

Guardio – Apartment Surveillance System

BY

**MD SOHAG HOSEN
ID: 191-15-12671**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Shah Md. Tanvir Siddiquee
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised By

Narayan Ranjan Chakraborty
Associate Professor
Department of CSE
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

DECEMBER 2022

APPROVAL

This Project/internship titled “Guardio - Apartment Surveillance System”, submitted by Md Sohag Hosen, ID No: 191-15-12671 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfilment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 26/01/2023.

BOARD OF EXAMINERS

Chairman

Dr. Touhid Bhuiyan

Professor and Head

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



Internal Examiner

Sazzadur Ahmed

Assistant Professor

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



Internal Examiner

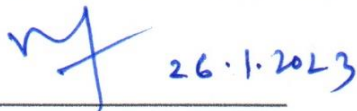
Ms. Sharmin Akter

Senior Lecturer

Department of Computer Science and Engineering

Faculty of Science & Information Technology

Daffodil International University



External Examiner

Dr. Ahmed Wasif Reza

Associate Professor

Department of Computer Science and Engineering

East West University

DECLARATION

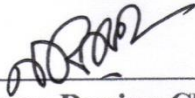
We hereby declare that, this project has been done by us under the supervision of **Shah Md. Tanvir Siddiquee, Assistant Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



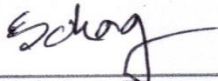
Shah Md. Tanvir Siddiquee
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:



Narayan Ranjan Chakraborty
Associate Professor
Department of CSE
Daffodil International University

Submitted by:



Md Sohag Hosen
ID: -191-15-12671
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete the final year project/internship successfully.

We really grateful and wish our profound our indebtedness to **Shah Md. Tanvir Siddiquee, Assistant Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of App and Web Development to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to Narayan Ranjan Chakraborty, Associate Professor and Prof. Dr. Touhid Bhuiyan, Head, Department of CSE, for his kind help to finish our project and also to other faculty member and the staff of CSE department of Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Our goal of this project is to give the apartment owner a complete and smart surveillance system and connection between Tenant and Apartment owner and security of apartment and Tenant living easy and hustle free. the Guardio system is one that aims to give the apartment a smart surveillance and high security. There are obviously many advantages of Guardio system for the Apartment owner, as well as the Tenant. The system will save money, time and hustle for both Apartment owner and Tenant. Tenant also will be able monitor his family members, guest, workers, deliveryman and their vehicles using our system. Apartment owner now can easily collect their rent using our system, they can also send any emergency notice to all the tenant from our system. Using our system tenant can send any objection to the Apartment owner directly. The system is very essential for the residential apartment where too many persons are lived, it will be an easy task to monitor them all. The Guardio apartment will insure safety and manage all associated things.

TABLE OF CONTENTS

CONTENTS	PAGE
Approval Page	i
Declaration	ii
Acknowledgements	iii
List of Figures	viii-ix
List of Table	x
CHAPTER	
CHAPTER 1: INTRODUCTION	1-3
1.1 Introduction	1
1.2 Motivation	1
1.3 Objective	1-2
1.4 Expected Outcome	2-3
1.5 Project Management and Finance	3
1.6 Report Layout	3
CHAPTER 2: BACKGROUND	4-5
2.1 Preliminaries/Terminologies	4
2.2 Related Work	4
2.3 Comparative Analysis	4-5

2.4 Scope of problems	5
2.5 Challenges	5
CHAPTER 3: REQUIRMENT SPECIFICATION	6-17
3.1 Business Process Model	6-11
3.2 Requirement Collection and Analysis	11-12
3.3 Use Case Modeling and Description	12-15
3.4 Logical Data Model	16
3.5 Design Requirements	17
CHAPTER 4: DESIGN SPECIFICATION	18-32
4.1 Front-end Design	18-29
4.2 Back-end Design	29-31
4.3 Interaction Design and User Experience (UX)	31
4.4 Implementation Requirements	31-32
CHAPTER 5: IMPLEMENTATION AND TESTING	33-35
5.1 Implementation of Database	33
5.2 Implementation of Front-end Design	33
5.3 Testing Implementation	33-34
5.4 Test Results and Reports	35

CHAPTER 6: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	46
	36
6.1 Impact on Society	46 36
6.2 Impact on Environment	46 36
6.3 Ethical Aspects	4366
6.4 Sustainability Plan	36 46
CHAPTER 7: CONCLUSION AND FUTURE SCOPE	37 47
	37
7.1 Discussion and Conclusion	37
7.2 Scope for Further Developments	48
	38
REFERENCES	

LIST OF FIGURES

FIGURES	PAGE NO
Figure 3.1: Operator System Business Process Model	6
Figure 3.2: Apartment Owner System Business Process Model	7
Figure 3.3: Tenant System Business Process Model	8
Figure 3.4: Operator System Block Diagram	9
Figure 3.5: Apartment Owner System Block Diagram	10
Figure 3.6: Tenant System Block Diagram	11
Figure 3.7: Operator System Use Case Model	13
Figure 3.8: Apartment Owner Use Case Model	14
Figure 3.9: Tenant Use Case Model	15
Figure 3.10: Logical Data Model	16
Figure 4.1: Entry Function	18
Figure 4.2: Exit Function	19
Figure 4.3: History	19
Figure 4.4: Setting	20
Figure 4.5: Add Member	20
Figure 4.6: Post Notice	21
Figure 4.7: Owner Splash Screen	22
Figure 4.8: Owner Homepage	22
Figure 4.9: Owner Navigation Menu	23
Figure 4.10: Owner Bills Section	23
Figure 4.11: Owner Add Notice Section	24
Figure 4.12: Owner Complain View Section	24
Figure 4.13: Tenant Splash Screen	25
Figure 4.14: Tenant Join Option	25
Figure 4.15: Tenant Homepage	26
Figure 4.16: Tenant Navigation Menu	26

Figure 4.17: Tenant Family Section	27
Figure 4.18: Tenant Guest Section	27
Figure 4.19: Tenant Workers Section	28
Figure 4.20: Tenant Bills Section	28
Figure 4.21: Tenant Notice View	29
Figure 4.22: Web Application Backend Files	30
Figure 4.23: API Files	30
Figure 4.24: Database	31

LIST OF TABLE

FIGURES	PAGE NO
Table 5.1: Test Case Result	33-34

CHAPTER 1

Introduction

1.1 Introduction

Guardio Apartment Surveillance System is a platform where a tenant and his family member, workers, guest will be monitored and Apartment owner can take house rent and notice for tenant, all access will be operated by the Operator. So we can consider our audience for this software is Apartment owners, Tenants and Operators. As Apartment owners need to collect their house rent so they have to use this platform often. Tenant will have unique QR code to show on the to enter and exit and they can also add other member for their apartment. Lastly we are considering Operator as our audience because Operator have to control these Apartment owners and Tenants information and have to manage the whole software from Guardio Operator end.

1.2 Motivation

Manual system involves guards who allow everybody just know the flat no or tenant name, if any thief enter using fake identity they can't detect. Sometimes on the big residential society they can't recognize the new tenant and disturb then every time they try to enter to the apartment. Unwanted guest come to house without permission. There is no record system for who enter into the apartment or leave the apartment. The apartment owner also face problem for collecting rent from the tenant. All type of notice needs to print and hang on notice board where many of the tenant see it after 3-4 days. In the manual system apartment owner needs many employees and hustles to insure apartment surveillance, collect rents and bills, giving notice etc. So we made this concept with all these solutions.

1.3 Objective

There are three types of users of the platform. The Operator (the care taker of the apartment), Tenant (a person who rent a house or flat from an Apartment owner) and the Apartment owner (a man/woman who rents out an apartment, building, or accommodation). So there will be 3 different tools in our system. A web portal for the Operator, a mobile application for Tenant and a mobile application for the Apartment Owner.

Apartment owner: The owner of an apartment [Apartment owner] will be able to add and collect bills and add/remove tenant and his family members, add notice for all tenant. All records of the tenant will be shown on apartment owner noticeboard and admin dashboard. Apartment owner can check any tenant profile, records and their payment history. Owner can view the complains from the tenant through the system. Owner will have the webcam access and gate access. He will collect all types of bills through the system. Moreover, after getting a payment from the Tenant, the Apartment owner can see the payment and approved and also can store all the transaction history of payment of a Tenant.

Tenant: There will be a profile of tenant with house member, guest, deliveryman, workers, vehicles information and records. All the member will have a separate QR or PIN to enter into the apartment. Tenant can reject their entry or exit from his app. Tenant can monitor all his member's activities. When the will enter into the apartment the record will be showed to the dashboard. Tenant will have an access to see web cam footage when any unknown person trying to enter into the apartment by calling flat no from the keypad attached on the main gate. Tenant can pay their rent using the system, also can complain for any issue they faced. In this tenant can allow any guest or delivery boy to enter into apartment and set a time for leave, if they don't leave the system will notify the tenant. For paying bills tenant can view all the bills list and pay bills separately to the Apartment owner using this app.

Operator: The care taker of an apartment will manage and operate whole system. He can add any tenant after getting the information from the Apartment owner and store it to the system and give then a unique QR code for open the main gate. The Operator will also open and close the main gate when QR code will successfully scanned by the Tenant and display the results on the portal of operator. The main function of the Operator is to add, remove and update tenant information, track records, post notice and open/close the gate.

1.4 Expected Outcome

Guardio will integrate the benefits of a physical guards with the convenience of a 'no-physical guard' smart gate system by QR scan, take record who enter into the apartment for how long,

manage the house rent collection problem, remainder any notice from the owner, complain any issue. It's a hustle free smart and secure apartment surveillance system. So it will so much helpful for the Apartment owner Dhaka city to insure their apartment safety and security and saves a lot of expense and time. Tenant also will feel safe and secure in that apartment.

1.5 Project Management and Finance

We have our business plan with our honorable supervisor and co-supervisor. After fully completing our project we will focus on collecting governmental funds for Guardio. We will also look for investor in our project. After making a good capital we will start our journey. As our system is subscription based, we will take 1000 BDT per month from every apartment want use our system.

1.6 Report Layout

We have six chapters in our report, including an introduction, background information, requirements and design specifications, implementation, and conclusion and features scope. The first chapter of my project's report describes the report format. In chapter two, we mostly discussed the overall context of my project and the accompanying activities. Comparative research, the intricacy of the problem, and the difficulties are relevant to my project. The design requirement, the logical data model, and the description are all covered in addition to the business process model, requirement collecting and analysis, and use case modeling. The main focus of chapter four is a detailed specification of my project, which covers front end design, back end design, interaction design, UX requirements, and implementation requirements. In chapter five, I went into detail about the implementation and testing phases, including the database implementation, the front-end design implementation, and the test outcome and report. I discuss the app's conclusion and future plan in chapter six.

CHAPTER 2

Background

2.1 Preliminaries/Terminologies

Guardio system has 2 part and web application software and mobile app. At first we've made a 54 page SRS then we've made 3 different UI/UX design for the Operator, Tenant and Apartment owner. We have developed the web application in raw code using PHP and MySQL. We also used HTML, CSS, JavaScript and Bootstrap for the design according to our UI/UX design. After completing the Web part, we start working on Tenant mobile application. We've used Android Studio to develop our app. We've used XML for the frontend and Java for the backend. For the Apartment owner app, we've followed the same. We have 3 different applications for the system but the database is same. So we have to make API for the apps to insert and retrieve data from the server. It's take much time to develop the web application and setup the database. After completing database and the API it takes short time to complete our android application.

2.2 Related Work

We have found My Gate app on the Playstore that giving service on India. They have made the gate system automated for entry and exit from the apartment. In Bangladesh there is government funding project Rokkhi. Their system is working for visitor management. All visitor will go to the manager and they fill up entry information by the manager and apartment owner can allow and reject their entry to the apartment. Then we have found another app name Voban where an apartment owner can manage flat rent and bills.

2.3 Comparative Analysis

My Gate app has only focused on the gate entry and exit. Rokkhi is working for visitor management only. All visitor will go to the manager and they fill up entry information on the app which takes lots of time. Voban app is for the management of tenant where an apartment owner can manage flat rent and bills. None of the system has as much as features that we have in our Guardio system has. We have automated QR scan for the entry and exit and a tenant app for tenant management where tenant can add family members, pay bills, view notice and an apartment app

to manage apartment's tenant, send notice, collect bills. In our system only the people have the QR/PIN from the Apartment owner and Tenant can enter and exit from the apartment.

