Project Documentation

Of

**DEVELOPMENT OF**

**SOCIETY MANAGEMENT SYSTEM**

**(MyGate)**



Submitted by

 21605 Aniruddha Patil

21624- Sanjay Gupta

   21636- Sachin Ambekar

 Under the guidance of

**Dr. Santosh Kumar Singh**

Submitted in partially fulfillment of requirement for Qualifying BSc-(IT) Semester VI Examination

Department of Information Technology Academic Year: **2021-2022**



**Certificate**

This is to certify that the Project entitled

**DEVELOPMENT OF**

**SOCIETY MANAGEMENT SYSTEM**

**(MyGate)**

Submitted by

**Mr. Aniruddha Kiran Patil (21605)**

In partial fulfillment for BSc.IT Degree (Semester V) Examination for the academic year 2021-2022 and has not been submitted for any other examination and does not form a part of any other course undergone by the candidate. It is further certified that he/she has completed all the required phases of the project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head Of Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Guide



**Certificate**

This is to certify that the Project entitled

**DEVELOPMENT OF**

**SOCIETY MANAGEMENT SYSTEM**

**(MyGate)**

Submitted by

**Mr. Sanjay Ashok Gupta(21624)**

In partial fulfillment for BSc.IT Degree (Semester V) Examination for the academic year 2021-2022 and has not been submitted for any other examination and does not form a part of any other course undergone by the candidate. It is further certified that he/she has completed all the required phases of the project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head Of Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Guide



**Certificate**

This is to certify that the Project entitled

**DEVELOPMENT OF**

**SOCIETY MANAGEMENT SYSTEM**

**(MyGate)**

Submitted by

**Mr. Sachin Subhash Ambekar (21636)**

In partial fulfillment for BSc.IT Degree (Semester V) Examination for the academic year 2021-2022 and has not been submitted for any other examination and does not form a part of any other course undergone by the candidate. It is further certified that he/she has completed all the required phases of the project.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Internal Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

External Examiner

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head Of Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Guide

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| SR NO | TOPIC | PAGE NO |
| 1 | Introduction   1. Objective and scope of the project 2. Theoretical background 3. Problem definition 4. User requirements 5. Feasibility Study 6. Details of hardware and software used |  |
| 2 | System analysis and design   1. Detailed life cycle of the project 2. Context Diagram 3. DFD Diagram 4. ERD Diagram 5. Class Diagram, 6. State Transition Diagram 7. Use Case Diagram 8. Activity Diagram 9. Component Diagram 10. Collaboration Diagram 11. Deployment Diagram 12. Architecture Diagram |  |
| 3 | System Planning   1. Gantt Chart 2. Pert Chart 3. Project Scheduling |  |
| 4 | System Implementation (Screen Shots) |  |
| 5 | Cost and Benefits analysis and Software parameter Estimation |  |
| 6 | Future Work |  |
| 7 | Conclusion |  |
| 8 | Bibliography/References |  |

1. **INTRODUCTION**

In society all jobs are decided in meetings and maintenance bills, contact no of members are noted in the papers. There is no automated system for doing all the things that generally happen in society so that members can come to know what is happening in society

This system of maintaining a society is made in such a way so that the most common problem faced by residential societies are solved. This system is a cloud-based system to manage the day - to - day activities of any cooperative housing society. They need the coordination between the respective management societies coupled with the vendors which provide these services so that the appropriate convenience can be provided. The main functionality of this project is that we have included three Login panels that includes residential login, staff login & Committee Login there is notice board system, complaint section, visitor/guest section, visitor management, society dues, parking and helpdesk.an online billing and accounting (payment gateway integration, income & expense tracking, etc.). In this system, the bills, receipts, and vouchers are created in an easy manner also the system is user-friendly

1. **Objective and Scope of the project**

**Objectives:**

The aim of our project is to ease the work of Society using the technology in such a way that:

* Manage Society administrative tasks effectively.
* Reach maximum people at the same time.
* Parking Space allocation.
* To notify important issues through notice board.

**Scope:**

Following functionalities are in the scope of system:

* Allowing members to view and manage the dues.
* Allowing members to view notices with the help of digital notice boards.
* Allowing members to Track Deliveries sent to the society Gate

1. **Theoretical Background**

**Module Description**

**a. Visitor Management**

This allows the staff members to keep record of the visitors which will also help in maintaining the security of the society by ensuring that each and every person visiting will tracked down in the system

1. **Parcel Delivery**

This allows the staff members to keep track of society member’s delivery

This step will also ensure that no wrong person is entering into the premises with any false intensions.

1. **Parking Allotment**

This feature will allot the parking space to the members and users can choose the parking slot they need.

1. **Notice/Events**

By using the Login as Member option, it will allow the members of the society to read the notice or any information related to any event. Similarly, by using the Login as Committee option, it will allow the committee members to create/delete or modify the notices or any information related to any events.

1. **Complaints**

This will allow the members to post their queries or complaints.

While committee members can view the complaints and can resolve their complaints.

## Development Environment

**Visual Studio IDE:**

* Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.
* Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists

**Flutter:**

* Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia and the web from a single codebase
* Flutter apps are written in the Dart language and make use of many of the language's more advanced features.
* On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just in Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state

**Language Used: Dart**

* Dart is a client-optimized language for developing fast apps on any platform. Its goal is to offer the most productive programming language for multi-platform development, paired with a flexible execution runtime platform for app frameworks.
* The Dart language is type safe; it uses static type checking to ensure that a variable’s value always matches the variable’s static type. Sometimes, this is referred to as sound typing. Although types are mandatory, type annotations are optional because of type inference. The Dart typing system is also flexible, allowing the use of a dynamic type combined with runtime checks, which can be useful during experimentation or for code that needs to be especially dynamic.
* Dart offers sound null safety, meaning that values can’t be null unless you say they can be. With sound null safety, Dart can protect you from null exceptions at runtime through static code analysis. Unlike many other null-safe languages, when Dart determines that a variable is non-nullable, that variable is always non-nullable. If you inspect your running code in the debugger, you’ll see that non-nullability is retained at runtime (hence sound null safety)

**Database Used: Back4app**

* Back4app is one of the most amazing open source backend as a Service provider
* Backend as a Service is a latest addition in this field. Backend as a Service also known as BaaS solution are leveraging developers to backend services without maintaining or building them. Back4app is one of the most amazing Backend as a Service solutions which is offering an extensive range of features and benefits to its users. Know everything about back4app in details below. This will certainly let you have a better idea about this application development solution in a more effective way.

1. **Problem Definition**
   1. Our proposed system is cloud based for storing and accessing data
   2. Housing society management system allows societies to create user accounts for each member in three ways. I.e., Committee Residential and Staff. This System allows members to login with their own account and get updated with society happenings.
   3. Using the housing society management system, Committee Members can publish Notices to notice boards and to owners at the click of a button.
   4. This software concentrates mainly on Digitization of Society Works whereas our system consists of various other features of society which are exact replicas of the real happenings in the society.
   5. We present the Society Management project. Society Management is an open Roof to Root System that handles all the activities and logs.
   6. A society management project that effectively manages and handles all the functioning of a cooperative housing society
2. **User requirements / SRS**

### **Functional Requirements**

Functional Components of the project are:

  Login as Member:

* The society members can read the notices and events information within few minutes
* The society members can write complaints within minutes.

       Login as Committee:

* The committee members can resolve the complaints of members.
* The committee members can create notices or event information within minutes.

       Login as Staff:

* The staff members can maintain the record of number of visitors easily.
* The staff members will confirm the parcel delivery status from the member and then only allow the parcel delivery.

### **Non - Functional Requirements**

* System should be able to give response on time. Loading time of system should not be too much.
* System should be user-friendly and informative.
* This is an Online software and hence always available to the user with internet connection

### **Hardware requirements**

* **Processor:** Intel i3 equivalent or more
* **Memory**: 4 GB
* **Disk Space**: 50 GB of free disk space

### **Software Requirements**

* **Operating System** – Windows 10, Android version min: - 8.0
* **Technology** – Dart, Flutter, Back4app
* **IDE** – Visual Code, Android Studio.

1. **Feasibility Study**

The main purpose of feasibility analysis is to check the economic viability of the proposed system. The result of the feasibility study will indicate whether to proceed with the proposed system or not. If the results of the feasibility study are positive, then can proceed to develop a system otherwise project should not be pursued. Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new module and debugging old running system. All system is feasible if they are unlimited resources and infinite time. The feasibility of the system can be categorized into three types:

a. Technical Feasibility

b. Operational Feasibility

c. Economic Feasibility

**a.**     **Technical Feasibility:**

Technical feasibility emphasizes on the technology required for the proposed system and also determines how this technology can be integrated. It also involves evaluation of the hardware and the software requirements of the proposed system.

