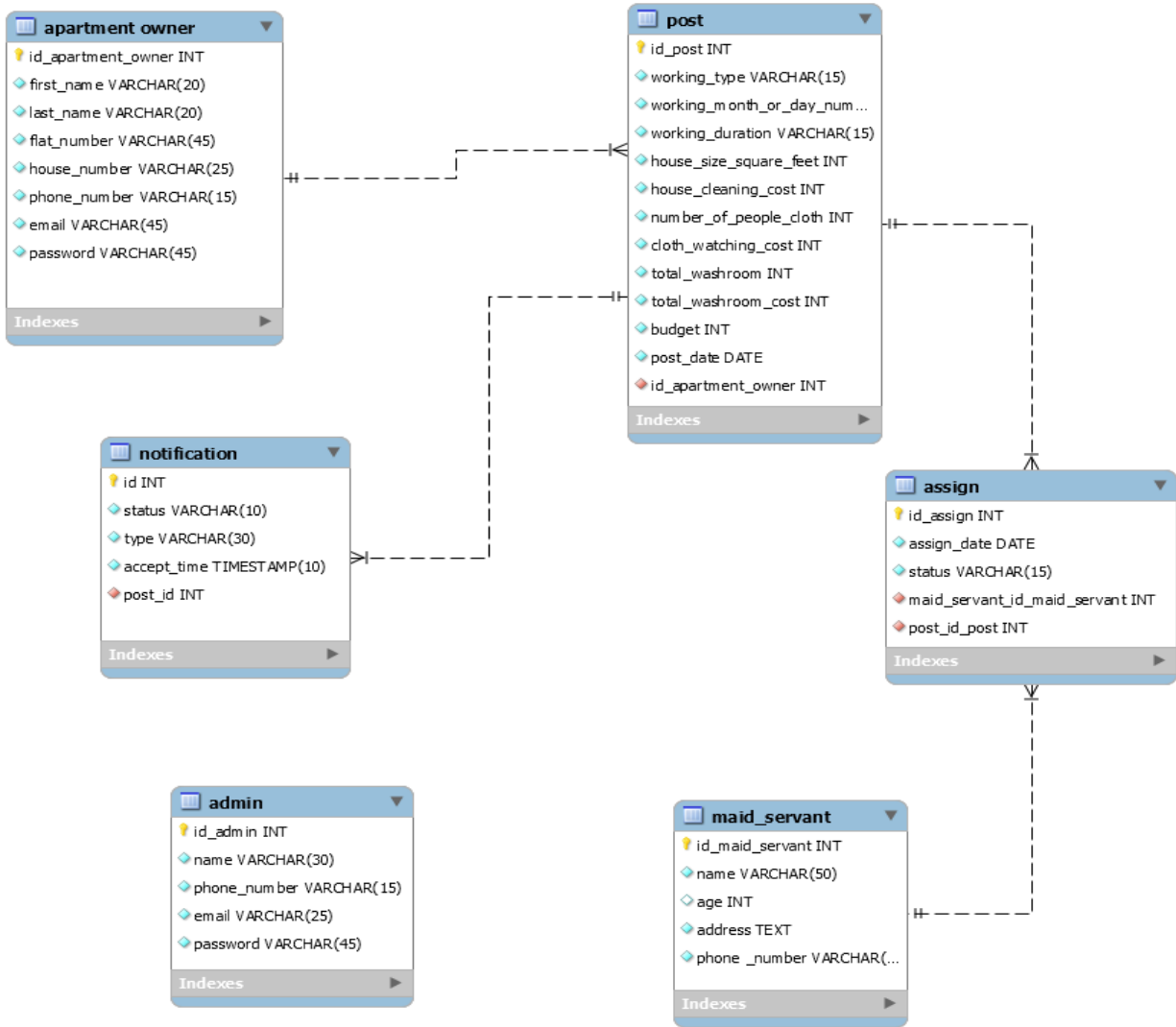


Figure 4.2: Database Diagram

Chapter 5

System Test

5.1 Testing Features

Feature Testing is the process of making change to add or modify the new functionality to the existing project. Testing these features is extremely important and they ought to be tested thoroughly in every stage of the development. 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 Features to be tested

Apartment owner part

Features	Priority	Description
Login	3	User must be authenticated by login
Post for maid-servant	3	Apartment owner provide all required information & submit the request post for maid-servant
Confirm the maid servant	3	After getting maid-servant list, apartment owner choose and confirm a maid-servant.
See previous history	2	Apartment owner can able to see their previous post history
See recent post	2	Apartment owner can able to see his recent post
View profile	1	Apartment owner click on "Profile" button to see his profile.
Edit profile	1	Apartment owner can edit there profile
View Maid-Servant Profile	2	Apartment owner can view the Maid-Servant Profile

Admin (System maintainer) part

Features	Priority	Description
----------	----------	-------------

Login	3	Admin must be authenticated by login
Create apartment owner profile	3	Admin can create the apartment owner profile
Create maid-servant profile	3	Admin can create the maid-servant profile
See previous history	2	Apartment Owner can able to see their previous post history
See recent post	3	Admin can see his recent post
Edit Profile	1	Admin can edit maid-servant profile
Delete Profile	2	Admin can delete both apartment owner and maid-servant profile
Send assign maid-servant list	3	Apartment owner post for maid-servant, then admin get this notification & send assign maid-servant list

Technological Features

Features	Priority	Description
Database	3	It is essential to use the database in almost every operation of our system. So, this part must be firmly controlled

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

5.2 Testing Strategy

The test strategy is a sketch that describes the experimental development of software development. Test Strategy is considered as common objective of the testing process. It is designed to highlight some important issues related to the testing process to project managers, testers and developers. Testing object, testing function methods, total available resources are included to the testing strategy. The strategy that is prepared for quality assurance should be reviewed with the application developers. This is a test guide which is followed in order to implement the test types. Examination strategies work with test aims, test environments, test methods, automation tools and strategies, and risk analysis. It is based on development design documents. It defines the main goal and helps achieve goals.

5.2.1 Test approach

A test approach is the test strategy implementation of a project. It defines the design to test and test all the exams. The testing method is made by a separate tester, according to the module or application means its own opinion or approach to its 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 mainly based on software requirements and specifications. Black box testing can be both of functional or non-functional. Black Box Testing attempts to derive sets of inputs that will fully exercise all the functional requirements of a system. In Black box Testing we will focus on the inputs and outputs of the software system.

5.2.1.2 Equivalent Class Partitioning

Equivalent class partitioning is known as equivalent partitioning. Equivalent Class Partitioning is also a black box technique. In Equivalent Class Partitioning inputs of the software are divided into group. That group of data is expected to perform as similar behavior. 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. Equivalence partitioning is a testing technique where input values set into classes for testing.