2.4 Scope of the Problems

Due of our system's limited cloud storage and bandwidth, we won't be able to handle heavy user traffic right away. When more than 2000 users are using our system simultaneously, the server may go down and an application not responding (ANR) error may result. The java null pointer exception is a common programmatic issue in the creation of apps. Runtime Exceptions can include Null Pointer Exceptions. Common null data are frequently assigned to an item reference in Java. When a program tries to use an item reference that refers to null data, Null Pointer Exception is thrown. As a result, some features might not operate.

As our system only give permission enter the apartment after a valid QR scan so both Tenant and Apartment owner have to keep their own QR private otherwise anyone can use their QR to enter to the apartment. Also they should block the unnecessary members from the app otherwise they can enter anytime without permission.

2.5 Challenges

We have the most challenging part in integrating one database to all application. As we use firebase and SQLite for mobile app development most of the time. We also face challenges to insure all features for the system. That's making SRS takes too long time. Then we have made UI/UX design on Adobe XD tool. Which isn't free anymore. So we have to trial version. But we can generate only one shareable link in trial so we used 3 different email for 3 design. On development we have added some new features that causes change to database again and again. On app development we also added some extra feature as well, for that we had to make changes on Web application code from the C panel live server which was so risky.

CHAPTER 3

Requirement Specification

3.1 Business Process Model

In order to spot potential changes, business process modeling is the graphical representation of a company's business processes or workflows. This is frequently accomplished using several graphing techniques, such as the flowchart, data-flow diagram, etc. The two states of the process are mapped using BP modeling: As-is, the state of the technique as it currently exists, without any adjustments or improvements, and To-be, the longer-term state, following the implementation of the changes or improvements.

In figure 3.1, this a flowchart of Operator web application business process model. Here we have showed the process how operator will perform the operator web application.

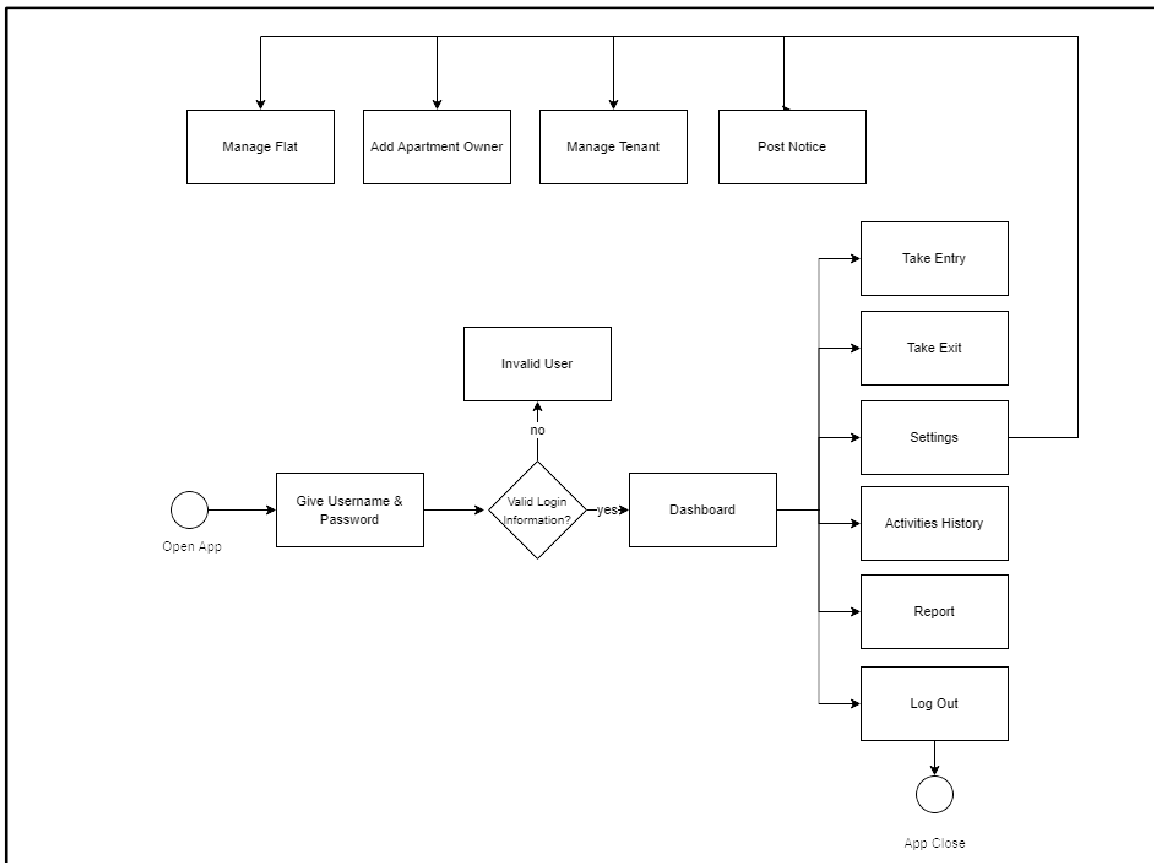


Figure 3.1: Operator System Business Process Model

In figure 3.2, this a flowchart of Apartment owner app business process model. Here we have showed the process how apartment owner will perform the Apartment owner mobile app.

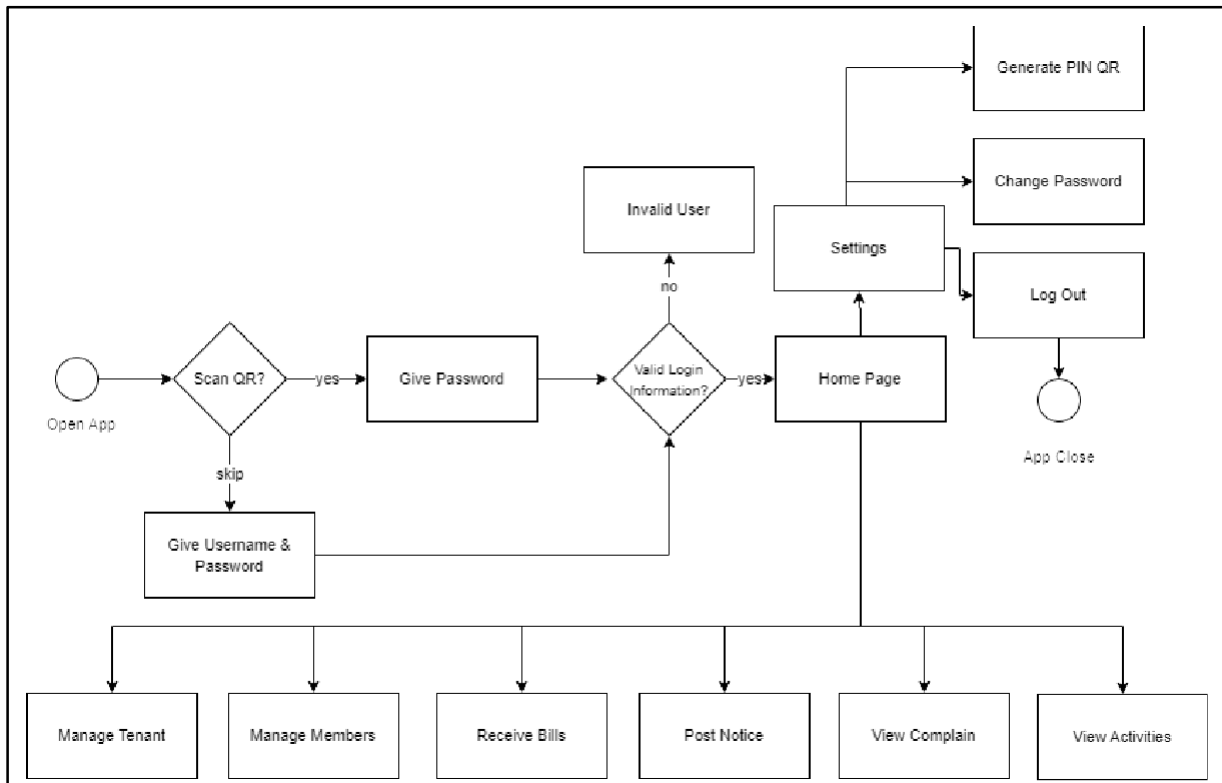


Figure 3.2: Apartment Owner System Business Process Model

In figure 3.3, this a flowchart of Tenant app business process model. Here we have showed the process how tenant will perform the Tenant mobile app.

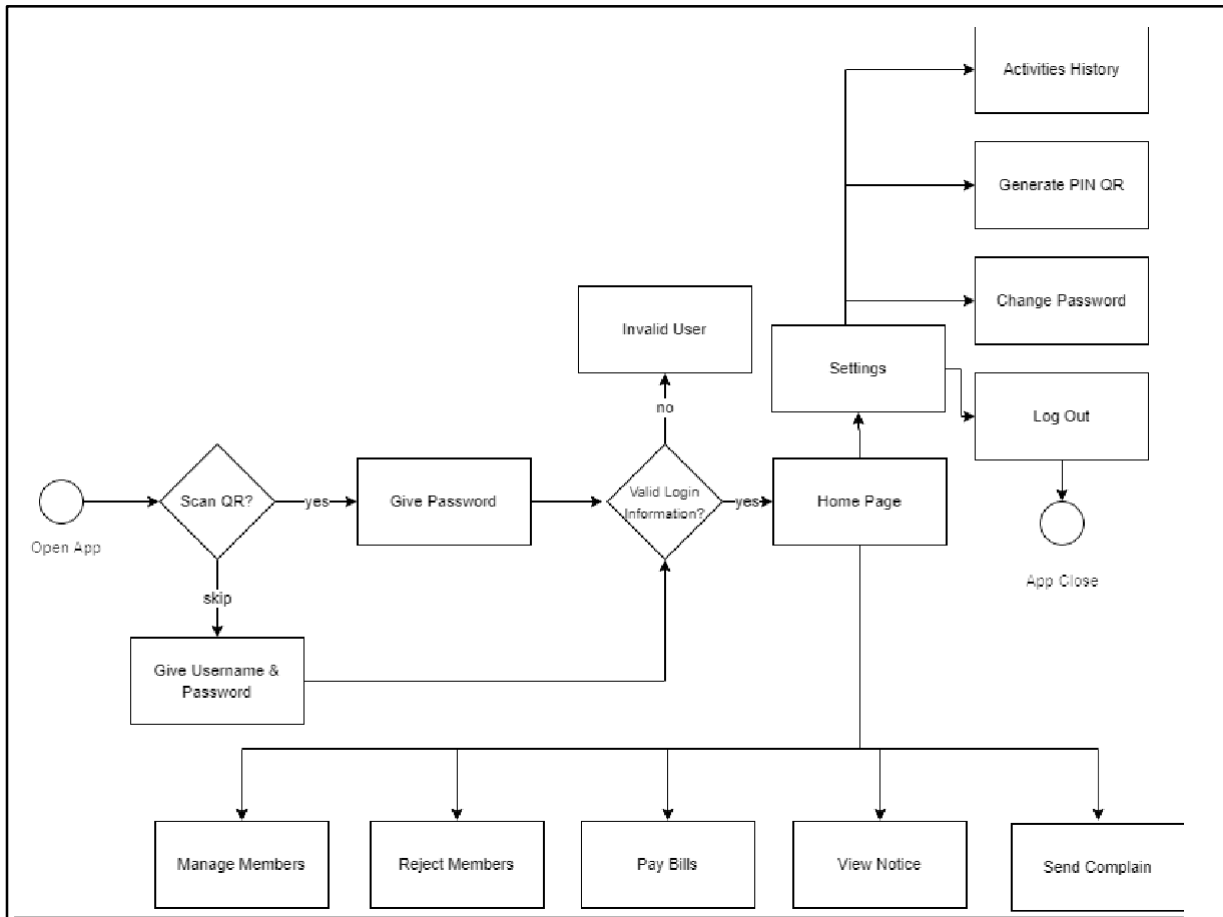


Figure 3.3: Tenant System Business Process Model

Block Diagram

A block diagram is a specific flowchart used in designing to imagine a system at a significant level. It is used to plan new systems or to explain and improve existing ones. We have showed three of our application flowchart here. The block diagram showing how our system working with database and when it taking data from database and when it inserting data into the database. We have also showed others functionality of the applications.