This system is developed using Flutter. The technical resources such as Visual Studio IDE, are used for developing this software and they are open source and hence easily available, so the proposed system is technically feasible.

**b.**     **Operational Feasibility:**

Proposed objects are beneficial only if they can be turned out into information system. That will meet the organizations operation requirements operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility if a project includes the following:

•        Will the system be used and work properly if it is being developed and implemented?

•       Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above mentioned issues. Beforehand, user requirements have been assumed. So there is no question of resistance from the users that can undermine the possible application benefits.

•        User need to have basic knowledge of computers.

**c.**      **Economic Feasibility:**

Economic feasibility is the cost and logistical outlook for a business project. Prior to embarking on a new venture, most businesses conduct an economic feasibility study, which is a study that analyzes data to determine whether the cost of the prospective new venture will ultimately be profitable to the company.

Proposed system requires development tools and software such as Visual Studio IDE, Flutter SDK, Android Studio AVD which are freely available. As the required hardware and software platform is already available, no investment is to be made in that direction. For developing proposed system various resources such as computer system, internet connection for Email.

**2) System analysis and design**

**A) Detailed Life cycle of the project**

To solve actual problems in an industry setting, software engineer or a team of engineers must incorporate a development strategy that encompasses the process, methods and tools and layers and generic phase. This strategy is often referred to as process model or a software engineering paradigm. A process model for software engineering is often chosen based on nature of the project and application, the methods and tools to be used, and the controls and deliverables that required.

**Types of Software Process Model:**

•        The Waterfall Model

•        The Prototyping Model

•        The Rapid Application Development (RAD) Model

•        The Incremental Model

•        The Spiral Model

•        Agile Software Model

This Application is developed using **Incremental Model.**

**Incremental model**

Incremental model the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

**Incremental Life Cycle Model**

**Advantages of Incremental model**

•        Generates working software quickly and early during the software life cycle.

•        This model is more flexible – less costly to change scope and requirements.

•        It is easier to test and debug during a smaller iteration.

•        In this model customer can respond to each built.

•        Lowers initial delivery cost.

•        Easier to manage risk because risky pieces are identified and handled during it’d iteration.

**When to use the Incremental model**

•        This model can be used when the requirements of the complete system are clearly defined and understood.

•        Major requirements must be defined; however, some details can evolve with time.

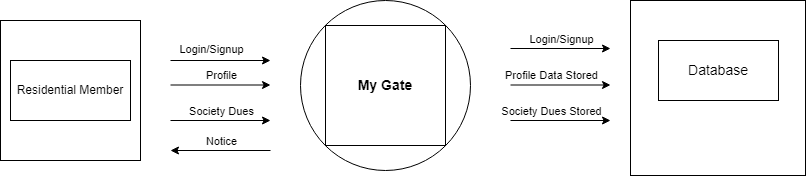
•        There is a need to get a product to the market early.

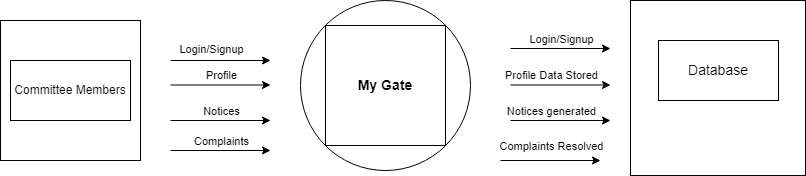
•        A new technology is being used

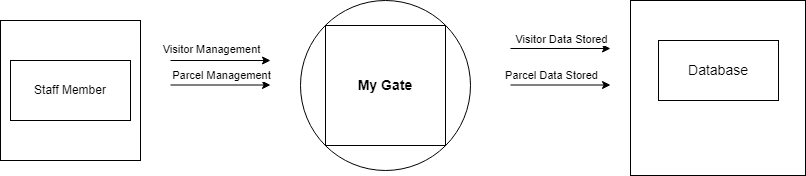
•        Resources with needed skill set are not available There are some high-risk features and goals.

**B) Context Diagram**

A context diagram, generally described as just a level 0 data-flow diagram, has been used to identify and illustrate the limits of a software system. It identifies information flows between the system and external entities. A whole software system is represented as a single process.

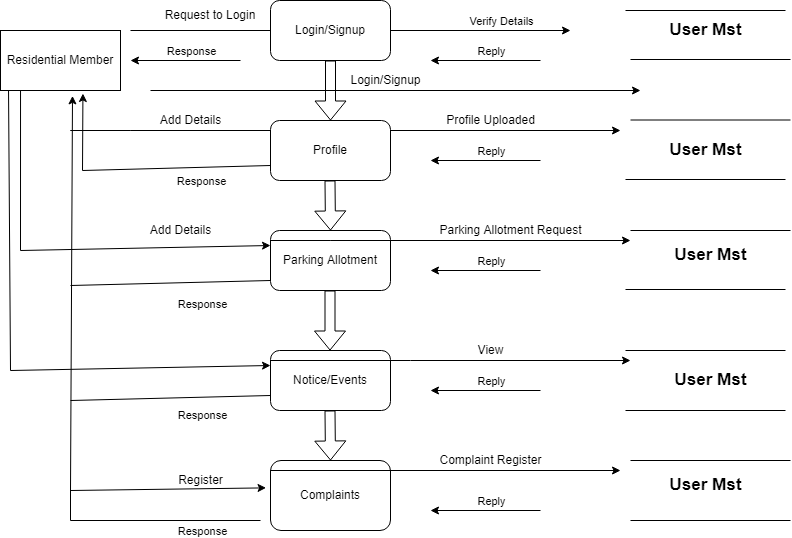
****

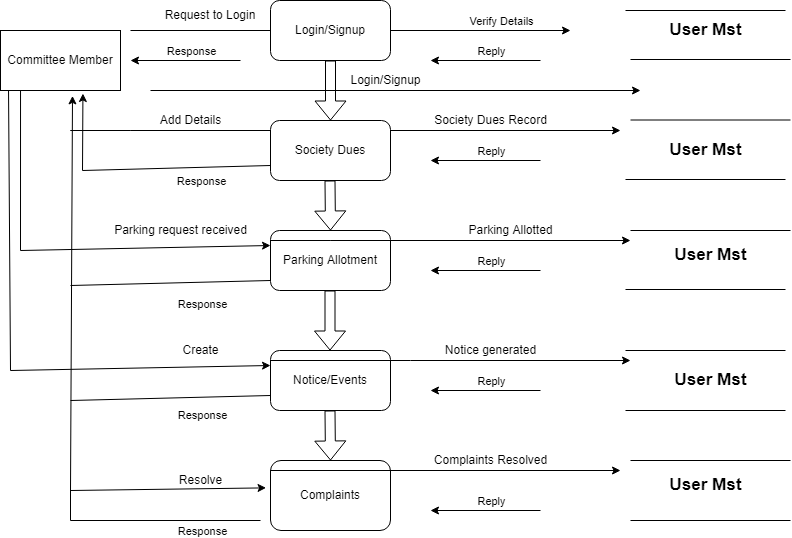
****

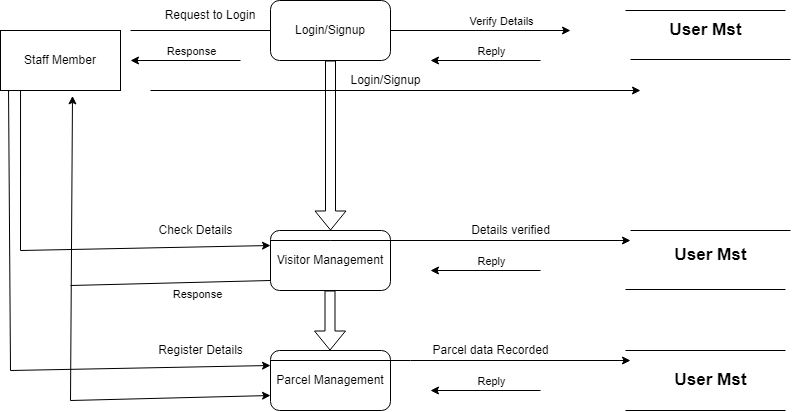
****

**C) DFD DIAGRAM**

A data-flow diagrams show the flow of data through a process or system (usually an information system). The DFD also contains information on each entity's and the process's outputs and inputs. There is no control flow in a data-flow diagram; there are no decision rules or loops.