5.2.1.3 Boundary Value Analysis

Boundary value analysis and Equivalence Class Partitioning both are test case design techniques in black box testing. Boundary value analysis is software testing by which input data can be included within a boundary range. 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:

- 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

Test Phase	Time
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 maid-servant	Use-case 3.1.2	Test-case 5.6.2	
BR-3	Functional	Confirm maid-servant	Use-case 3.1.3	Test-case 5.6.3	
BR-4	Functional	See previous history	Use-case 3.1.4	Test-case 5.6.4	
BR-5	Functional	See recent post	Use-case 3.1.5	Test-case 5.6.5	
BR-6	Functional	View profile	Use-case	Test-case	

			3.1.6	5.6.6	
BR-7	Functional	Edit Profile	Use-case 3.1.7	Test-case 5.6.7	
BR-8	Functional	View Maid-Servant Profile	Use-case 3.1.8	Test-case 5.6.8	

Admin (System maintainer) part

Project Manager			Business Analyst Lead		
QA Lead			Target Implementation Date		
BR#	Category /Functionality/ Activity	Requirement Description	Use Case Reference	Test Case Referen ce	Comments
BR-1	Functional	Login	Use-case 3.1.9	Test- case 5.6.9	
BR-2	Functional	Registration apartment owner	Use-case 3.1.10	Test- case 5.6.10	
BR-3	Functional	Create Maid- Servant Profile	Use-case 3.1.11	Test- case 5.6.11	
BR-4	Functional	See previous history	Use-case 3.1.12	Test- case 5.6.12	
BR-5	Functional	See recent post	Use-case 3.1.13	Test- case 5.6.13	
BR-6	Functional	Edit Profile	Use-case 3.1.14	Test- case 5.6.14	
BR-7	Functional	Delete Profile	Use-case 3.1.15	Test- case 5.6.15	
BR-8	Functional	Send maid- servant profile list	Use-case 3.1.16	Test- case 5.6.16	

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 Admin
- Developers
- Tester

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

Test case #1	Test case name: Log in
---------------------	-------------------------------

System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A			
Designed by: Md. Shakib Kazi		Designed date: 15-Mar-2019			
Executed by:		Executed date:			
Short description: The registered users need to login enters into the application. And before that our application will check the authentication and authorization.					
Pre-conditions:					
<ul style="list-style-type: none"> • Users are always redirected to the login page if they have not authenticated by our application. • Assume that, the email is mamun123@gmail.com and password is “123456” 					
Step	Email	Password	Response	Pass/Fail	Comment
1	kazi@gmail.com	123411	Wrong email and password	Fail	
2	pakhi@gmail.com		Wrong email and password	Fail	
3		123456	Email can't be blank	Fail	
4	mamun123@gmail.com	123456	Successfully login into the application	Pass	
Post-conditions: Apartment owner will successfully login into the application.					

5.6.2 Post for maid-servant

Test case #2	Test case name: Post for Maid-Servant
---------------------	--

System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 15-Mar-2019		
Executed by:		Executed date:		
Short description: Apartment owner provide all required information. Submit the request post for maid-servant.				
Pre-conditions:				
<ul style="list-style-type: none"> • The user type should be the owner of the apartment. • The apartment owner must log in to the system. • To submit the post, fill up all require information. 				
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.	The fields have been completed and the post also.	Pass	
Post-conditions: After submit the post for maid-servant; apartment owner can see that post in recent post.				

5.6.3 Confirm maid-servant

Test case #3		Test case name: Confirm maid-servant		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: After submitted the maid-servant assigning request, admin send some maid-servant profile list to apartment owner for final confirmation.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be apartment owner • Apartment owner must be authenticated • Apartment owner have a maid-servant assigning request 				
Step	Action	Response	Pass/Fail	Comment
1	Apartment owner not choose and not confirm any maid-servant profile	No action has been occurred then.	Fail	
2	Apartment owner choose and confirm any maid-servant profile	Apartment owner confirm the maid-servant	Pass	
Post-conditions Apartment Owner choose and confirm the maid-servant profile				

5.6.4 See previous history

Test case #4		Test case name: See previous history		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Apartment owner click on “Previous History” button to see their previous history. Previous history data appear to apartment owner.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Apartment Owner • Apartment Owner must be logged in • Apartment owner have some previous activities 				
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	
Post-conditions: Apartment Owner can able to see his previous history				

5.6.5 See recent post

Test case #5		Test case name: See recent post		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Apartment owner click on “Recent Post” button to see his previous history. Previous history data appear to apartment owner.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Apartment Owner • Apartment Owner must be logged in • Apartment owner have some previous activities 				
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	
Post-conditions: : Apartment Owner can able to see his recent post				

5.6.6 View profile

Test case #6		Test case name: View profile		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Apartment owner click on “Profile” button to see his profile. Apartment owner profile appears to apartment owner.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Apartment Owner • Apartment Owner must be logged in • Apartment owner have a profile 				
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	
Post-conditions: Apartment Owner can able to view his own profile				

5.6.7 Edit Profile

Test case #7		Test case name: View profile		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Apartment owner go to their profile and they can edit their profile.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Apartment Owner • Apartment Owner must be logged in • Apartment Owner have a profile 				
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.	The fields have been completed and the post also.	Pass	
Post-conditions: Apartment Owner can edit their own profile				

5.6.8 View Maid-Servant Profile

Test case #8		Test case name: View Maid-Servant Profile		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Maid-Servant data comes from their profile. Maid-Servant profile appears to apartment owner.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Apartment Owner • Apartment Owner have a post for maid-servant 				
Step	Action	Response	Pass/Fail	Comment
1	Apartment owner have no post for maid-servant	Maid-servant profile are not sending to apartment owner	Fail	
2	Apartment owner have a maid-servant assigning post.	Admin send maid-servant list to apartment owner and apartment owner view those profile	Pass	
Post-conditions: Maid-Servant profile visible to Apartment Owner				

5.6.9 Login for Admin

Test case #9		Test case name: Log in for Admin			
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A			
Designed by: Md. Shakib Kazi		Designed date: 15-Mar-2019			
Executed by:		Executed date:			
Short description: The admin need to login for enters into the application. And before the login our application will check the authentication and authorization.					
Pre-conditions:					
<ul style="list-style-type: none"> • User type must be Admin • Assume that, the email is kazi@gmail.com and password is “123456” 					
Step	Email	Password	Response	Pass/Fail	Comment
1	Kazi1230@gmail.com	123411	Wrong email and password	Fail	
2	pakhi@gmail.com		Wrong email and password	Fail	
3		123456	Email can't be blank	Fail	
4	kazi@gmail.com	123456	Successfully login into the application	Pass	
Post-conditions: Admin will successfully login into the application.					

5.6.10 Create apartment owner profile

Test case #10		Test case name: Create apartment owner profile		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 15-Mar-2019		
Executed by:		Executed date:		
Short description: The admin create apartment owner profile and give it to the apartment owner. And apartment owner enter the system by using this profile.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Admin • Admin must be authenticated. • Admin have apartment owner information 				
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	
Post-conditions: An apartment owner profile is created.				

5.6.11 Create maid-servant profile

Test case #11	Test case name: Create maid-servant profile			
System: Maid-Servant Assigning for a Colony Based Area	Subsystem: N/A			
Designed by: Md. Shakib Kazi	Designed date: 15-Mar-2019			
Executed by:	Executed date:			
Short description: The admin create maid-servant profile with valid require information like name, phone number, address etc.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Admin • Admin must be authenticated. • Admin have maid-servant information 				
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 maid-servant.	Adding maid-servant is successful	Pass	
Post-conditions: A maid-servant profile is created.				

5.6.12 Admin view previous history

Test case #12		Test case name: Admin view previous history		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Admin can see all types of previous history information. Previous history information is enabled to see admin. .				
Pre-conditions: <ul style="list-style-type: none"> • User type must be Admin • Admin must be logged in • Apartment owner have some previous activities record 				
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	
Post-conditions: Admin can able to see all type of previous history				

5.6.13 Admin view recent post

Test case #13		Test case name: Admin view recent post		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Admin can view recent post that is performed by apartment owner. Recent post data appear to admin.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Admin • Admin must be logged in • Apartment owner have some recent post 				
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 allrecent post	Pass	
Post-conditions: Admin can able to see all recent post				

5.6.14 Admin edits profile

Test case #14		Test case name: Edit profile		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Apartment owner go to their profile and they can edit their profile.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Admin • Admin must be logged in • Apartment owner have a profile • Maid-servant have a profile 				
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	
Post-conditions: Apartment Owner profile is updated				

5.6.15 Admin deletes Profile

Test case #15		Test case name: Delete profile		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: Apartment owner go to their profile delete the profile.				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Admin • Admin must be logged in • Apartment Owner have a profile 				
Step	Action	Response	Pass/Fail	Comment
1	Go to profile and hit the “Delete” button	The profile is deleted	Pass	
Post-conditions: Apartment Owner or Maid-Servant profile is deleted				

5.6.16 Send maid-servant profile list to apartment owner

Test case #16		Test case name: Send maid-servant profile list to apartment owner		
System: Maid-Servant Assigning for a Colony Based Area		Subsystem: N/A		
Designed by: Md. Shakib Kazi		Designed date: 05-Apr-2019		
Executed by:		Executed date:		
Short description: After getting the maid-servant assigning request, admin send some maid-servant profile list to apartment owner				
Pre-conditions:				
<ul style="list-style-type: none"> • User type must be Admin • Admin must be authenticated • Admin have a maid-servant assigning request 				
Step	Action	Response	Pass/Fail	Comment
1	Admin is not sending any maid-servant profile	No action has been occurred then.	Fail	
2	Admin send some maid-servant profile list.	Apartment owner get some maid-servant profile list for confirmation	Pass	
Post-conditions Apartment get the maid-servant profile list				

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. The apartment owner can go through this page to see slaves, maid service, laundry services, and other pages, with previous posts. Now I will provide the screenshot of login page below.