In figure 3.4, this a block diagram of Operator web application. Here we have showed the process how operator will login to the system, how operator will manage the system and when operator will send and collect data from the database.

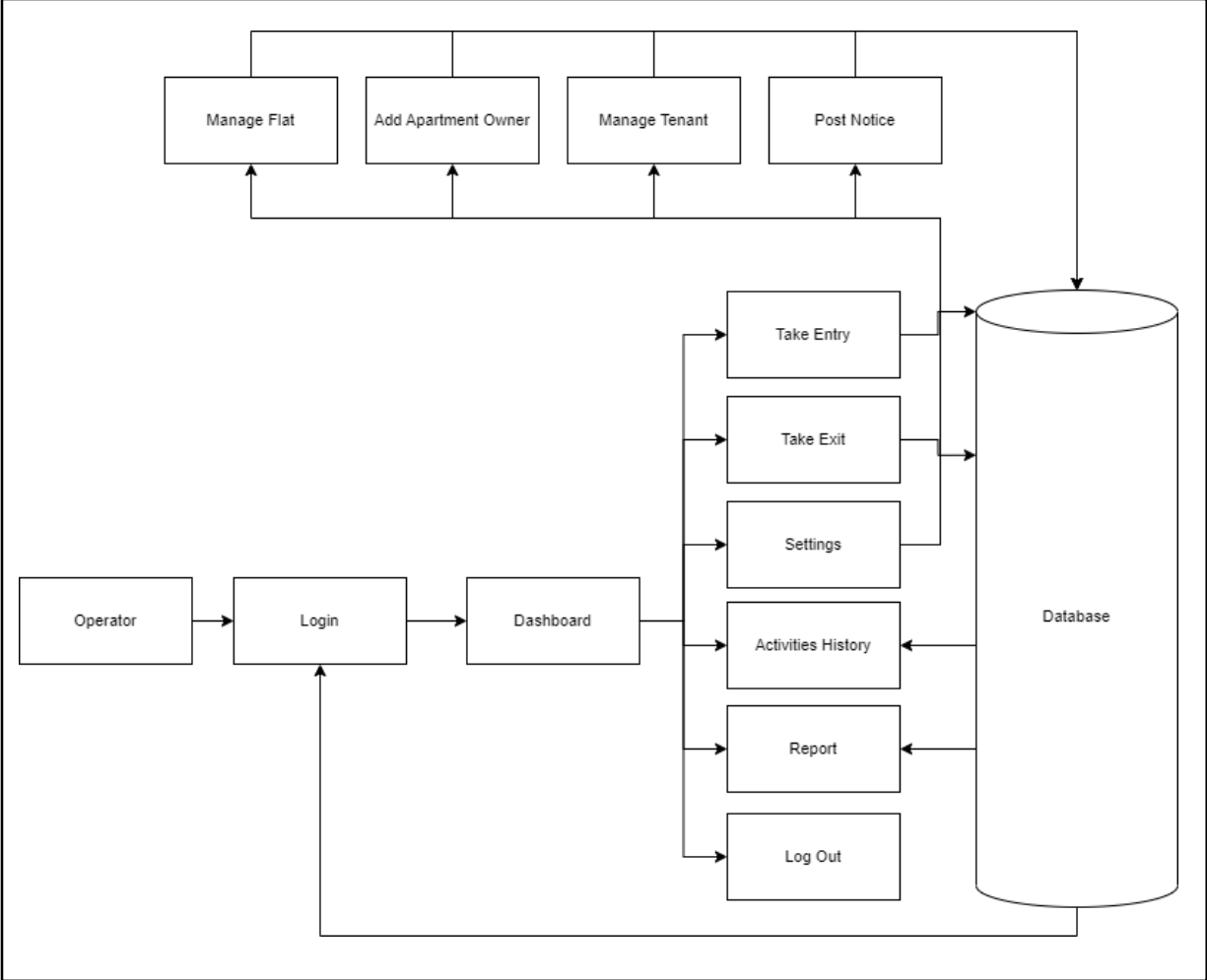


Figure 3.4: Operator System Block Diagram

In figure 3.5, this a block diagram of Apartment owner mobile app. Here we have showed the process how apartment owner will login to the app, how apartment owner will manage the app and when apartment owner will send and collect data from the database.

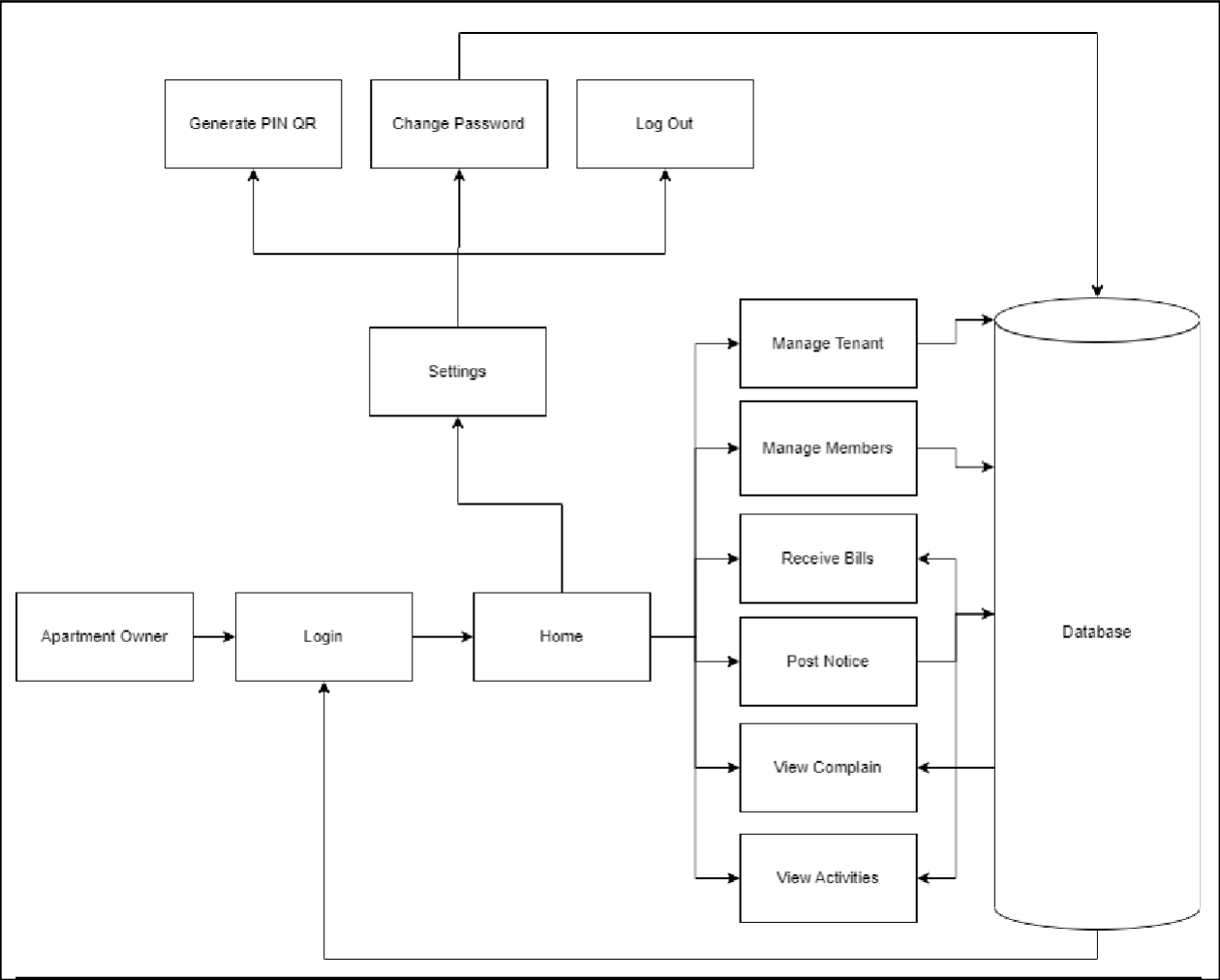


Figure 3.5: Apartment Owner System Block Diagram

In figure 3.6, this a block diagram of Tenant mobile app. Here we have showed the process how tenant will login to the app, how tenant will manage the app and when tenant will send and collect data from the database.

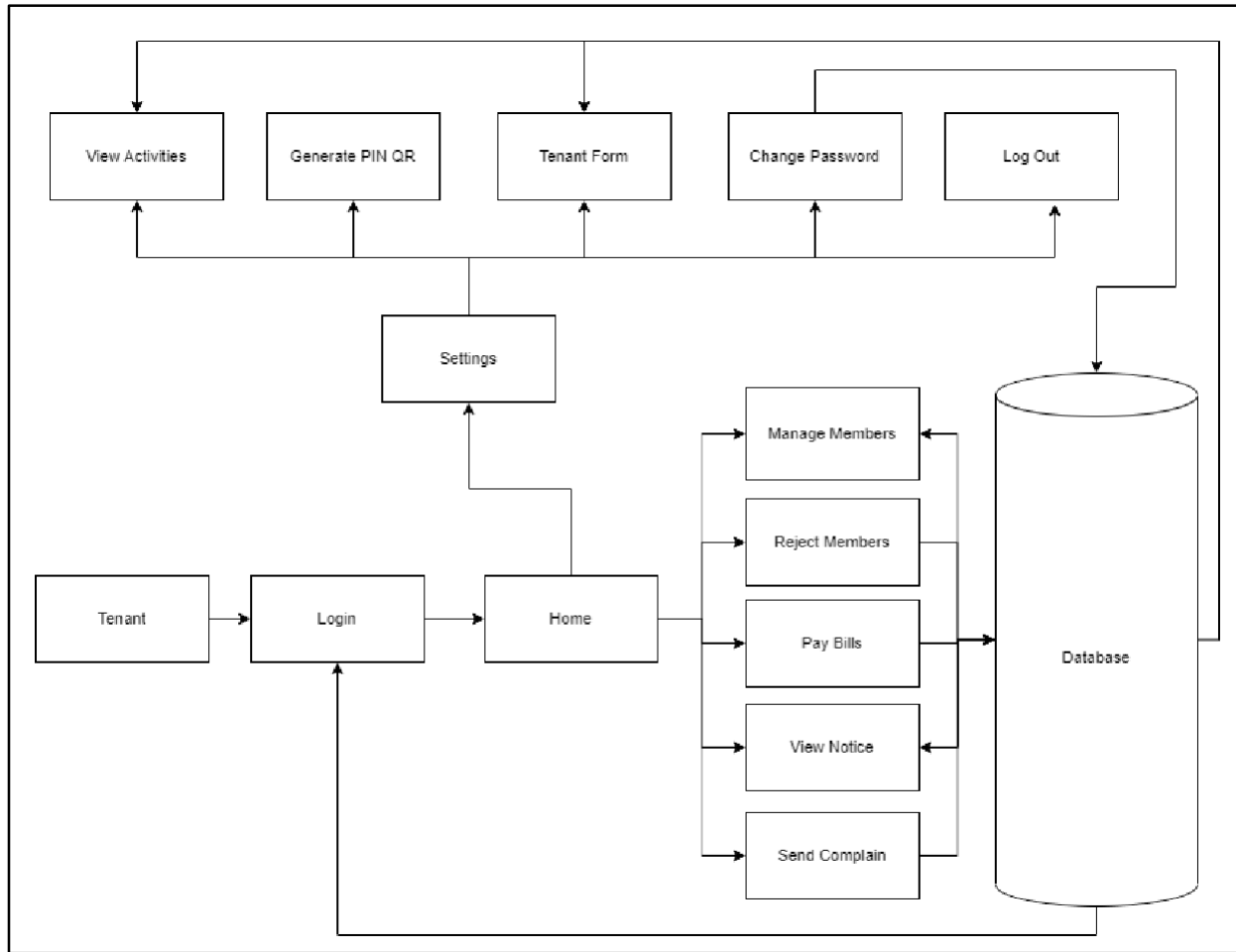


Figure 3.6: Tenant System Block Diagram

3.2 Requirement Collection and Analysis

The process of identifying user requirements for brand-new or modified applications is called requirement engineering, also referred to as requirement analysis. In the context of software engineering, it is frequently referred to as requirements gathering or requirements capture. Requirements analysis encompasses all of the activities involved in identifying the requirements or conditions that must be fulfilled for a new or modified product or project, taking into account

the potentially conflicting requirements of different stakeholders, and analyzing, documenting, validating, and managing software or system requirements. We also gathered the needs of a few apartment owners, spoke with the manager, and contrasted them with the Bangladeshi system that is now in place.