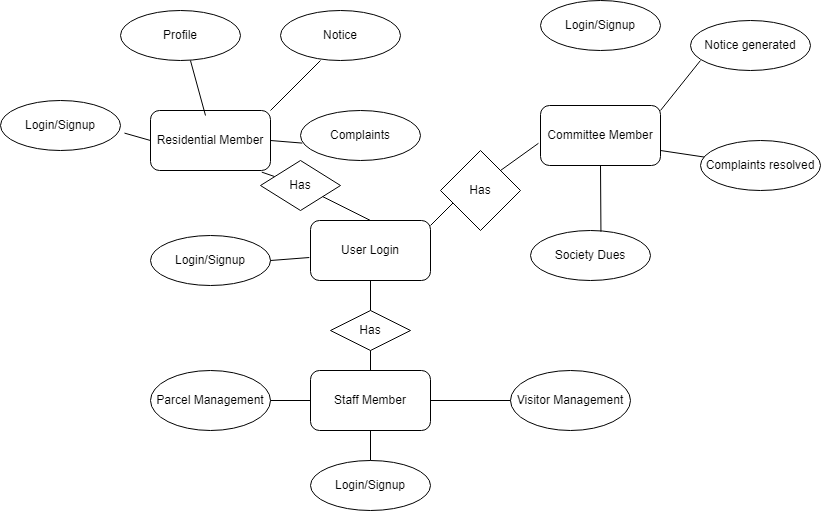
****

****

****

**D) ERD DIAGRAM**

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation that displays relationships between people, things, locations, concepts, or events in an information technology (IT) system. An ERD employs data modelling techniques to help design business processes and act as the foundation for a relational database.

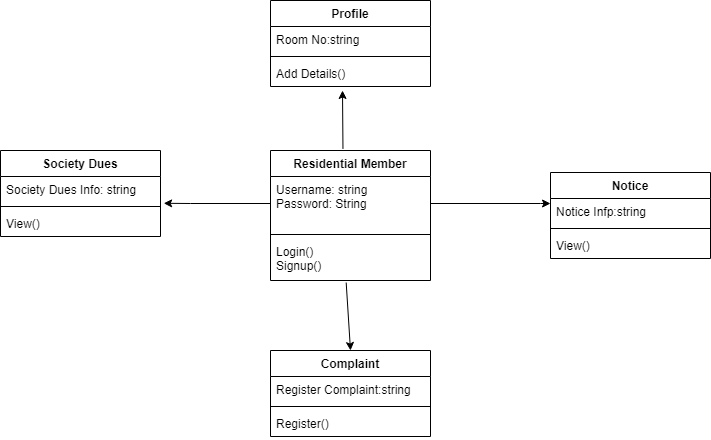
****

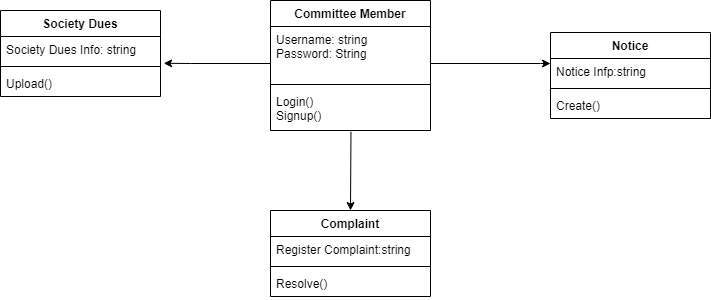
**E) CLASS DIAGRAM**

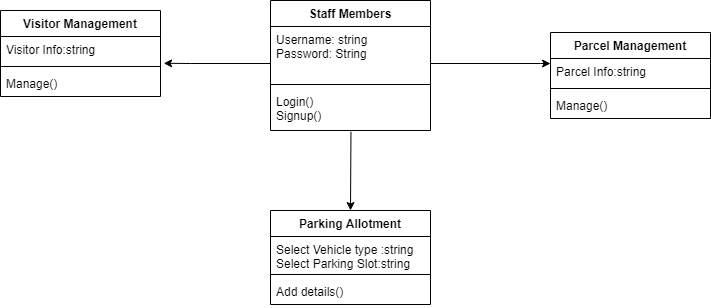
Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

The purpose of the class diagram can be summarized as −

* Analysis and design of the static view of an application.
* Describe responsibilities of a system.
* Base for component and deployment diagrams.
* Forward and reverse engineering.