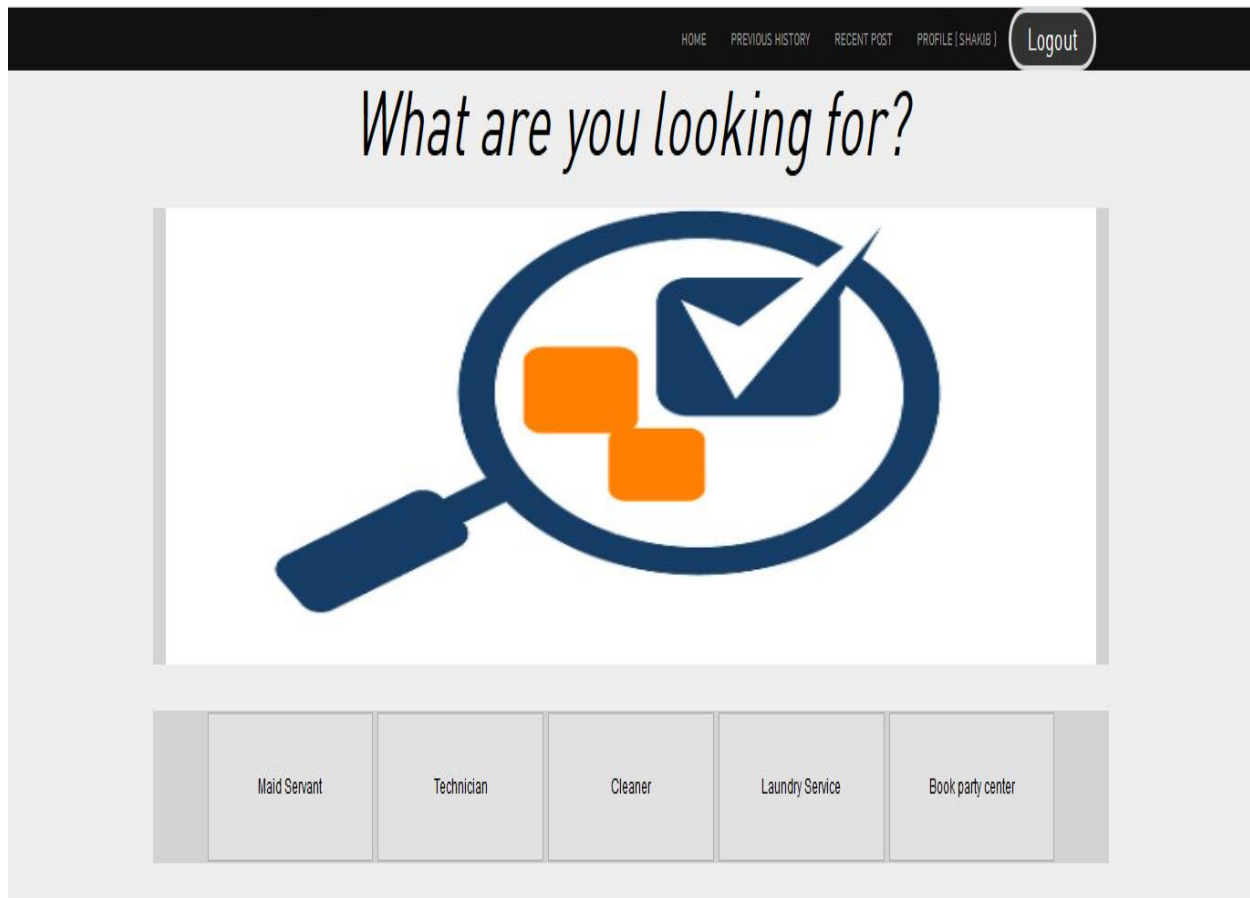


Figure 6.2: Home page of apartment owner

6.3 Apartment owner post for maid-servant

After login successfully, the apartment owner enters into home page and can able to post for Maid-Servant. For assign, a maid-servant apartment owner must be Fill-up some require field, then submit the post. Now I will provide the screenshot of login page below.