Mobile gadgets are becoming more and more common due to their portability and practical applications. Today, corporations as well as individuals use portable computers for a variety of minor and large jobs. On the other side, the popularity of mobile users and programmers is rising due to the quick uptake of mobile technology. Developers aim to build a platform that can be utilized on a number of gadgets, such as smartphones, tablets, laptops, and PCs, while building a mobile application. The most crucial analysis for developing an app is this one. You must conduct a feasibility analysis of our project concept after selecting an idea. That is, does it solve a problem for the user with this idea, is it unique, is it appropriate for the target audience, is the app stable and functional, and is your service practical for the user? An unbiased evaluation of an app design to determine viability and merit is known as a feasibility analysis. We conducted a feasibility analysis before implementing our plan. When we first started looking online, I discovered that two systems shared a few functions but had very distinct concepts.

3.3 Use Case Modeling and Description

A use-case model is a condensed illustration of how particular user types interact with a technology to address a challenge. The most crucial model elements are usage cases, actors, and their interactions. A portion of the model is visualized using a use-case diagram to facilitate communication. We displayed a use-case graphic for three of our applications that included all of their features.

In figure 3.7, this is a use-case model of Operator web application. Here we have showed all functionality of the operator in our operator web application.

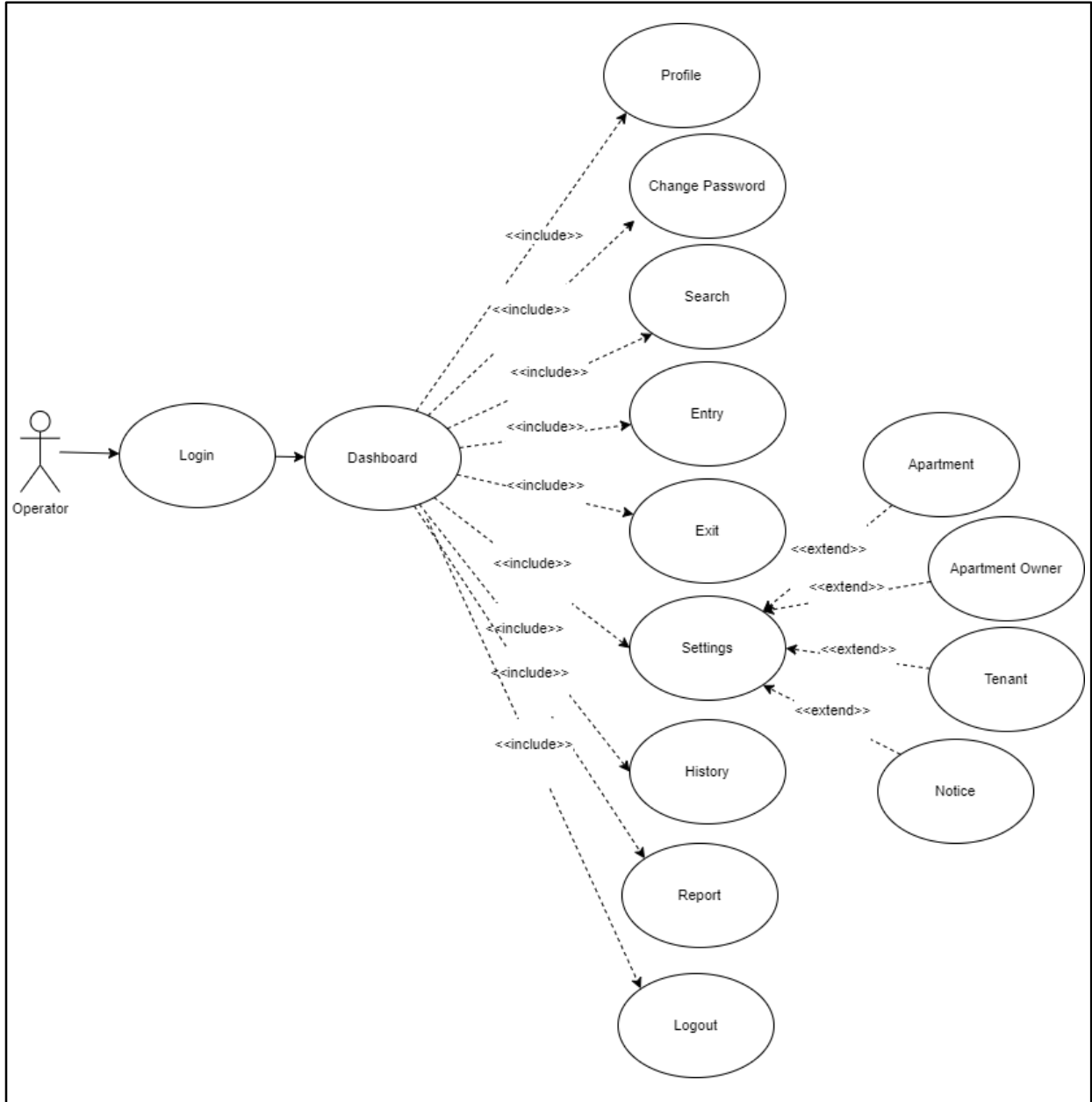


Figure 3.7: Operator System Use Case Model

In figure 3.8, this is a use-case model of Apartment owner app. Here we have showed all functionality of the apartment owner mobile app.

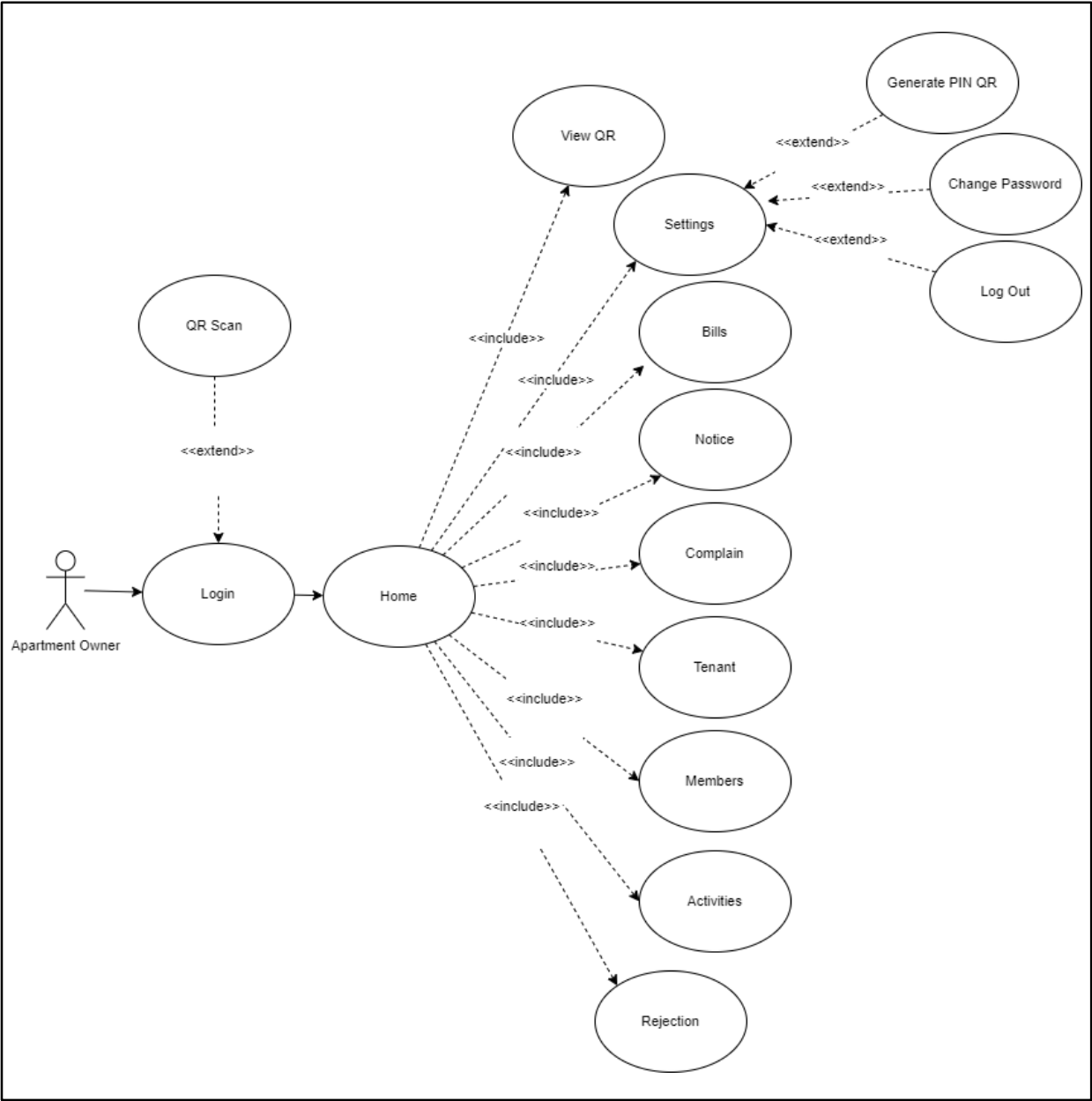
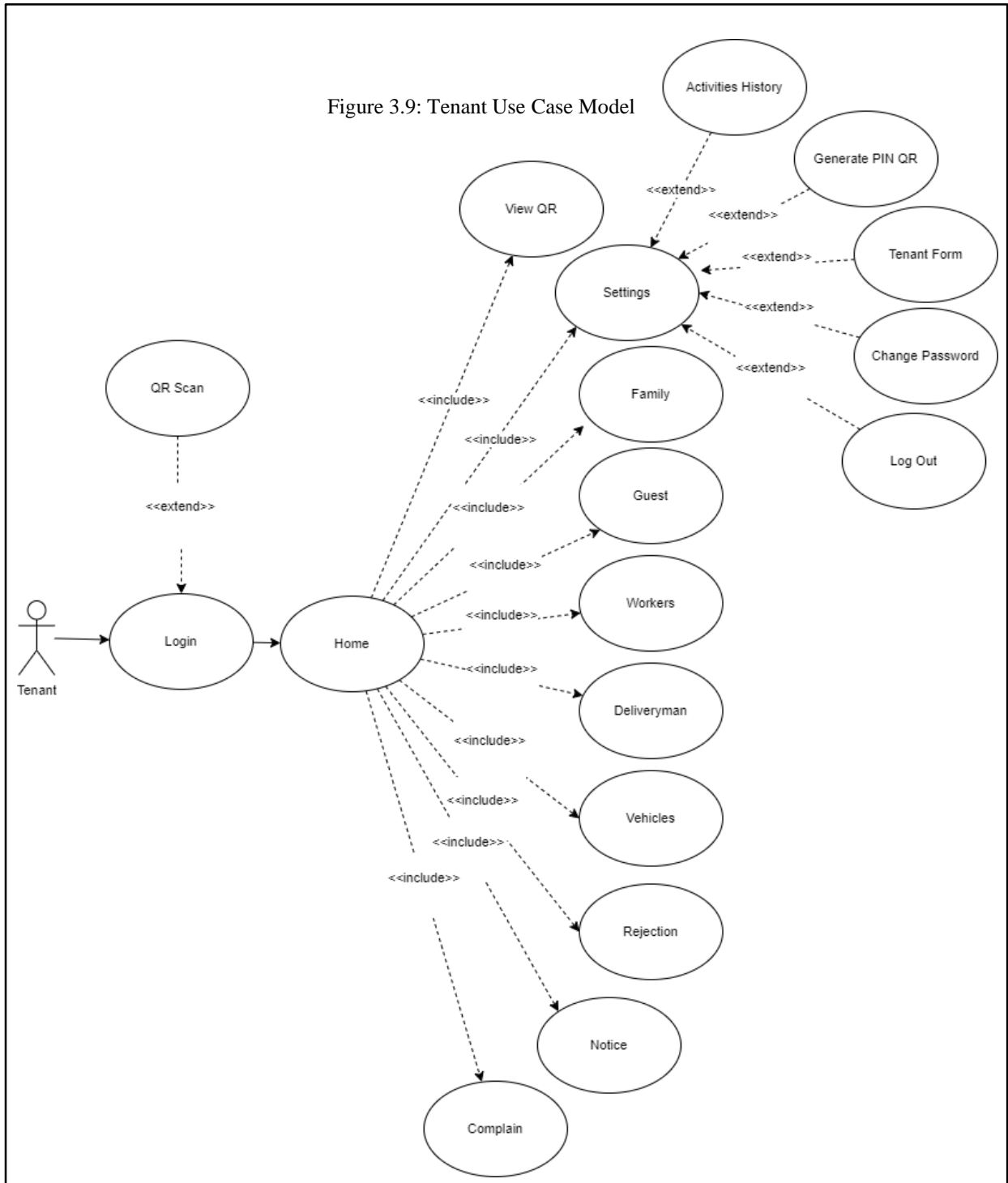


Figure 3.8: Apartment Owner Use Case Model

In figure 3.9, this is a use-case model of Tenant app. Here we have showed all functionality of the tenant mobile app.