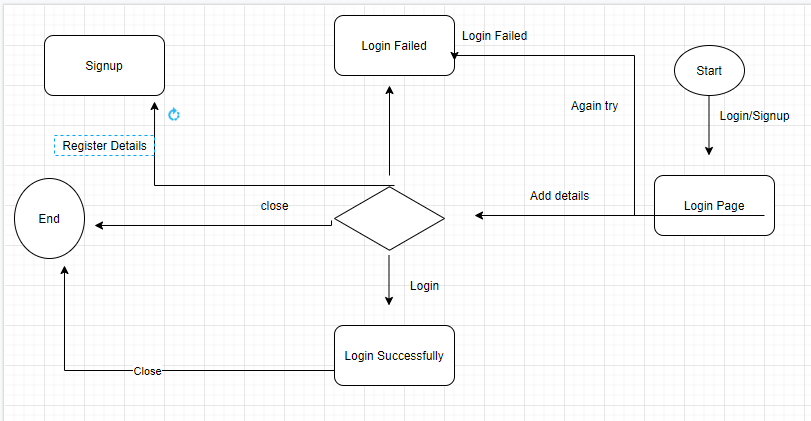
****

****

****

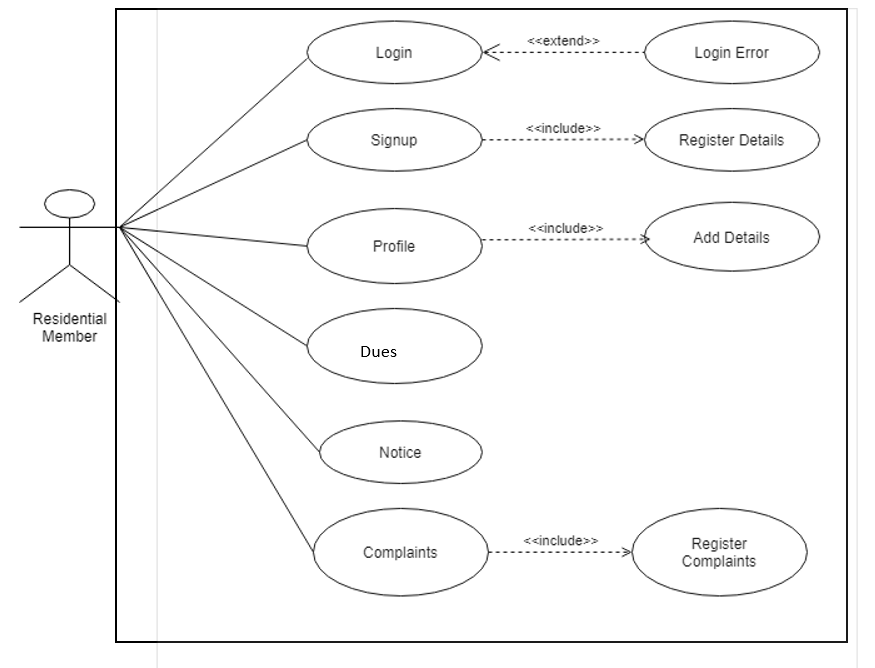
**F) STATE TRANSITION DIAGRAM**

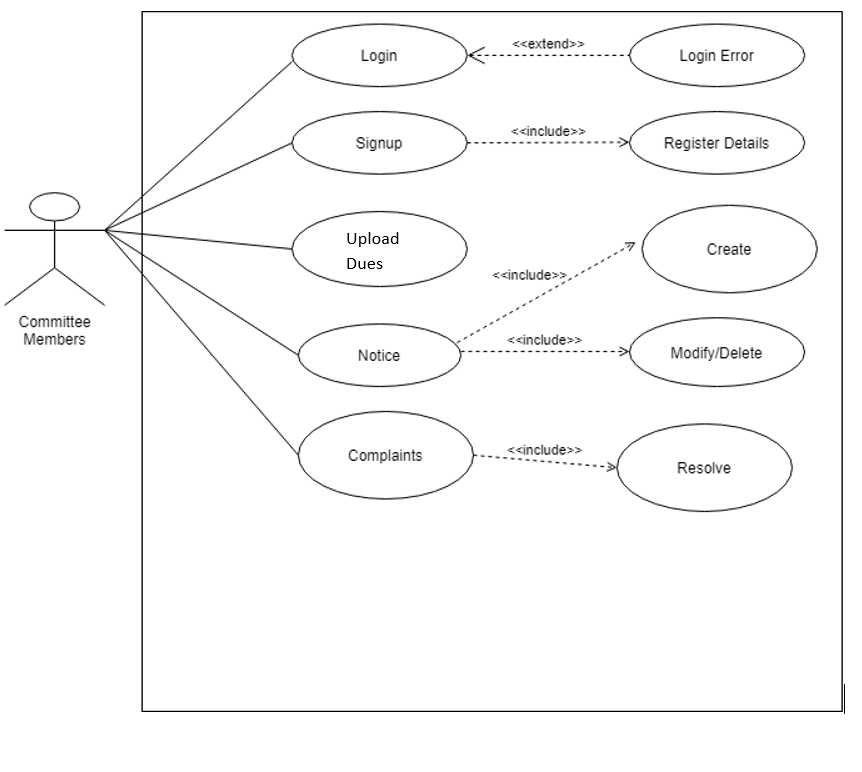
A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