HOME PREVIOUS HISTORY RECENT POST PROFILE (SHAKIB) Logout

Make Your Post for Maid Servant

Select Working Type
Full Time

Working Duration
Day 1

Work Strating Date
mm / dd / yyyy

Work starting time
--:-- --

Work ending time
--:-- --

House cleaning (Square feet)
0 = 0

Cloth Washing
Select person * 10 Taka = 0

Toilet clening
Select toilet * 10 Taka = 0

Perday total cost
0

Details
Details

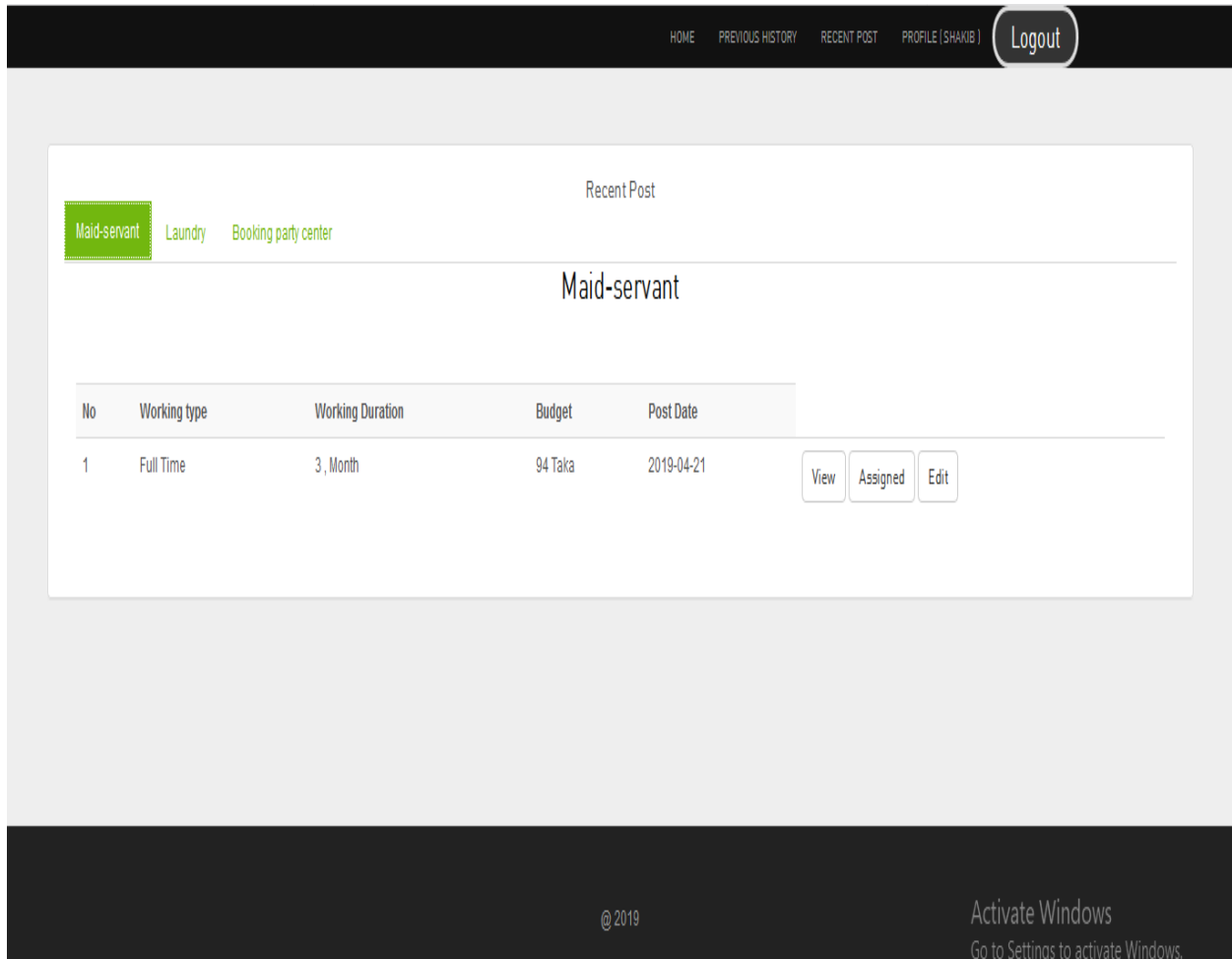
Submit

Do you forget to fill up some field ?

Figure 6.3: Apartment owner post for maid-servant

6.4 Apartment Owner sees recent post

Apartment Owner is able to see their recent post. Now I will show that page below.



The screenshot displays a web application interface. At the top, a dark navigation bar contains links for HOME, PREVIOUS HISTORY, RECENT POST, and PROFILE [SHAKIB], along with a Logout button. The main content area is titled 'Recent Post' and features a breadcrumb trail: Maid-servant > Laundry > Booking party center. Below this, the title 'Maid-servant' is centered. A table lists the post details:

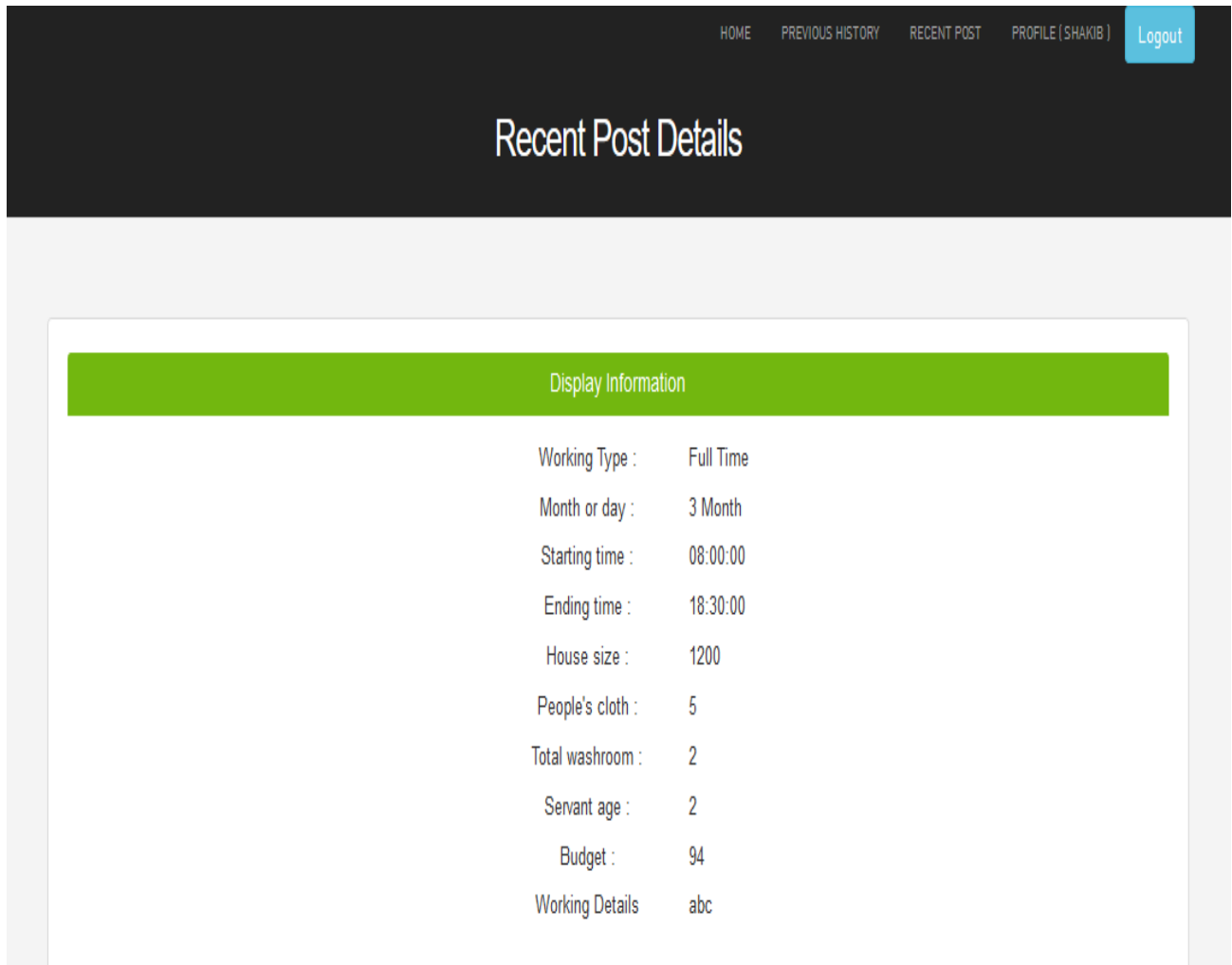
No	Working type	Working Duration	Budget	Post Date	
1	Full Time	3, Month	94 Taka	2019-04-21	View Assigned Edit

At the bottom of the page, there is a copyright notice '@ 2019' and a Windows watermark: 'Activate Windows. Go to Settings to activate Windows.'

Figure 6.4: Apartment owner sees recent post

6.5 Apartment Owner sees recent post details

Apartment Owner is able to see their recent post details also. Now I will show that page below.