3.4 Logical Data Model

In order to spot potential changes, business process modeling is the graphical representation of a company's business processes or workflows. This is frequently accomplished using several graphing techniques, such as the flowchart, data-flow diagram, etc. BP modeling is used to map 2 distinct process states: Without any modifications or enhancements, the approach is used in its current form. After implementing the enhancements or modifications, the longer-term state. Here we have showed our system logical data model in figure 3.10

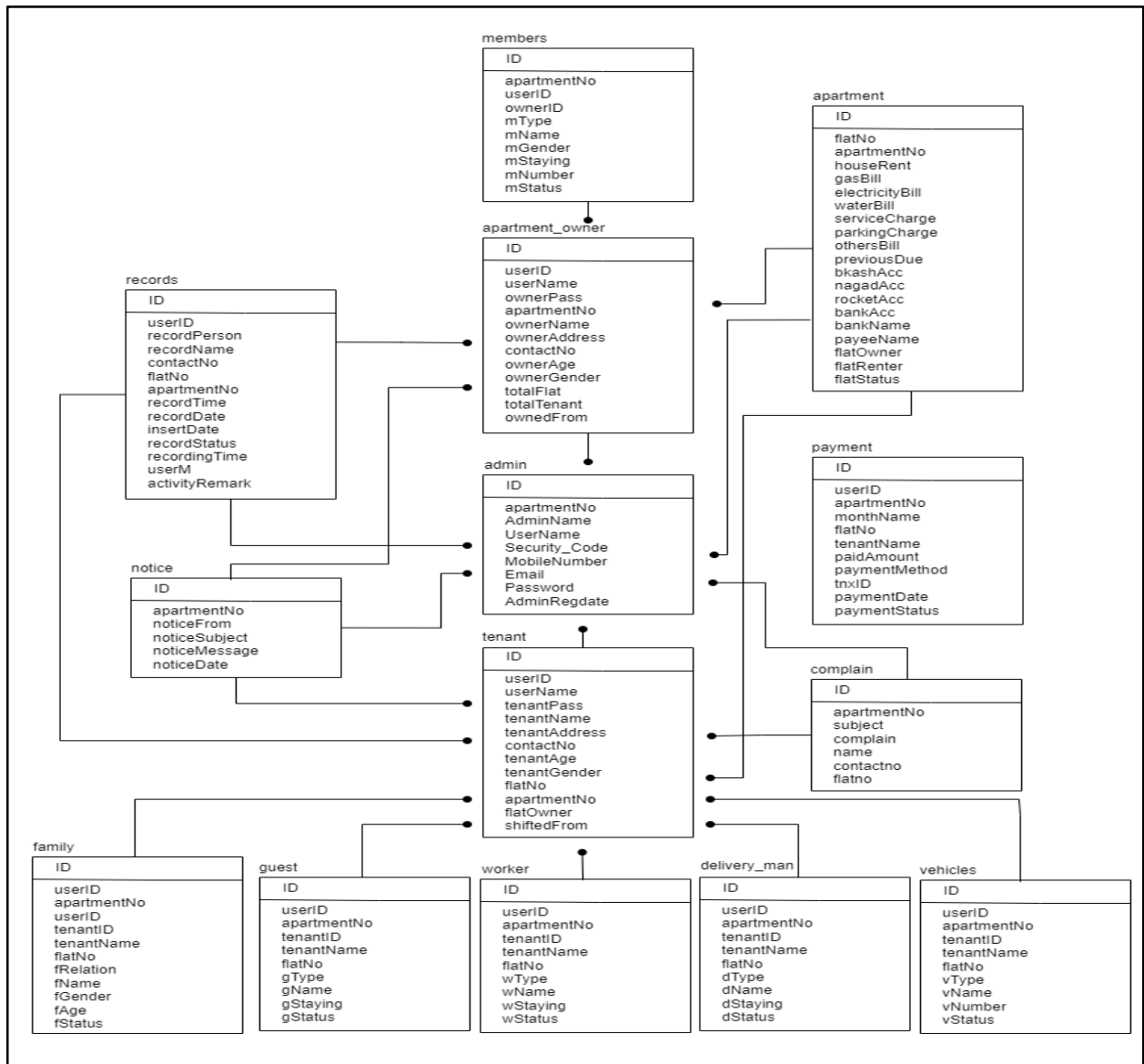


Figure 3.10: Logical Data Model

3.5 Design Requirements

The first stage in designing an application is defining the requirements. Since the design of the requirements affects every step, Some pertinent requirements are as follows:

- Operator Dashboard
- Operator Access & Option
- Operator Navigation Menu
- Apartment Owner App Homepage
- Apartment Owner Navigation Menu
- Apartment Owner App Design
- Apartment Owner Access & Options
- Tenant App Homepage
- Tenant Navigation Menu
- Tenant App Design
- Tenant Access & Options

CHAPTER 4

Design Specification

4.1 Frontend Design

In web application design for the operator, we have used html, css, javascript and bootstrap. In mobile app for both apartment owner and tenant we've used java and xml in Android Studio framework.

Operator Web Portal:

In figure 4.1, showing the entry functionality where every members of the apartment will scan their QR to give entry record and get permit to enter to the apartment.

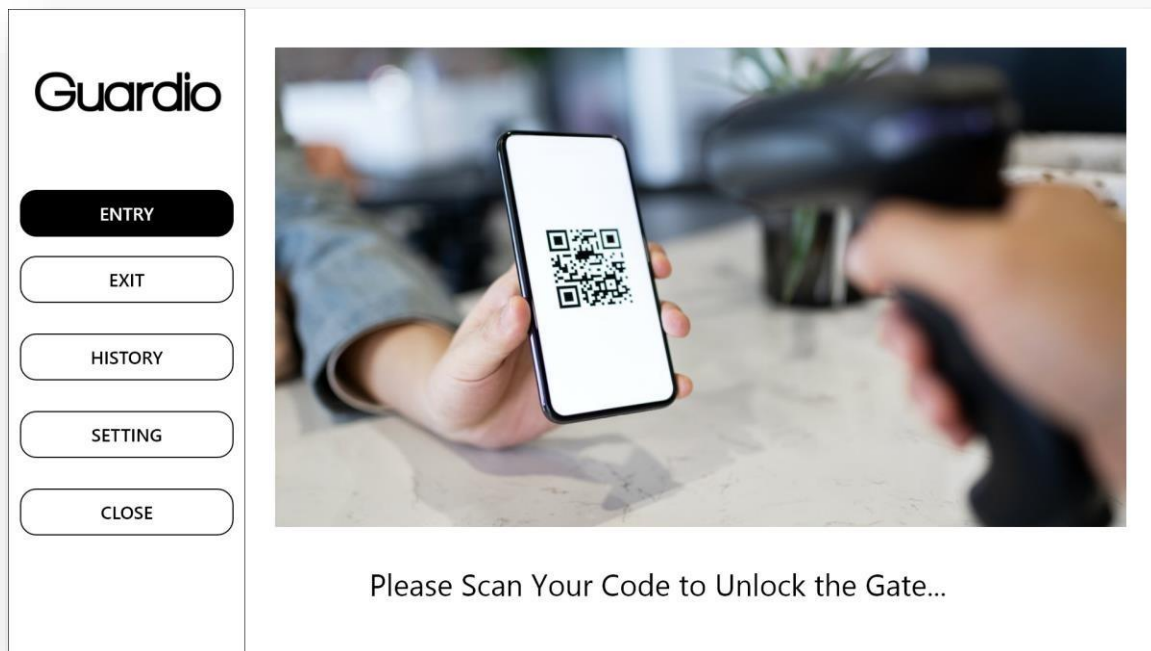


Figure 4.1: Entry Function

In figure 4.2, showing the exit functionality where every members of the apartment will scan their QR to give exit record and get permit to exit from the apartment.

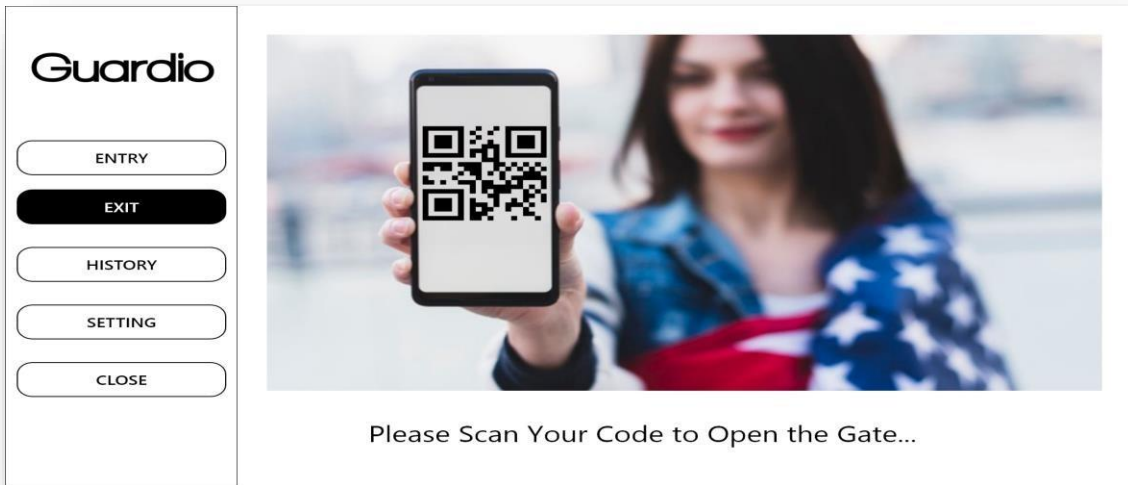


Figure 4.2: Exit Function

In figure 4.3, showing the entry and exit history of the all apartment members, workers, guest and visitors.