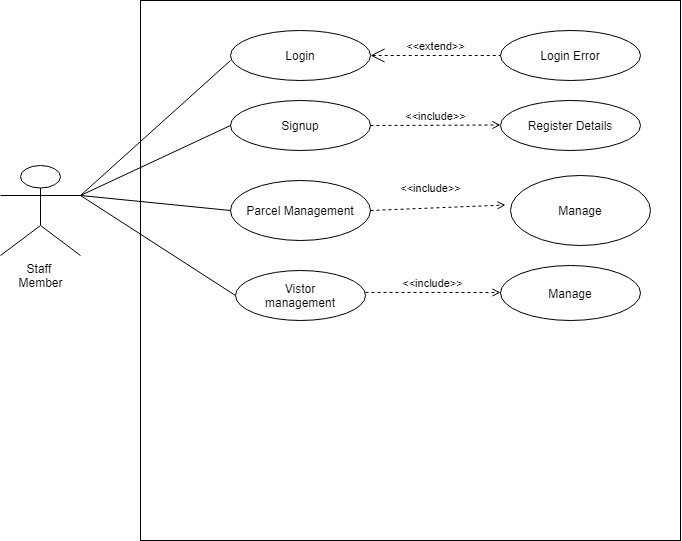
****

**G) USE CASE DIAGRAM**

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

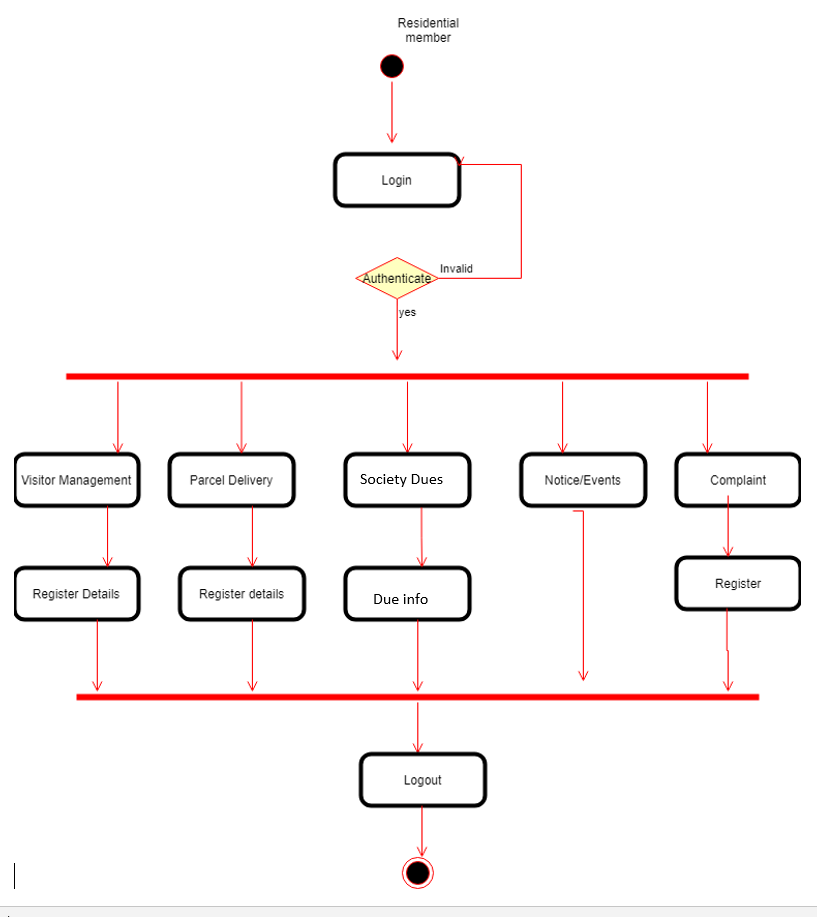
****

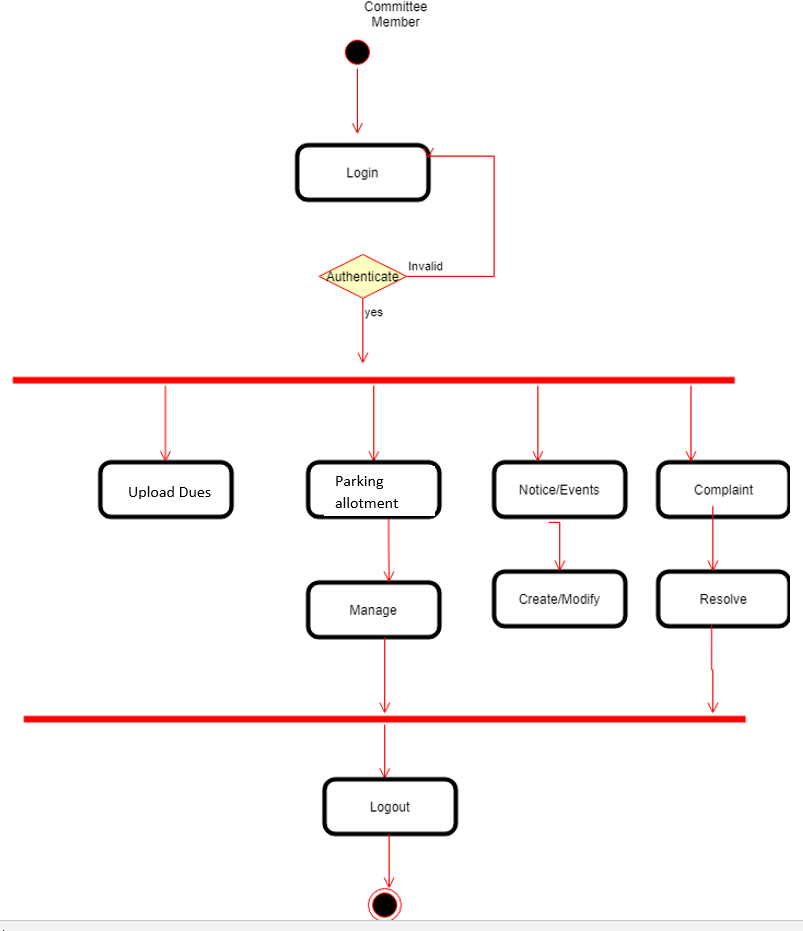
****

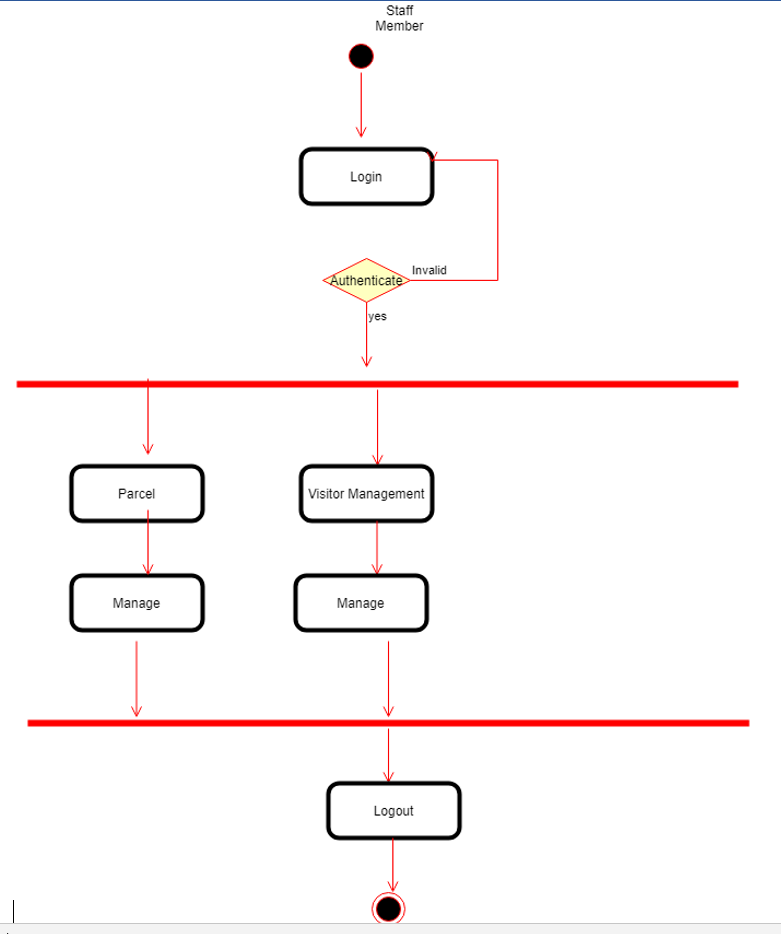
****

**H) ACTIVITY DIAGRAM**

An activity diagram is a behavioral diagram i.e., it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed

****

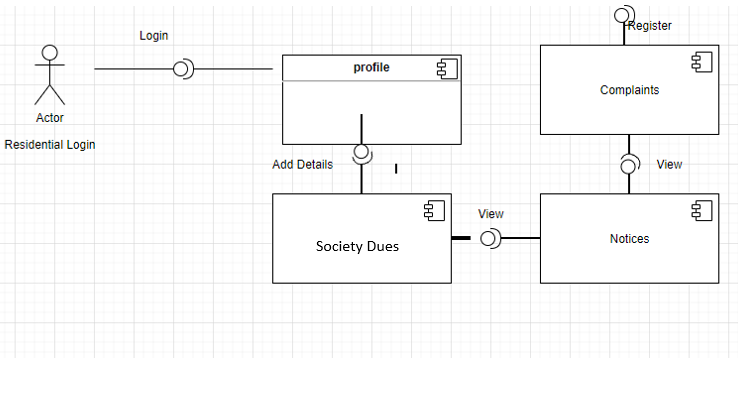
****

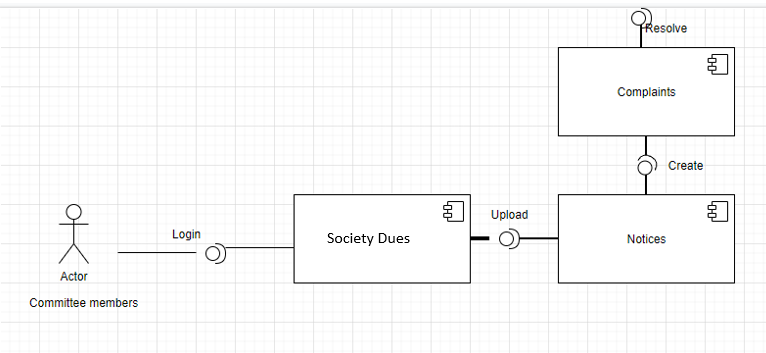
****

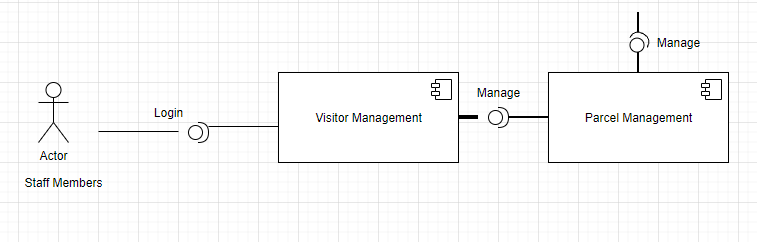
* 1. **COMPONENT DIAGRAM**

Component diagram is a special kind of diagram in UML.

Component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

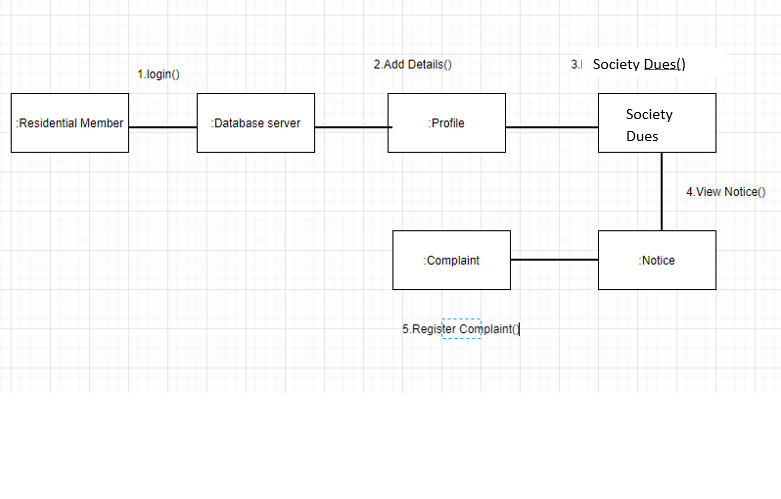
****

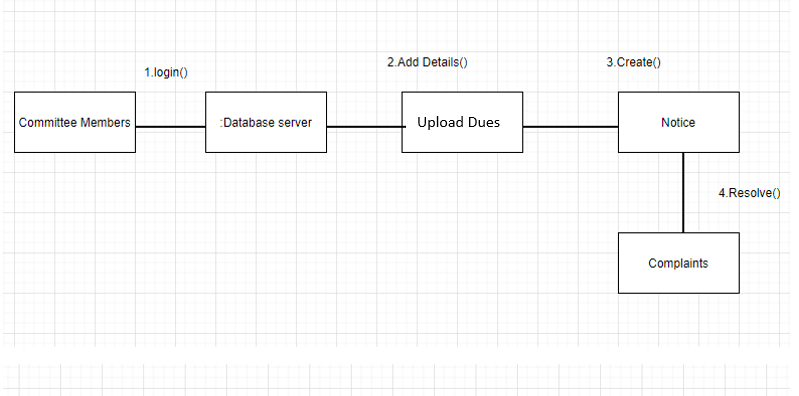
****

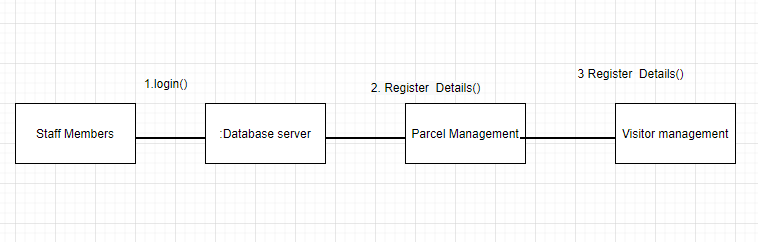
****

**J) COLLLABORATION DIAGRAM**

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). These diagrams can be used to portray the dynamic behavior of a particular use case and define the role of each object.