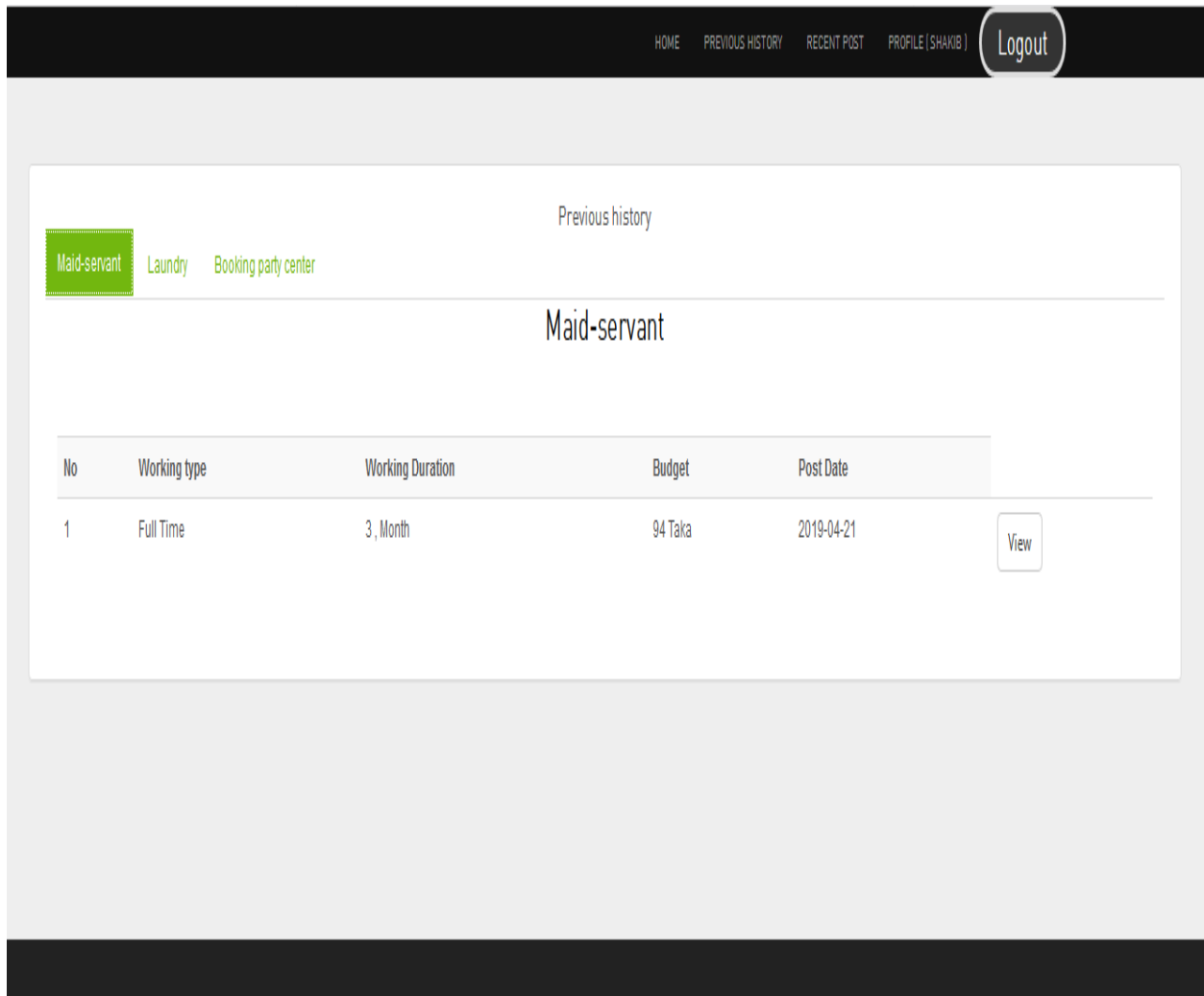


Display Information	
Working Type :	Full Time
Month or day :	3 Month
Starting time :	08:00:00
Ending time :	18:30:00
House size :	1200
People's cloth :	5
Total washroom :	2
Servant age :	2
Budget :	94
Working Details	abc

Figure 6.5: Apartment owner sees recent post details

6.6 Apartment Owner sees previous history

Apartment Owner is able to see their previous post history also. Now I will show that page below.



The screenshot displays a web interface for an apartment owner. At the top, a navigation bar includes links for HOME, PREVIOUS HISTORY, RECENT POST, and PROFILE (SHAKIB), along with a Logout button. The main content area is titled 'Previous history' and features a breadcrumb trail: Maid-servant > Laundry > Booking party center. The 'Maid-servant' category is selected. Below this, a table lists the user's previous posts. The table has five columns: No, Working type, Working Duration, Budget, and Post Date. A single entry is shown with a 'View' button to its right.

No	Working type	Working Duration	Budget	Post Date	
1	Full Time	3, Month	94 Taka	2019-04-21	View

Figure 6.6: Apartment owner sees previous history

6.7 Apartment Owner confirm assigning the maid servant

After submit the maid-servant assigning request, apartment owner get back some maid-servant list and choose the maid-servant from the list for final confirmation..Now I will show that page below.

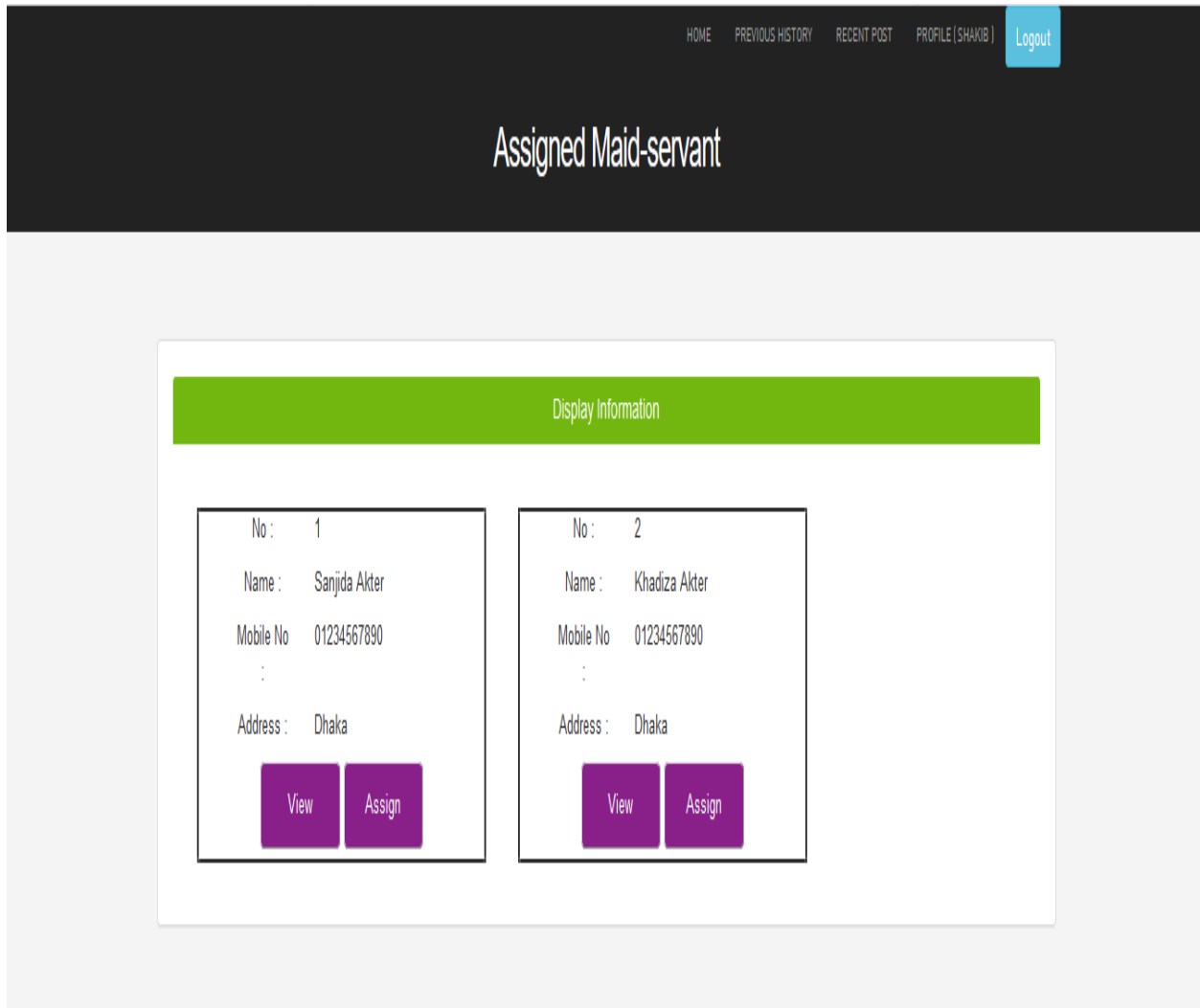


Figure 6.7: Apartment owner confirm assigning the maid servant

6.8 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.