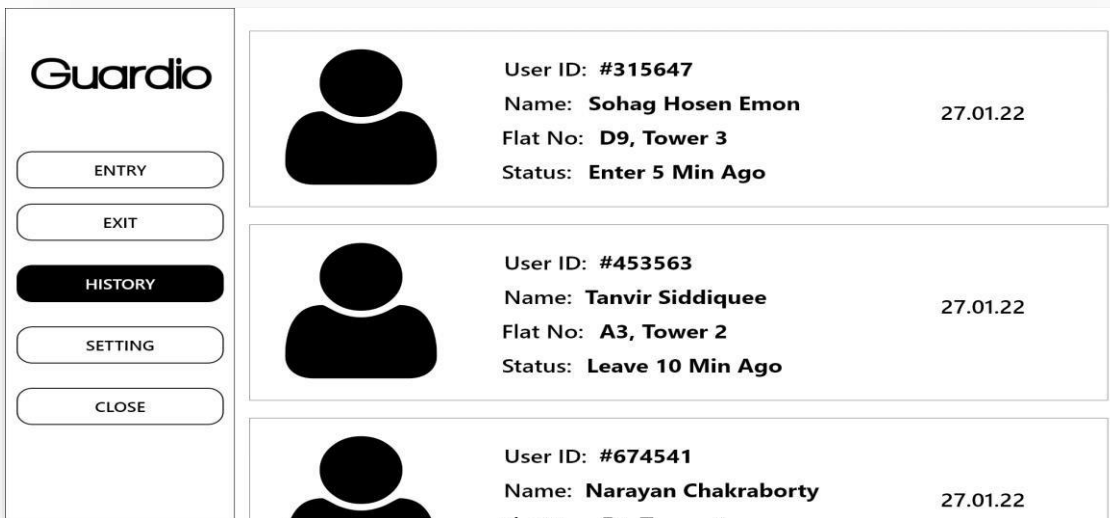


Figure 4.3: History

In figure 4.4, showing the setting functionality and the setup options for the operator



Figure 4.4: Setting

In figure 4.5, showing the add members option for the operator

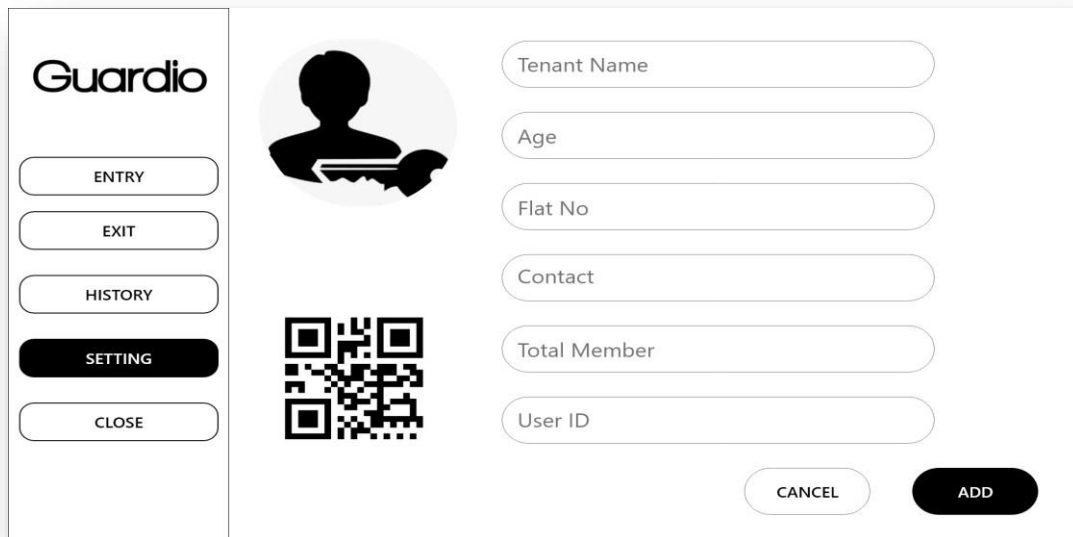
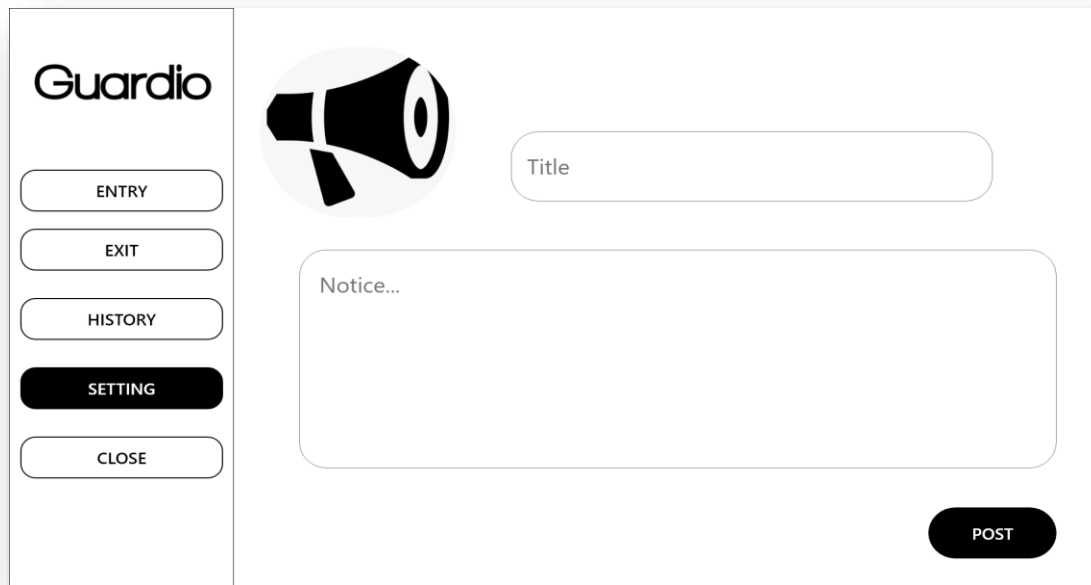


Figure 4.5: Add Member

In figure 4.6, showing the notice section for the operator. From this section an operator can send any notice to the all members of the apartment.



The screenshot displays the 'Post Notice' interface within the Guardio application. On the left side, there is a vertical sidebar with the 'Guardio' logo at the top. Below the logo, there are five buttons: 'ENTRY', 'EXIT', 'HISTORY', 'SETTING' (which is highlighted in black), and 'CLOSE'. The main content area on the right features a megaphone icon in a circle. To the right of the icon is a text input field labeled 'Title'. Below the 'Title' field is a larger text area labeled 'Notice...'. At the bottom right of the main content area, there is a black button labeled 'POST'.

Figure 4.6: Post Notice

Apartment Owner App:

In figure 4.7, this is the splash screen for the apartment owner mobile app

In figure 4.8, this is the homepage for the apartment owner mobile app.



Figure 4.7: Owner Splash Screen

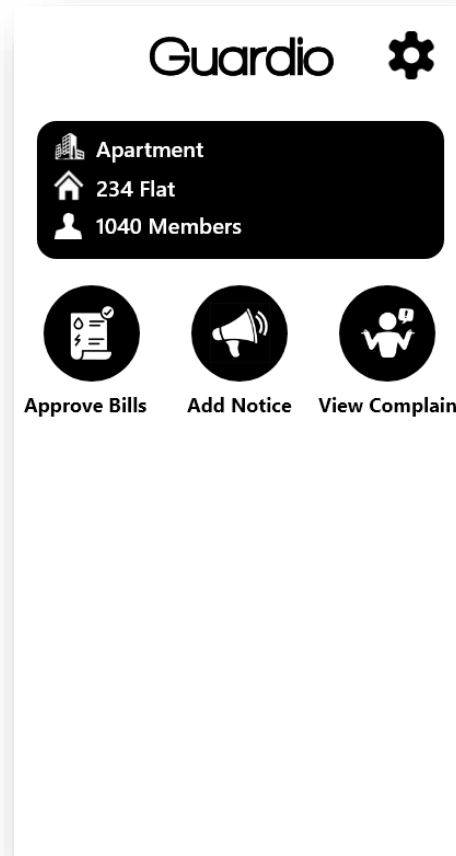


Figure 4.8: Owner Homepage

In figure 4.9, this is the navigation menu for the apartment owner mobile app

In figure 4.10, this is the bills section for the apartment owner mobile app.

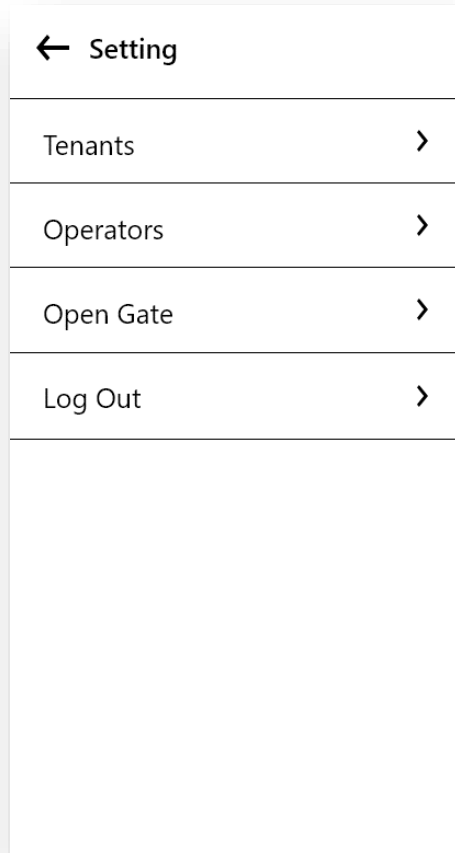


Figure 4.9: Owner Navigation Menu



Figure 4.10: Owner Bills Section

In figure 4.11, this is add notice for the apartment owner mobile app

In figure 4.12, this is the view complain section for the apartment owner mobile app.

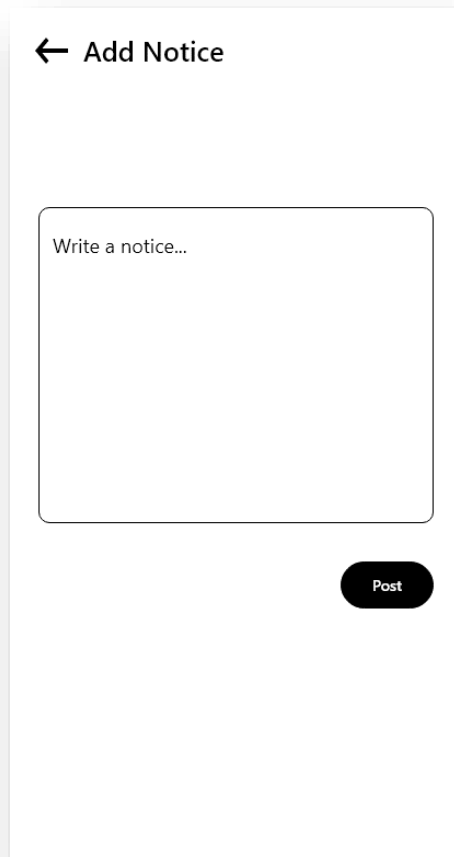


Figure 4.11: Owner Add Notice Section

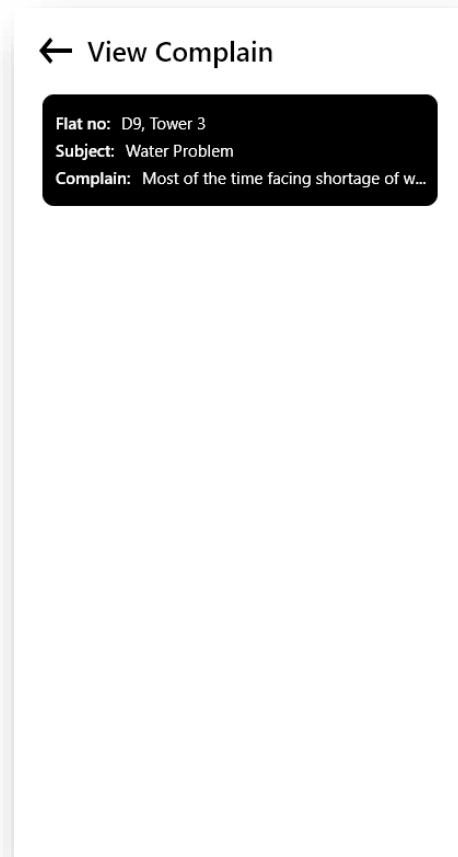


Figure 4.12: Owner Complain View Section

Tenant App:

In figure 4.13, this is splash screen for the tenant mobile app

In figure 4.14, this is the join option for the tenant mobile app.