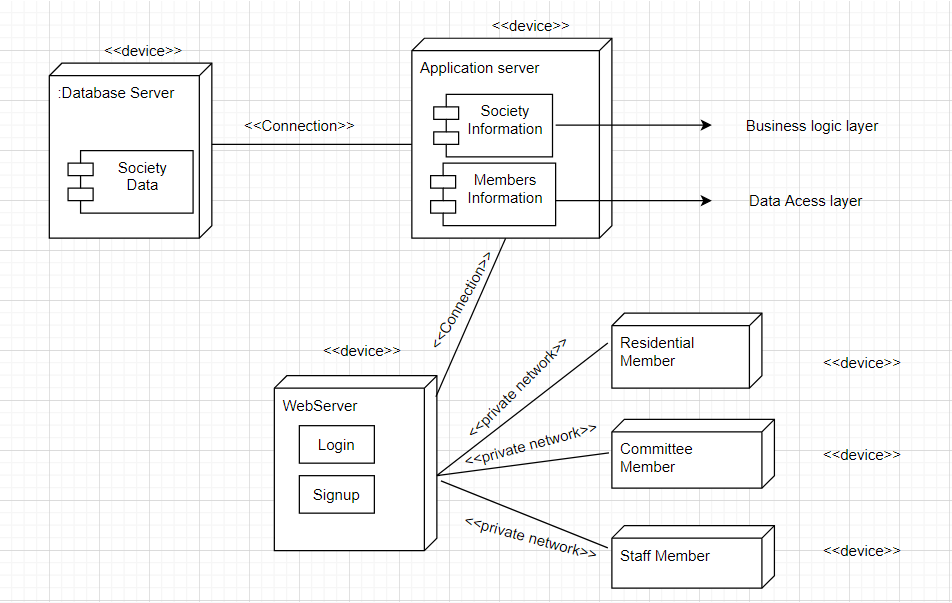
****

****

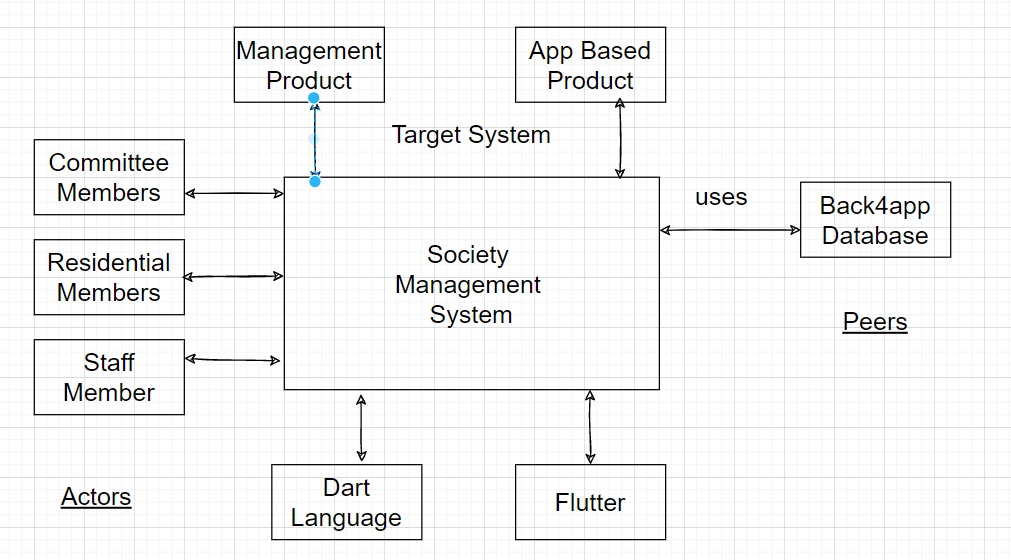
****

**K) DEPLOYMENT DIAGRAM**

A deployment diagram in the Unified Modeling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components exist, what software components run on each node, and how the different pieces are connected

****

**L) ARCHITECTURE DESIGN**

****

**1) Super ordinate System:** using the target system as part of some higher-level processing scheme.

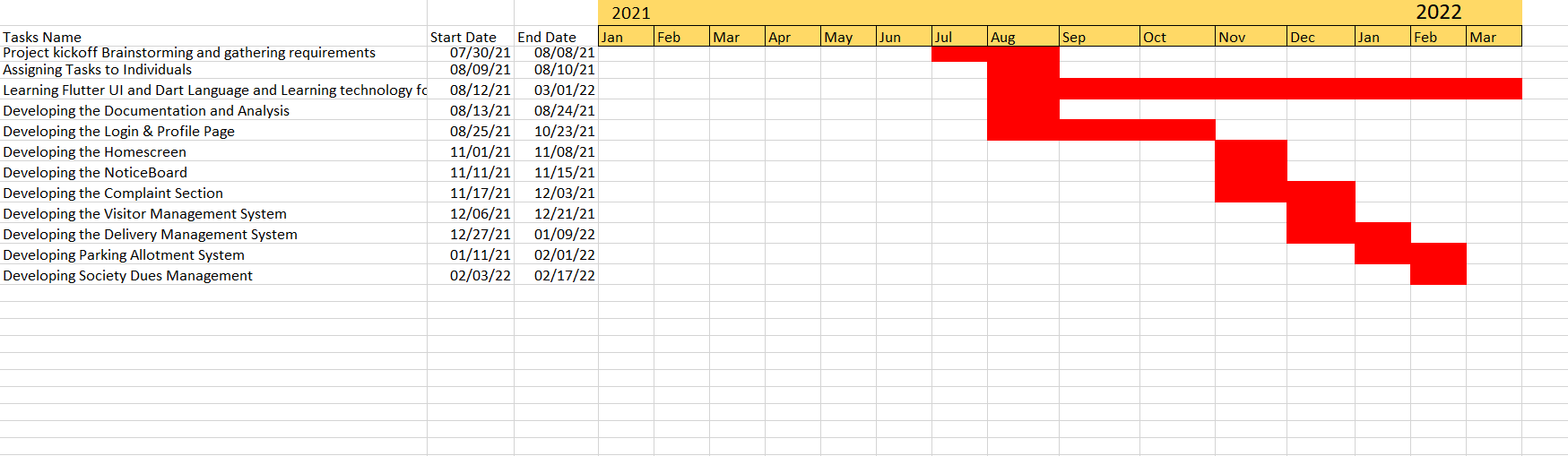
**2) Sub ordinate System: used** by the target system to provide data or processing needed to complete the target system.

**3) Peer level System:** Producing or consuming information needed by peers and the target system.

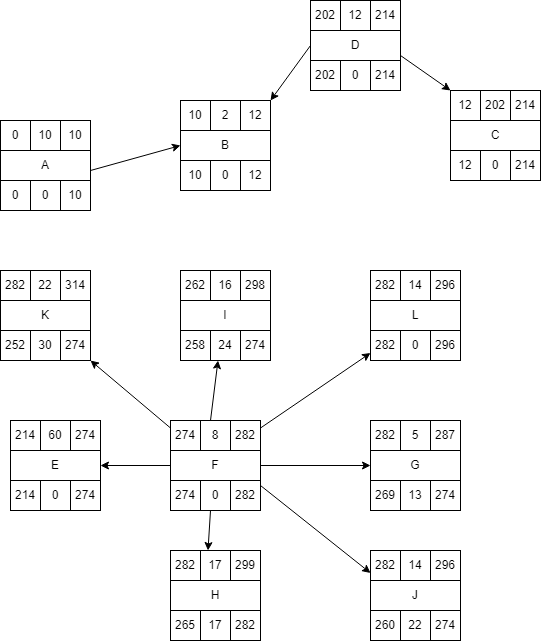
**4) Actors:** Those entities interact with the target system by producing or consuming information that is necessary to requisite.

**3) SYSTEM PLANNING**

**A) Gantt Chart**

****

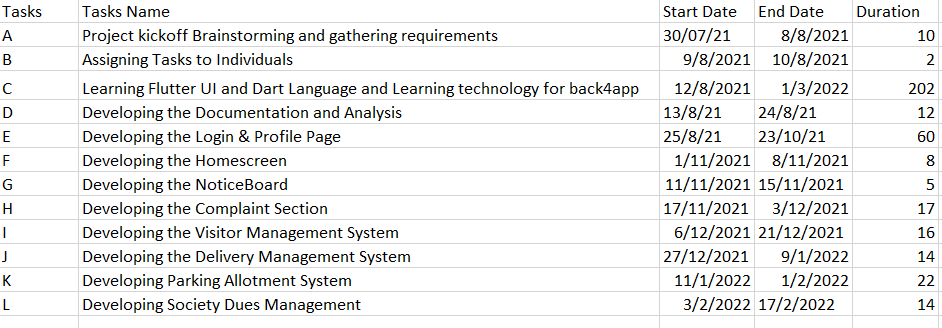
**B) PERT CHART**

****

|  |  |
| --- | --- |
| Tasks | Tasks Name |
| A | Project kickoff Brainstorming and gathering requirements |
| B | Assigning Tasks to Individuals |
| C | Learning Flutter UI and Dart Language and Learning technology for back4app |
| D | Developing the Documentation and Analysis |
| E | Developing the Login & Profile Page |
| F | Developing the Home screen |
| G | Developing the Noticeboard |
| H | Developing the Complaint Section |
| I | Developing the Visitor Management System |
| J | Developing the Delivery Management System |
| K | Developing Parking Allotment System |
| L | Developing Society Dues Management |
|  |  |

|  |  |  |
| --- | --- | --- |
| Tasks | Dependent | Duration(Days) |
| A | - | 10 |
| B | - | 2 |
| C | B | 202 |
| D | C,B | 12 |
| E | C | 60 |
| F | E | 8 |
| G | F | 5 |
| H | F | 17 |
| I | F | 16 |
| J | F | 14 |
| K | F | 22 |
| L | F | 14 |