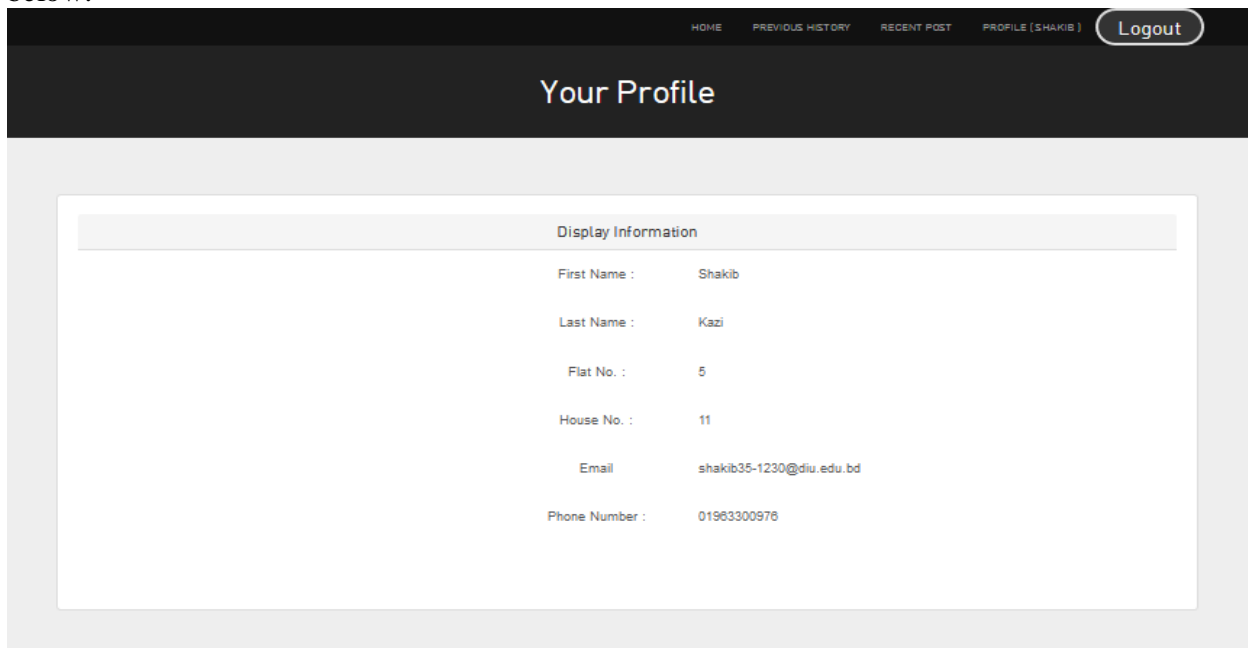


Figure 6.8: Apartment owner view his/her own profile

6.9 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.

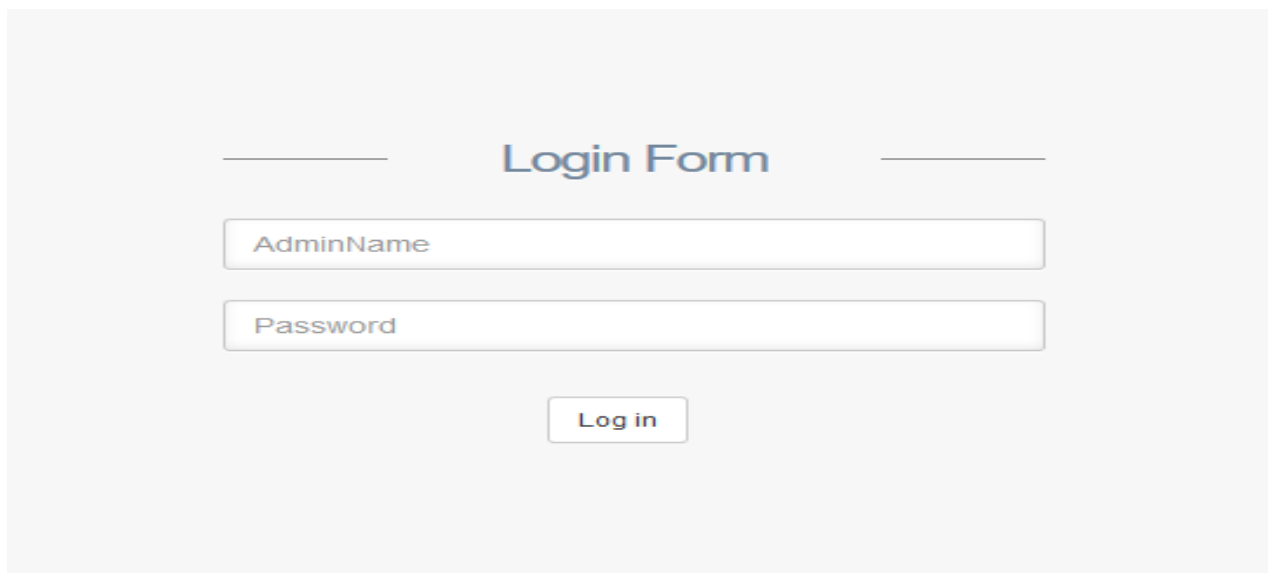
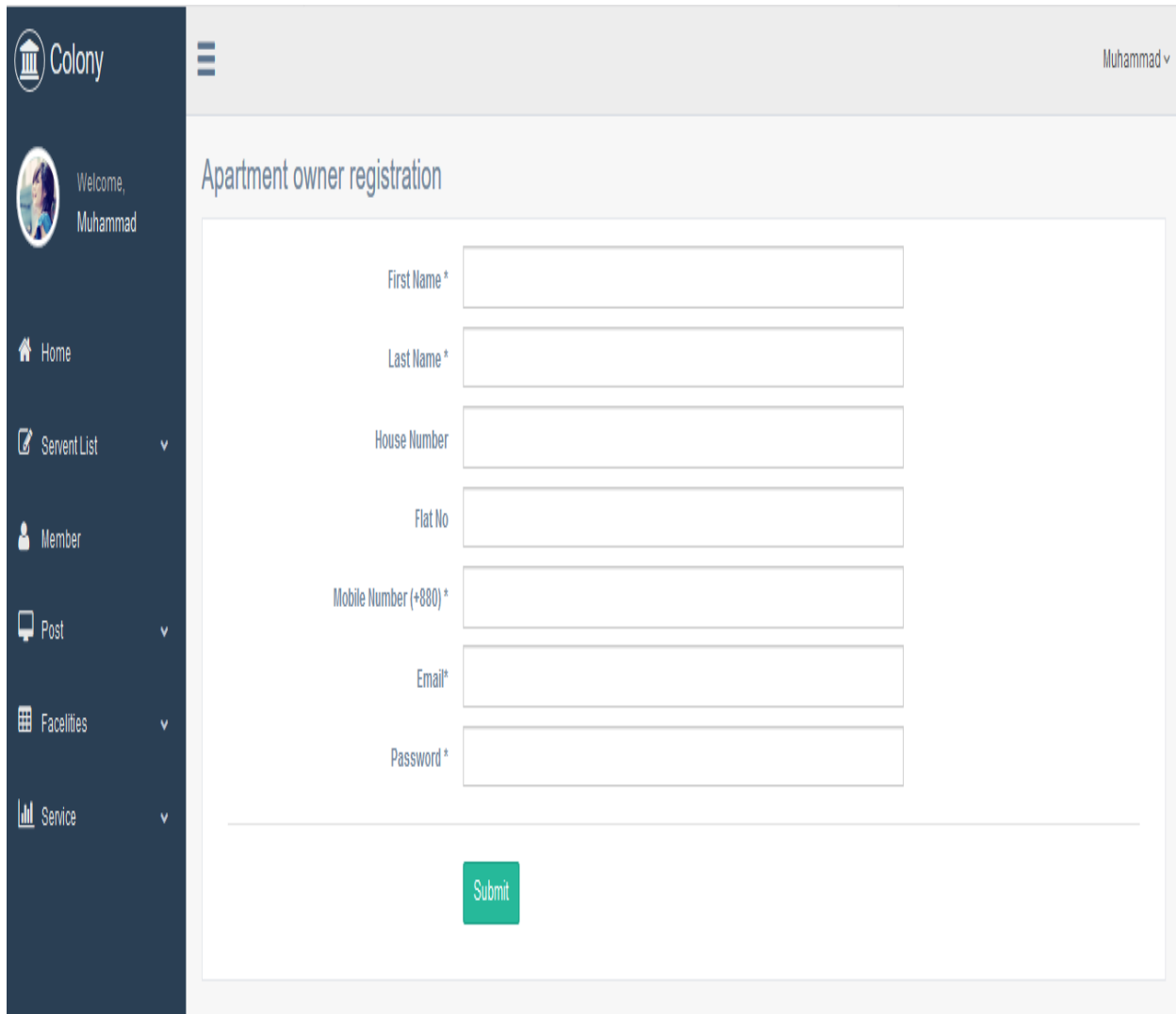


Figure 6.9: Admin login page

6.10 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.