Figure 4.13: Tenant Splash Screen

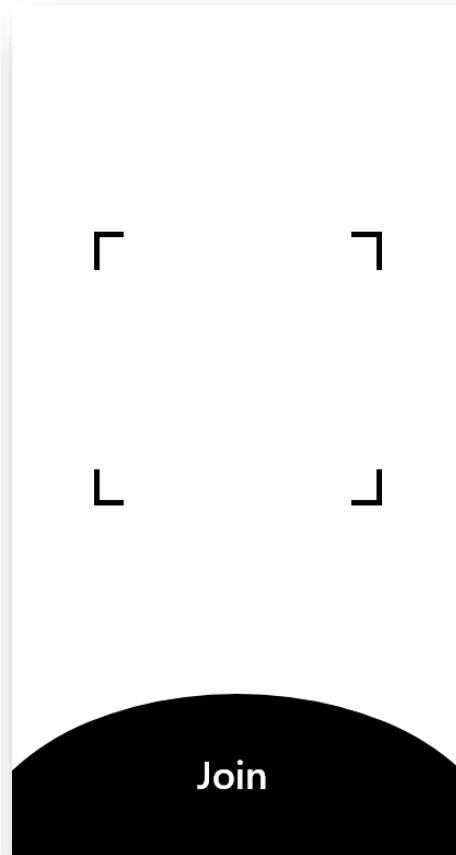


Figure 4.14: Tenant Join Option

In figure 4.15, this is homepage for the tenant mobile app

In figure 4.16, this is navigation menu for the tenant mobile app.



Figure 4.15: Tenant Homepage

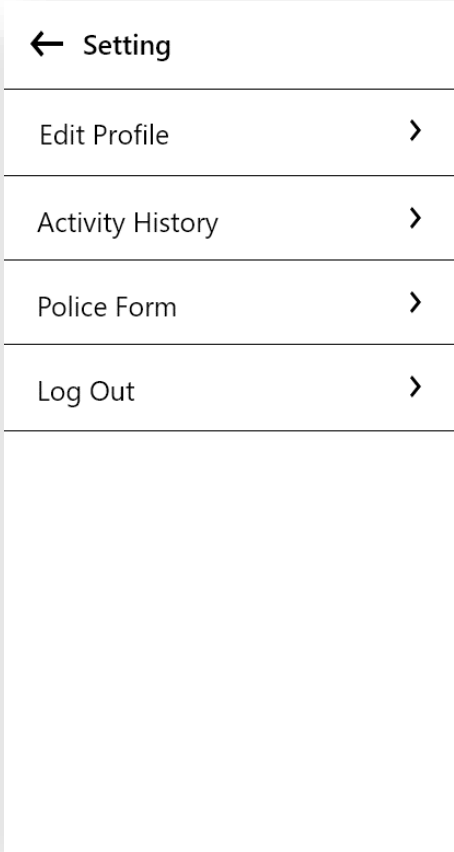


Figure 4.16: Tenant Navigation Menu

In figure 4.17, this is managing family section for the tenant mobile app

In figure 4.18, this is guest managing section for the tenant mobile app.

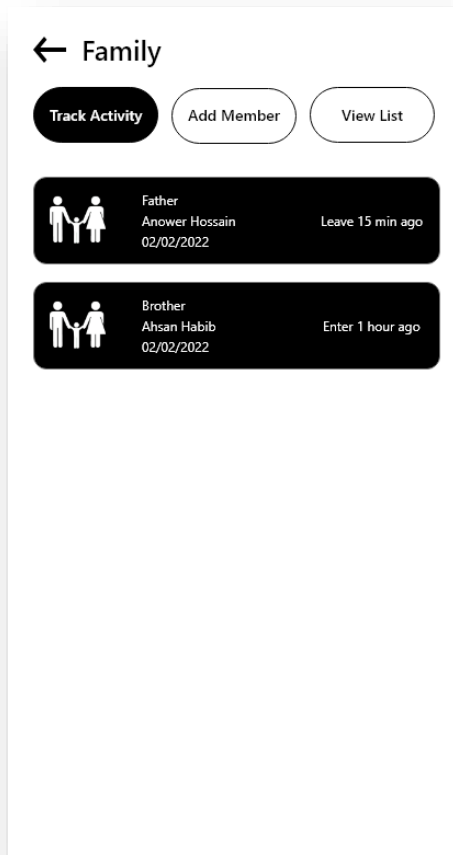


Figure 4.17: Tenant Family Section

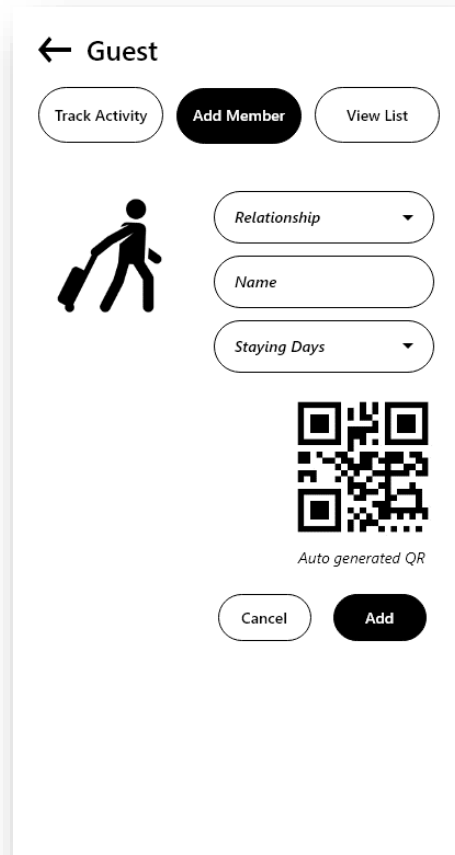


Figure 4.18: Tenant Guest Section

In figure 4.19, this is managing workers section for the tenant mobile app

In figure 4.20, this is bills section for the tenant mobile app.



Figure 4.19: Tenant Workers Section

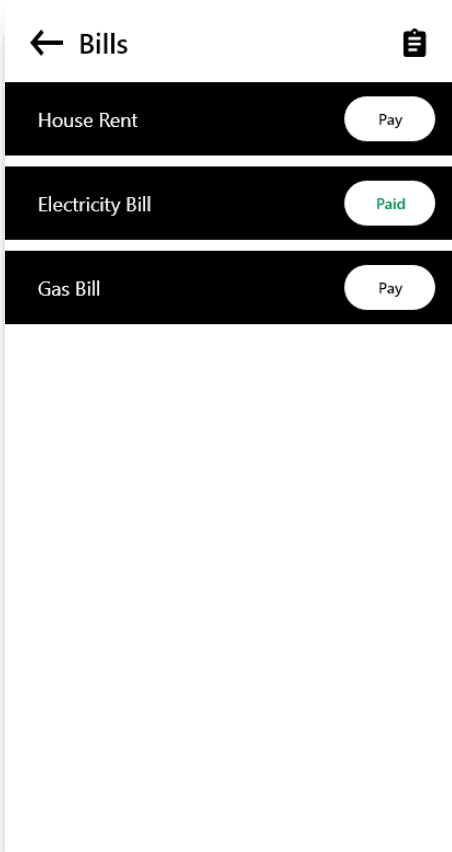


Figure 4.20: Tenant Bills Section

In figure 4.17, this is notice view page for the tenant mobile app

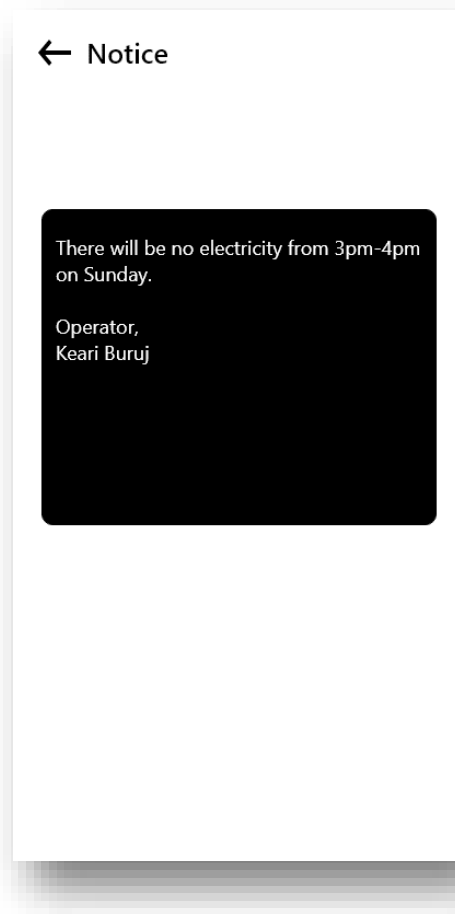


Figure 4.21: Tenant Notice View

4.2 Backend Design

In web application backend for the operator, we have used php and mysql. In mobile app for both apartment owner and tenant we've used API java using API from the server.

In figure 4.22, showing the web application files in php which are working as backend for the operator web application.

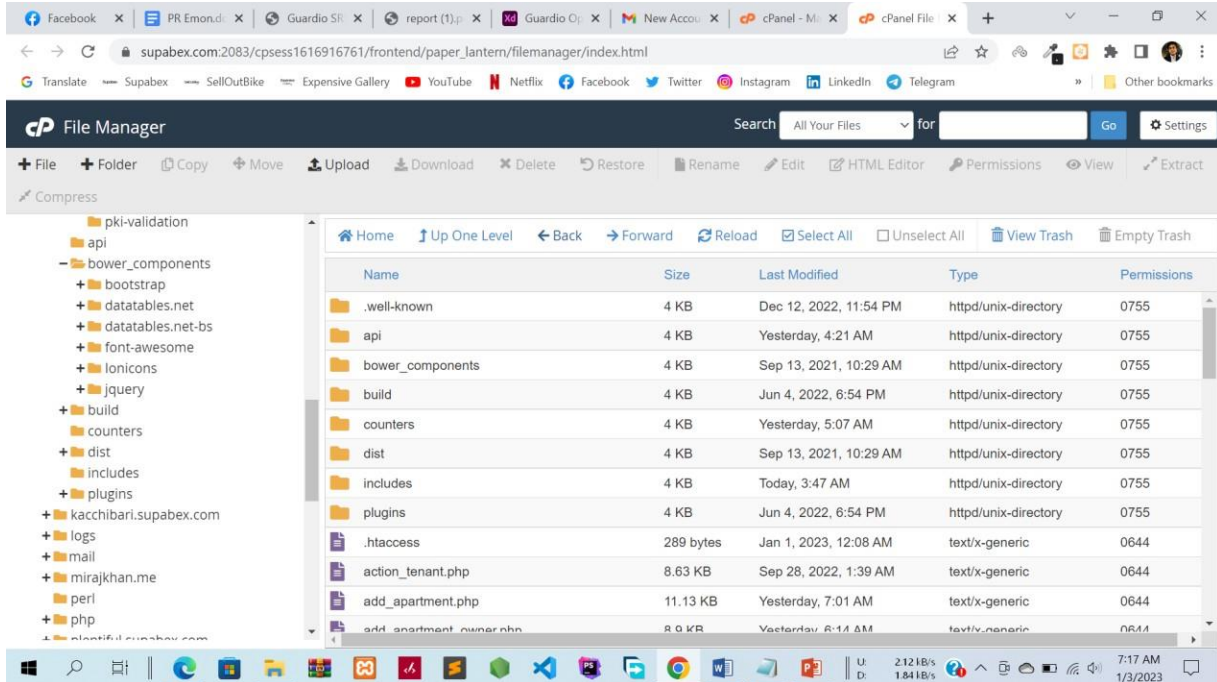


Figure 4.22: Web Application Backend Files

In figure 4.23, showing the API files which is used as the backend of both apartment owner and tenant mobile app.