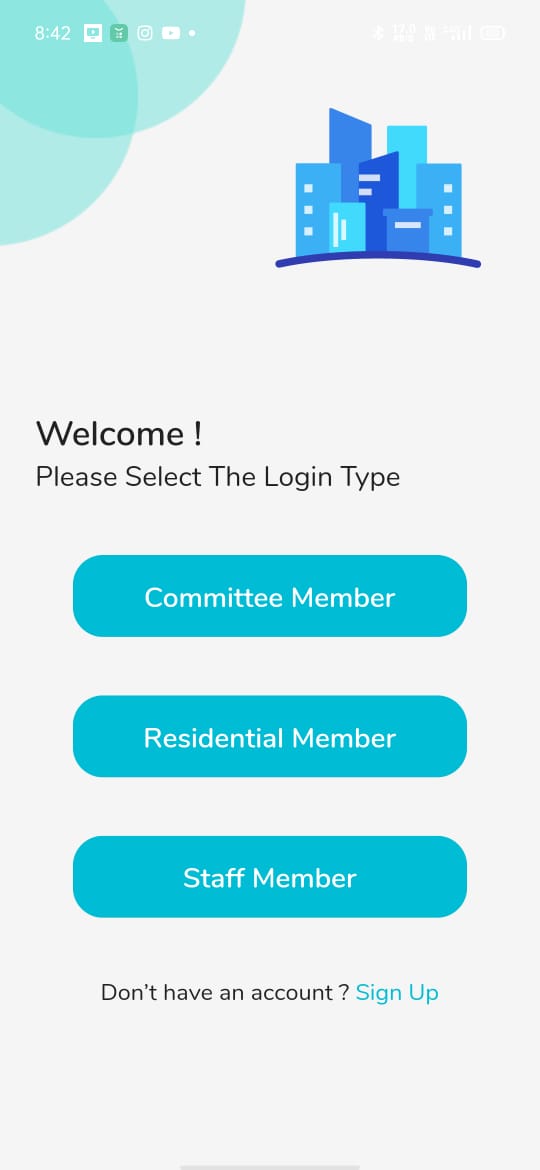
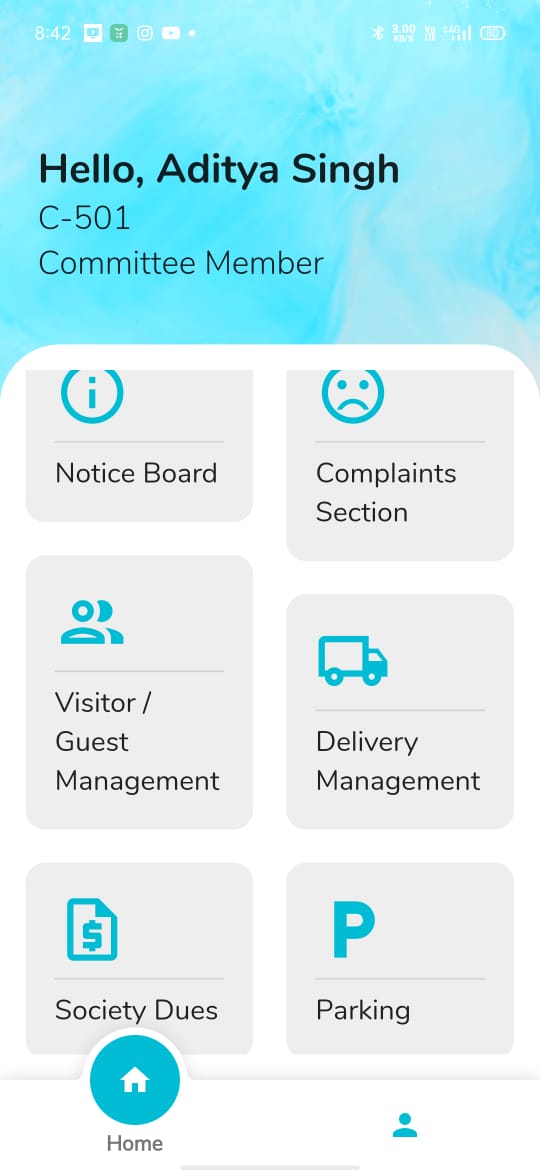
**C) Project Scheduling**

****

**4) System Implementation**

**Home Screen**

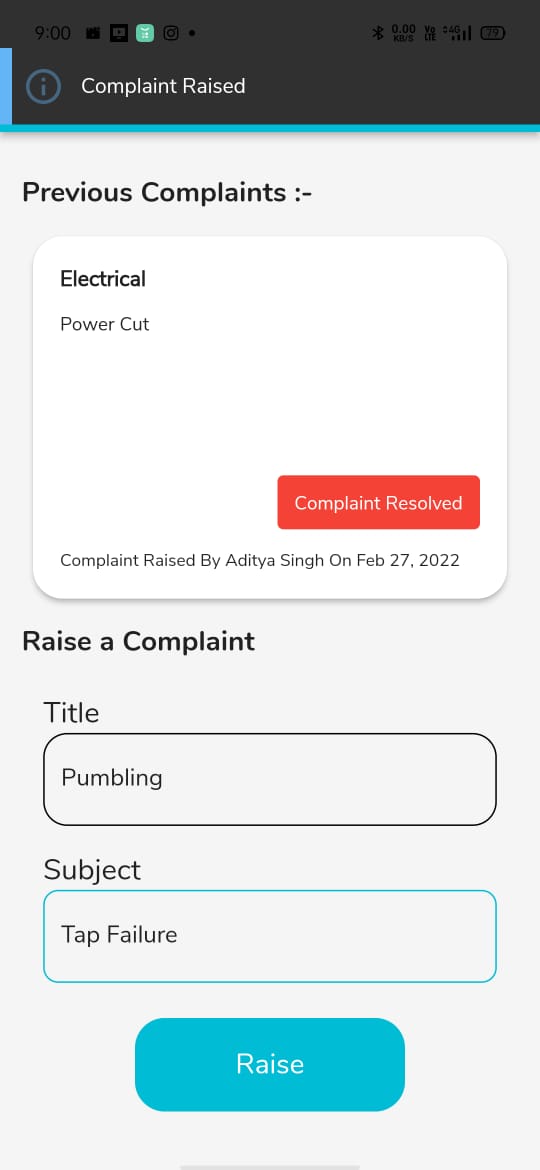
**Login Info**

****

Explanation of the Above Screenshot: -

There is a login page where user will select his/her role and after that home screen will appear for their respective roll