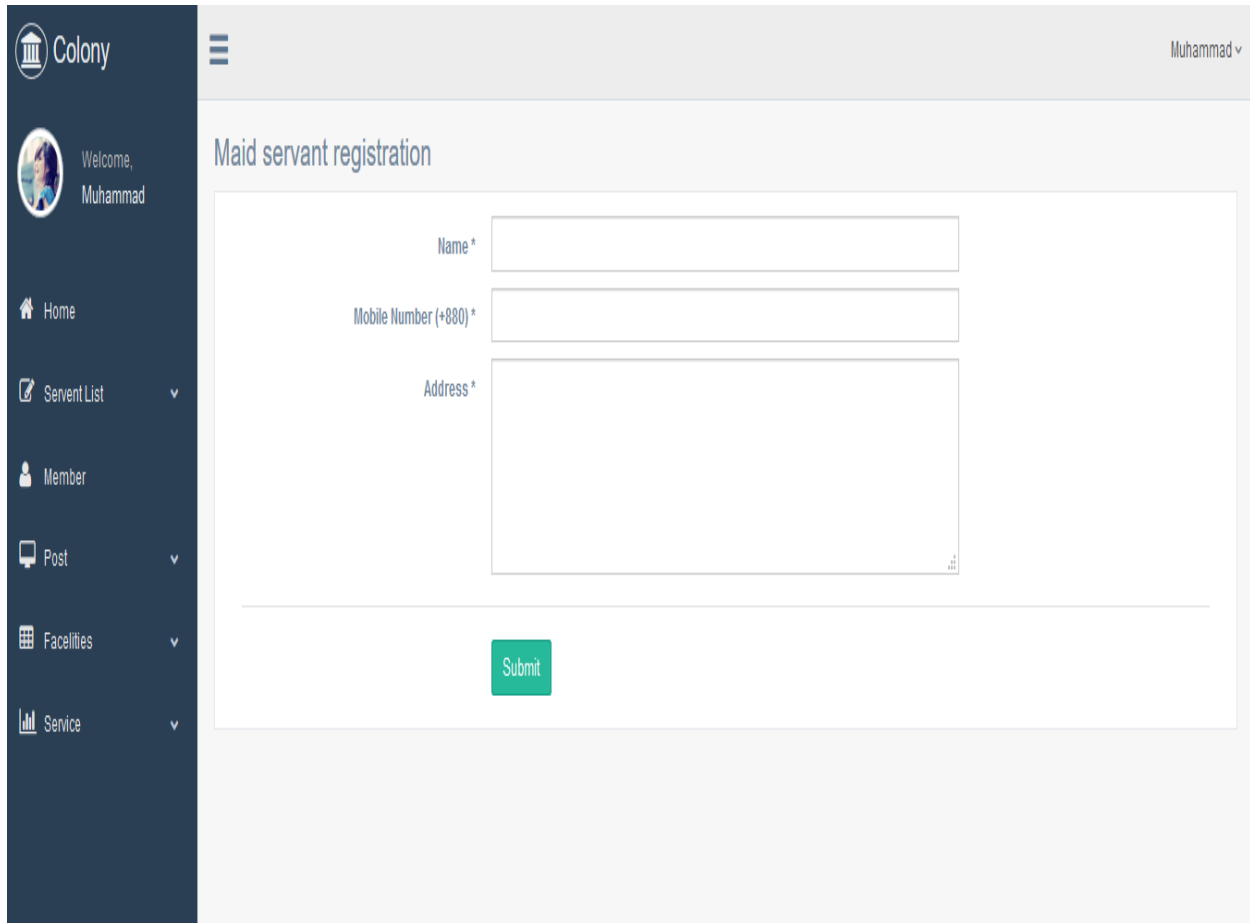


The screenshot shows a web application interface for apartment owner registration. On the left is a dark blue sidebar with the 'Colony' logo and a user profile for 'Muhammad'. The main content area is titled 'Apartment owner registration' and contains a form with the following fields: First Name*, Last Name*, House Number, Flat No, Mobile Number (+880)*, Email*, and Password*. A green 'Submit' button is located at the bottom of the form.

Figure 6.10: Apartment owner registration form

6.11 Admin create maid-servant profile

Admin also created maid-servant profile with the information such as Name, Address, and Phone. Now I will provide the screenshot of the page below.



The screenshot shows a web application interface for 'Maid servant registration'. On the left is a dark blue sidebar with the 'Colony' logo and a navigation menu including 'Home', 'Servent List', 'Member', 'Post', 'Facilities', and 'Service'. The main content area has a header with 'Welcome, Muhammad' and a title 'Maid servant registration'. The registration form contains three input fields: 'Name *', 'Mobile Number (+880) *', and 'Address *'. A green 'Submit' button is located below the form.

Figure 6.11: Admin create maid-servant profile

6.12 Maid-servant profile

Admin can access to maid-servant profile & view her all information. Besides, he can edit her profile. Now I will provide the screenshot of the page below.

The screenshot displays a web application interface for managing maid-servant profiles. On the left is a dark sidebar with the 'Colony' logo and navigation menu items: Home, Servent List, Member, Post, Facelities, and Service. The main content area is titled 'Maid servant Profile' and shows the profile for 'Sanjida Akhter'. It includes a profile picture, contact information (phone number 01234567890 and location Khilgaon, Dhaka), and an 'Accept' count of 10. A green 'Edit Profile' button is visible at the bottom of the profile card. The top right of the page shows the user 'Muhammad'.

Figure 6.12: Maid-servant profile

6.13 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.