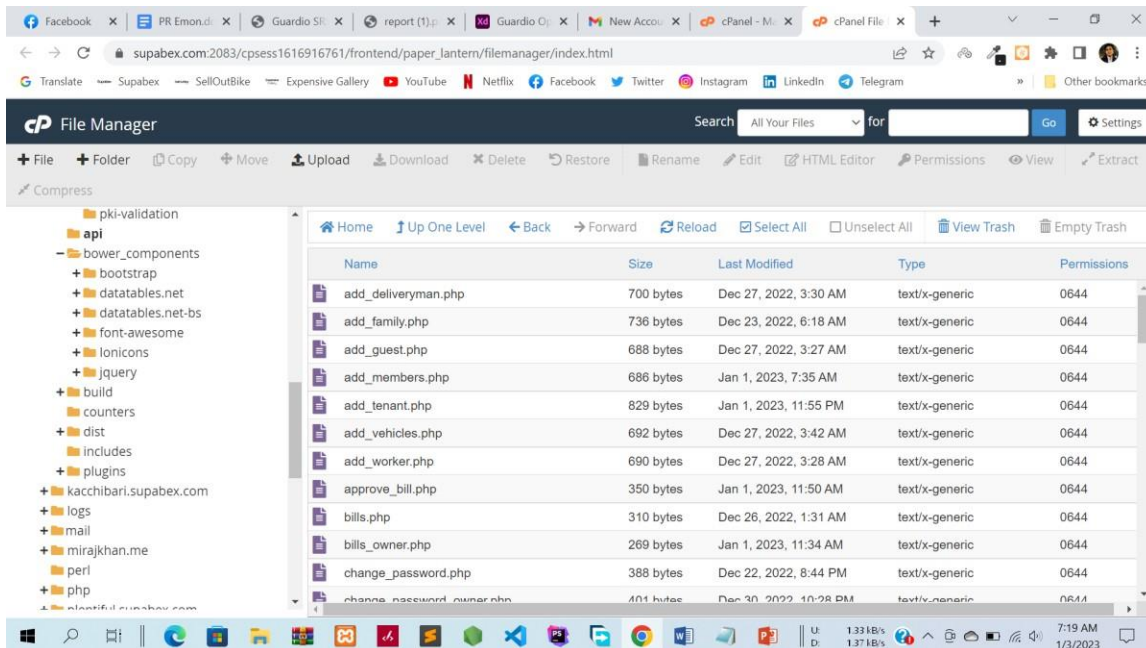


Figure 4.23: API Files

In figure 2.24, showing the mysql database used in the whole system backend.

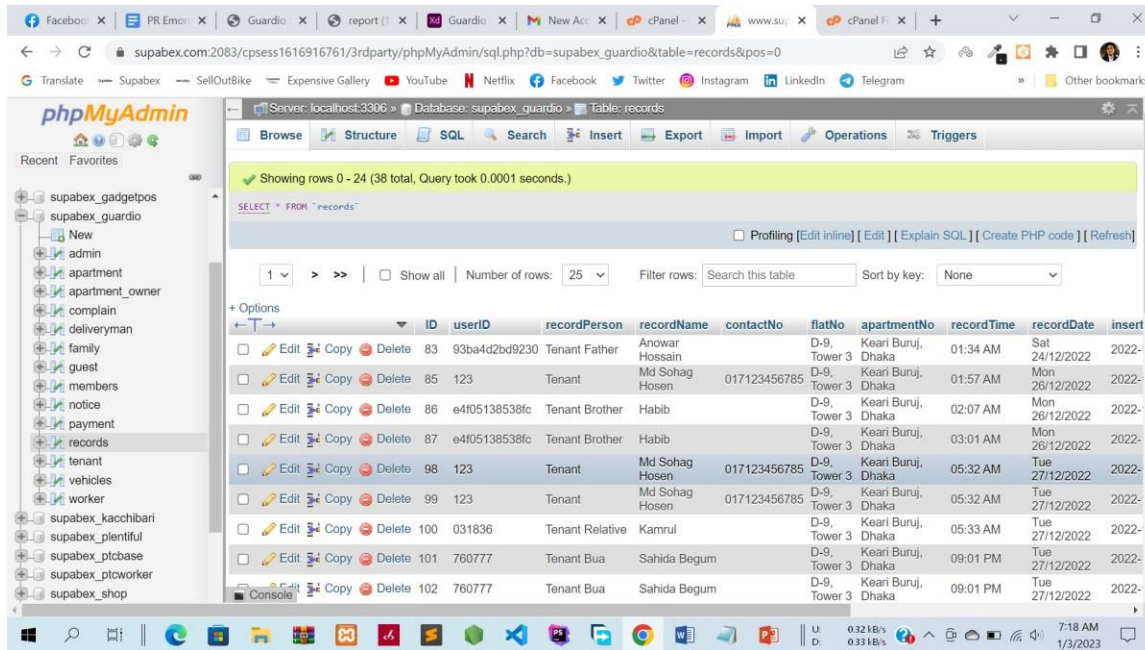


Figure 4.24: Database

4.3 Interaction Design and User Experience (UX)

A basic idea is interaction design, which is the method for bringing the customer and the material into contact. People typically think about programming when they consider collaboration plans for things like software or websites. Understanding the client's problem, measuring it and figuring out the typical consequence, taking action based on the outcomes, and finally permanently fixing the problem are all steps in the cycle of interaction design. The goal of the intuitive plan is to produce products that assist clients in achieving their goals as quickly and effectively as feasible. We designed our system to be user-friendly for each of the three different user types. It would ensure improved user experience and better engagement with our system.

4.4 Implementation Requirements

We required Android Studio, a Google open source platform, as well as a programming language like Java, PHP, or MYSQL in order to develop this system. In order to enforce all specifications, a third-party API or library is necessary.

Java

The bulk of native app developers design their apps in the java programming language. In addition, a variety of languages, including Dart, Flutter, and Kotlin, can be used to create apps. Our software for apartment owners and tenants uses Java. An object-oriented language is the Java programming language. The Android SDK and Java programming language are the recommended approaches for Android devices.

PHP

PHP is the most common language for web development. There are many benefits of PHP for the web application. We have used PHP for the Operator web portal. PHP work smooth and fast. We can easily connect database and insert data to the database and retrieve from database.

MYSQL

On the web server, the database is MYSQL, and PHP is used to retrieve data from the database. Basic parameters from our application are sent to the PHP page, which then contacts the MYSQL database, retrieves the results, and sends them back to us. PHP is also used to get a record from the MySQL database when it is created.

Google Material Design

Designing using material for mobile apps. It is a plan language developed by Google that, at its core, is a very intricate and well-defined collection of guidelines to aid in the imitation of Google's work by both producers and end users as well as to explain why things in Google behave in the manner they do. However, we used Google Material Design in our mobile app for the content spacing, font size, text color, margin, grids, scale active icon, app color, and other elements.

CHAPTER 5

Implementation and Testing

5.1 Implementation of Database

Databases keep all kinds of information so that ID and power can be shown on a straightforward stage. The client can easily, quickly, and adaptably access data thanks to the information base. A chart is made via the analysis of relative records. We must first put together the table's layout, which comprises the table name, table section names, unfamiliar key, and important key, before planning and storing data.

I utilized MySQL in my software. The database is created and maintained using this information base management system. It offers a methodical way of recovering, creating, enhancing, and removing data.

5.2 Implementation of Frontend Design

The front-end procedure is crucial for mobile applications since users frequently engage with them. The backend code won't be visible to them. No matter how well written your code is, if the design is poor, people won't utilize your program. Responsive design, which assures that the design would seem the same on all phone display sizes, is the most potent aspect of front-end design. To make our web application slick and easy to use, we use HTML, CSS, JavaScript, and Bootstrap in our web portal. We employ user constraints in our mobile app to keep the UI as straightforward as possible. This design is responsive to all screen sizes and is mobile-friendly. We occasionally utilize a linear style with a weight total, which makes the design more sensitive.

5.3 Testing Implementation

Based on user activity, the test Case table of Guardio user system features

Test Case	Test input	Expected output	Actual Output	Result	Tested On
Database Connection	Test in Guardio Web Application and Apps	Successfully Connected	Connection Successful	Passed	27-12-2022
Create	Test in Guardio Web	Successfully	Created	Passed	27-12-2022

Account	Application and Apps Insert API	Created	Successful		
Login	Test in Guardio Apartment Owner and Tenant App Login API	Successfully Login	Login Successful	Passed	27-12-2022
Entry & Exit	Test in Guardio Web Application Insert QR	Will scan and keep person record	Scanned and Recorded	Passed	27-12-2022
Add Members	Test in Guardio Apartment Owner and Tenant App Insert Data	Members will be added	Members added Successfully	Passed	27-12-2022
Rejection	Test in Guardio Apartment Owner and Tenant App	Member will be rejected	Members Rejected	Passed	27-12-2022
Bills Pay	Test in Guardio Tenant App	Bills will be paid	Bills paid Successfully	Passed	27-12-2022
View Notice	Test in Guardio Tenant App	Notice will show	Notice Showed	Passed	27-12-2022
Send Complain	Test in Guardio Tenant App	Complain will be sent	Complain sent Successfully	Passed	27-12-2022

Table 5.1: Test Case Result

5.4 Testing Results and Reports

The test report must reflect testing results in a formal manner, providing an opportunity to assess testing outcomes immediately. All three applications provided 100% correct and anticipated outcomes for our testing. We were highly motivated to move forward with this project after reading the testing report.

CHAPTER 6

Impact on society, Environment and Sustainability.

6.1 Impact on Society

Guardio system a surveillance system for an apartment so it will be help for our society that illegal and unexpected issues for Apartment owner and Tenant will be reduced. The children and the women will be safe and secure in their apartment. Violence rate in housing society will be below average.

6.2 Impact on Environment

In this system everything will be controlled by device and internet. Where device like Computer, Laptop, QR Scanner and Phones will be used and all of them is electrical devices. So no pollution will happen here.

6.3 Ethical Aspect

Our system will maintain multiple security in Apartment. This system will not use for hampering other's personal information. Here we've everything recorded on our system and we can monitor it 24 hours, so there is very low chance to happened any type of unethical incident in our house like steal, robbery etc.

6.4 Sustainability Plan

Whole system will be controlled by devices and internet, so until devices and internet connection is ok our system will run properly without any interruption.

CHAPTER 7

Conclusion and Future Scope

7.1 Conclusion

Guardio system store all types of data of the apartment and works like a surveillance system for the apartment. This system is going to make a huge change on the apartment management system. It will keep the apartment safe and secure for living. The apartment owner will be benefited mostly. All management of an apartment owner will come easier. Tenant also will be helpful from our system. They can track records of their family members, guest, workers from outside of home. As we aimed to do business with this system after completing graduation we will be more focused on the project and improve the system quality and insure safety for the people living in the apartment where our Guardio system is installed.

7.2 Future Scope

Right now our system is semi-automated. We must need an operator to scan the QR and take entry and exit records. In our system we've plan to user Smart door at the gate with QR scanner. When no operator will be required to the system and our system will be fully automated. In the future we also have a plan replace QR with facial recognition.

Reference:

- [1] MyGate, available at << <https://mygate.com> >>, last accessed on 28-06-2022 at 9:00 PM.
- [2] Material Design, available at << <https://material.io/> >>, last accessed on 26-07-2022 at 7:00 PM.
- [3] Rokkhi, available at << <https://rokkhi.com/> >>, last accessed on 28-08-2022 at 3:00 PM.
- [4] Stack Overflow, available at << <https://stackoverflow.com/> >>, last accessed on 05-09-2022 at 4:00 PM.
- [5] Google Play, available at << <https://play.google.com/store/apps/> >>, last accessed on 21-09-2022 at 2:00 PM.
- [6] WebHoastBd, available at << <https://www.webhostbd.com/> >>, last accessed on 03-10-2022 at 3:00 PM.
- [7] W3school, available at << <https://www.w3schools.com/> >>, last accessed on 28-12-2022 at 3:00 PM.
- [8] Vobon, available at << [https:// https://play.google.com/store/apps/details?id=com.vobon.app](https://play.google.com/store/apps/details?id=com.vobon.app) >>, last accessed on 28-12-2022 at 3:00 PM.

Thank You!
Best of Luck

Final Project Report

ORIGINALITY REPORT

27 %	24 %	1 %	19 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	16 %
2	Submitted to Daffodil International University Student Paper	6 %
3	Submitted to University of Liberal Arts Bangladesh Student Paper	1 %
4	Submitted to Red & Yellow Creative School of Business Student Paper	1 %
5	Submitted to Jacksonville University Student Paper	1 %
6	Submitted to Cyryx College, Maldives Student Paper	1 %
7	Submitted to Midlands State University Student Paper	<1 %
8	Submitted to University of Salford Student Paper	

<1%

9

Submitted to Laureate Education Inc.

Student Paper

<1%

Submitted to Universitas Sanata Dharma
Student Paper

<1%

11

www.coursehero.com
Internet Source

<1%

Exclude quotes On

Exclude matches < 4 words

Exclude bibliography On