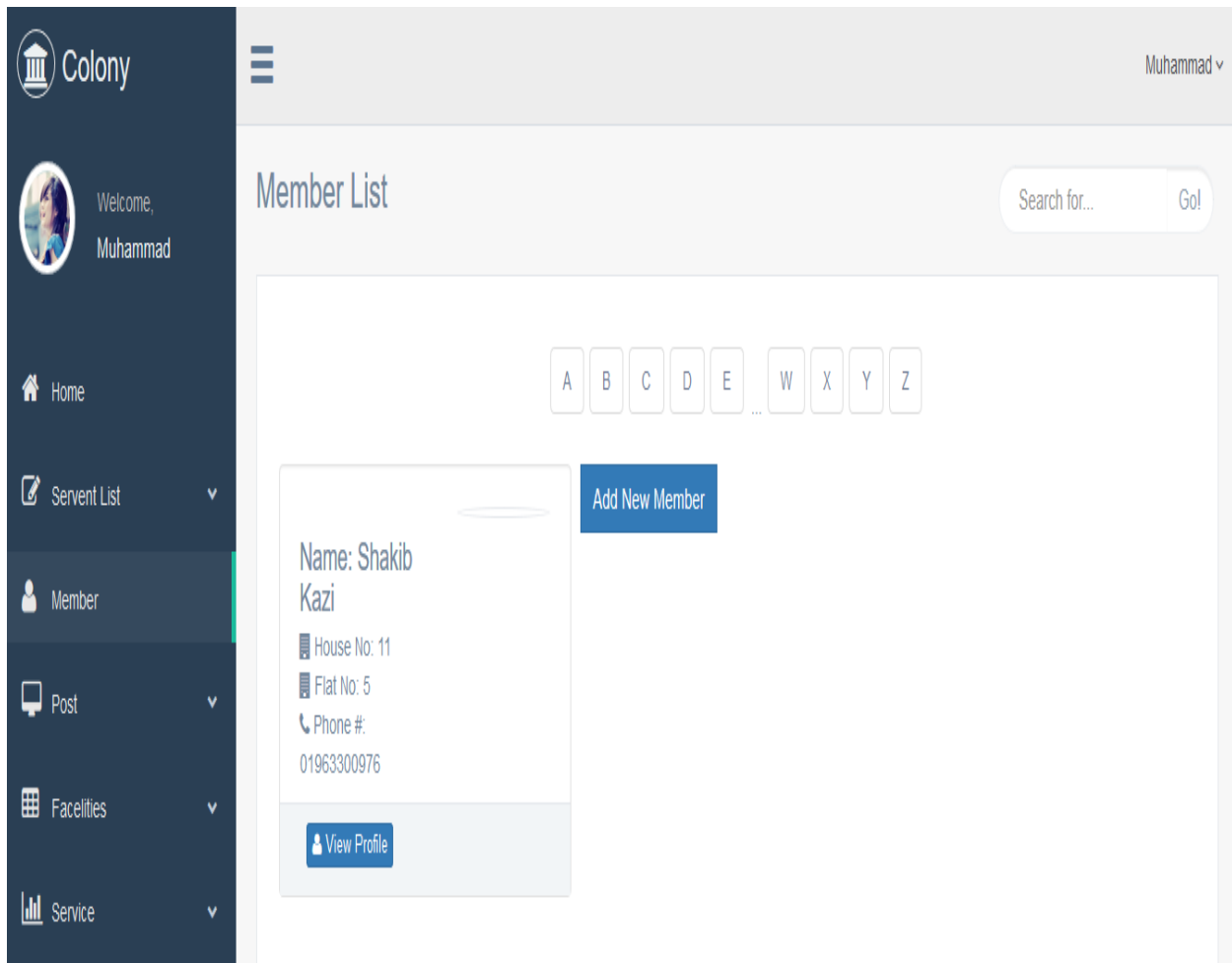


Figure 6.13: Apartment owner list & add apartment owner

6.14 Admin view recent post

Admin is able to see apartment owner's recent post. Now I will show that page below.

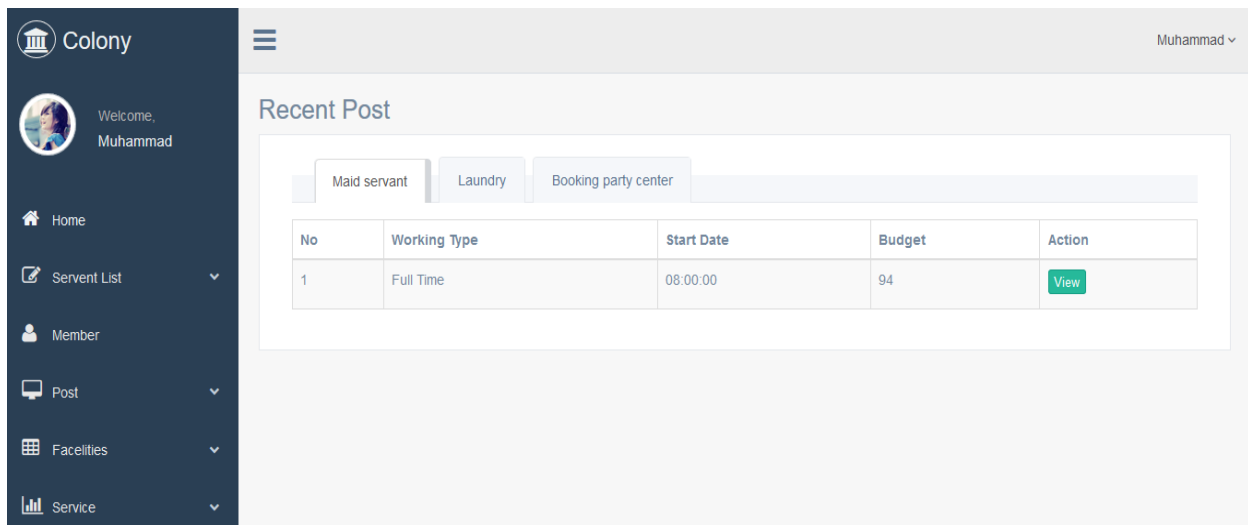


Figure 6.14: Admin view recent post

6.15 Admin view recent post details

Admin is able to see the apartment owner's recent post details also. Now I will show that page below.

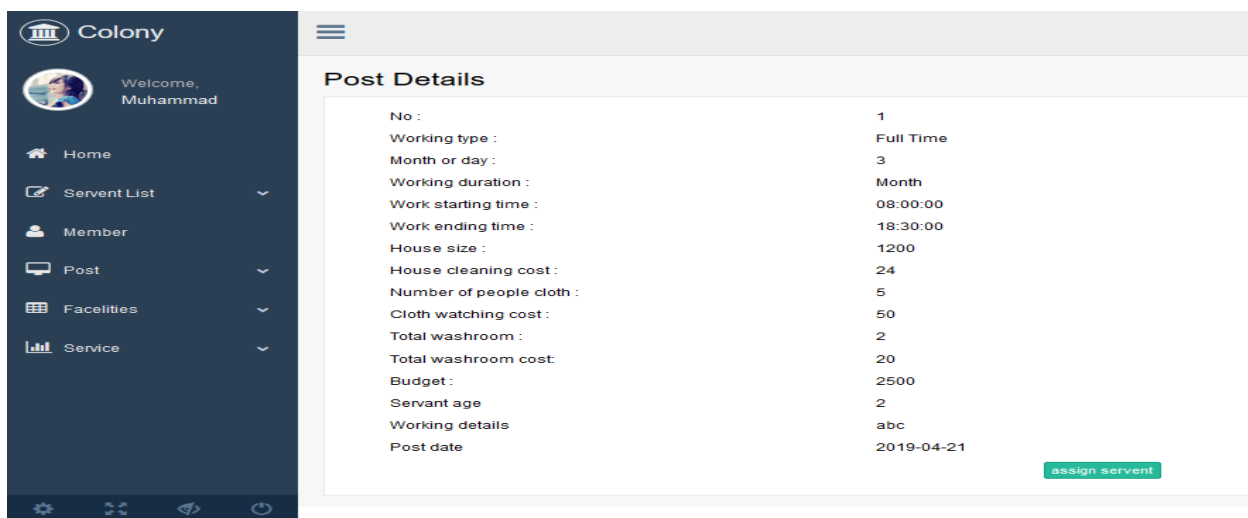


Figure 6.15: Admin view recent post details

Chapter 7

Conclusion

7.1 GitHub Link

https://github.com/Shakib1230/coloni_apartment_service

7.2 Project 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.

7.3 Limitations

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

- **Payment Method:** Payment method: We do not have any payment method directly in this application. So users can not make any payment transactions through our system. The user pays for him to use our system.
- **Only Web Version:** Our system is only web-based but there are many other platforms that are popular such as Mobile version, iOS version, etc.

7.4 Obstacles and Achievements

When a project is developing, there are many obstacles in front of you and those obstacles will challenge you and you have to overcome these challenges. You cannot learn anything without the

challenge, the challenge comes with the opportunity and here comes instead of burying your opportunity.

By doing this project we learn how to collect and analyze requirements. Brainstorming is a must for any project you're doing to make sure it's done. System analysis and database design that we learn about this project correctly. Our supervisor helps us at 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

When I completed my project, I gained a lot of knowledge from different types of platforms. I must mention the references, the references that I have been able to complete my project. For making my project successful those resources help me a lot. Not I will mention the names below.

- www.google.com
- www.youtube.com
- www.github.com
- www.wikipedia.com
- www.php.net
- www.mysql.com
- www.mysql.com/products/workbench
- www.getbootstrap.com
- www.w3schools.com
- www.jquery.com
- www.sublimetext.com/3
- www.apachefriends.org