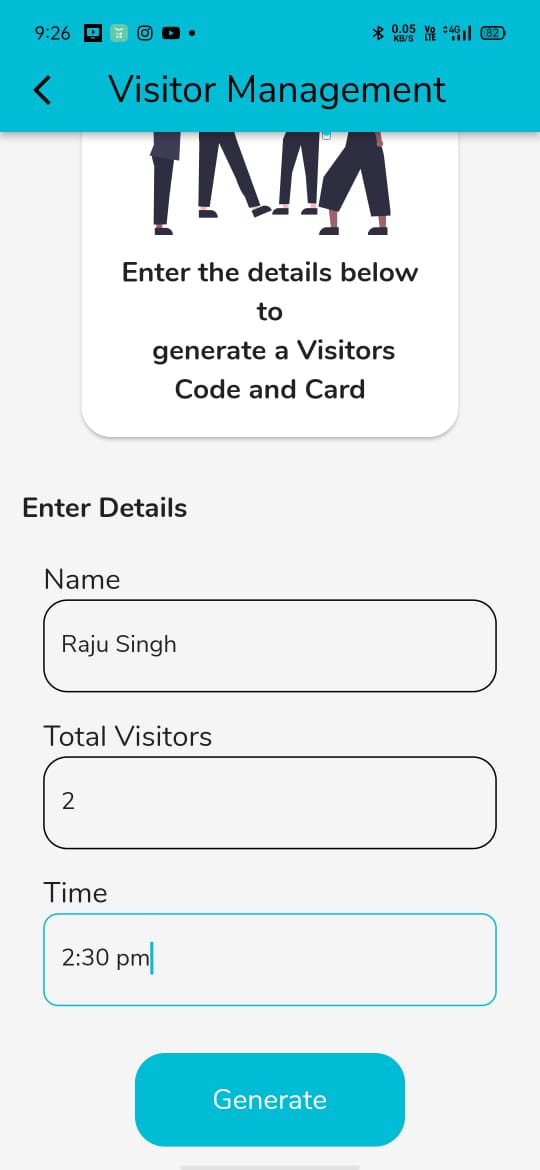
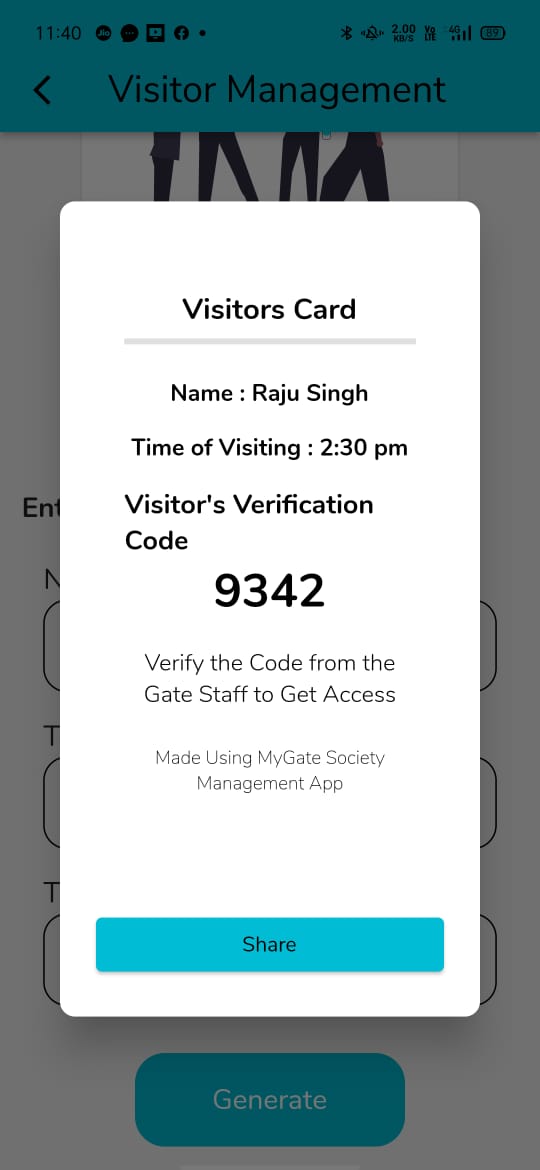
**Complaint Section**

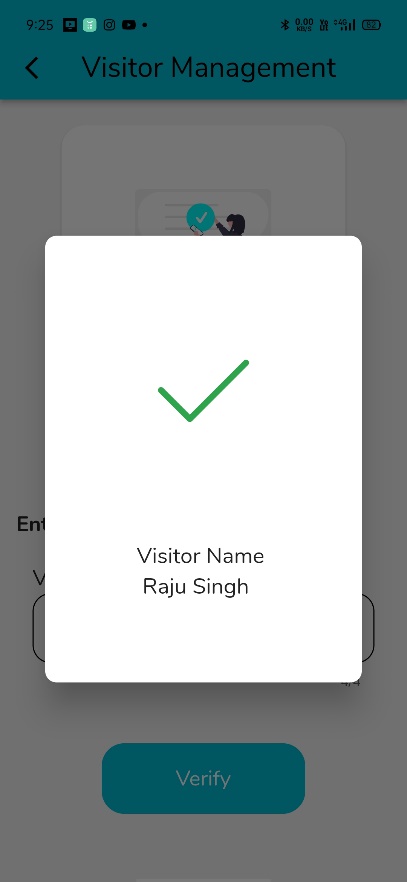
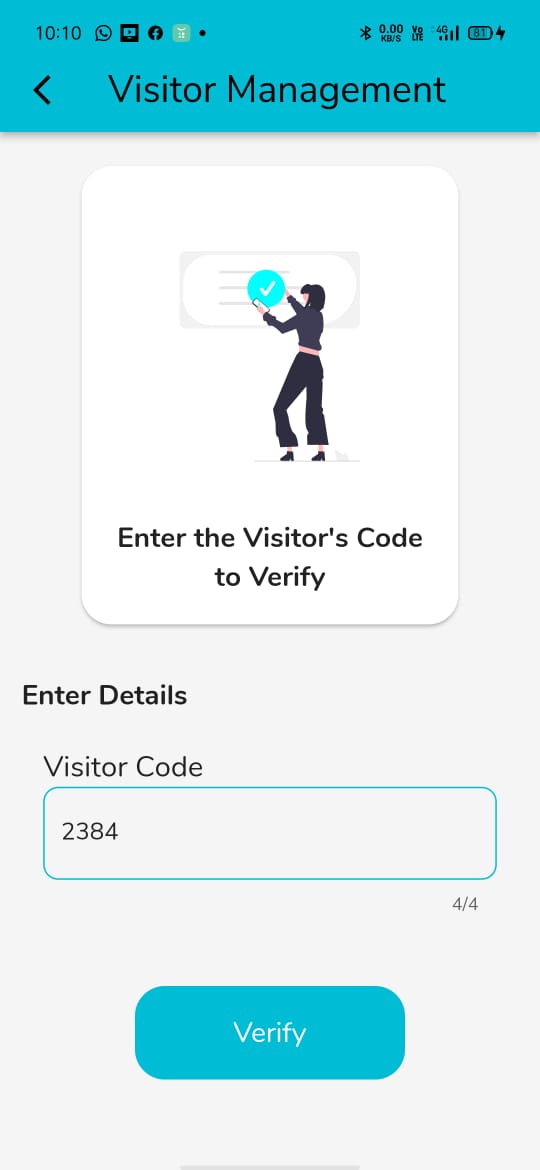
****

Explanation of the Above Screenshot: -

This is a complaint module where the user can add a complaint and a "complaint resolved" button where the user can click once their complaint has been resolved.

**Visitor management**

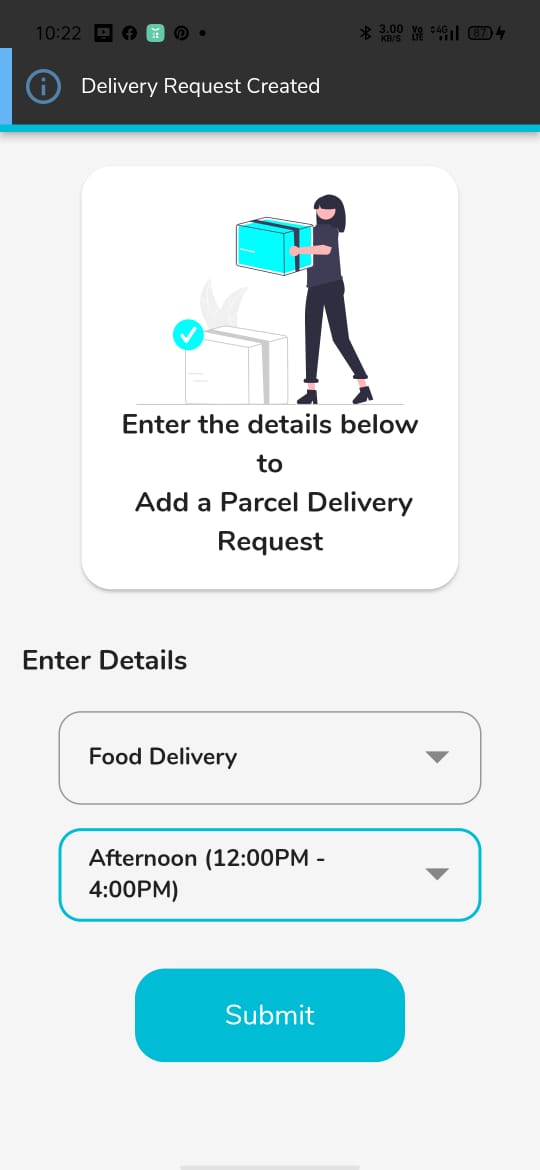
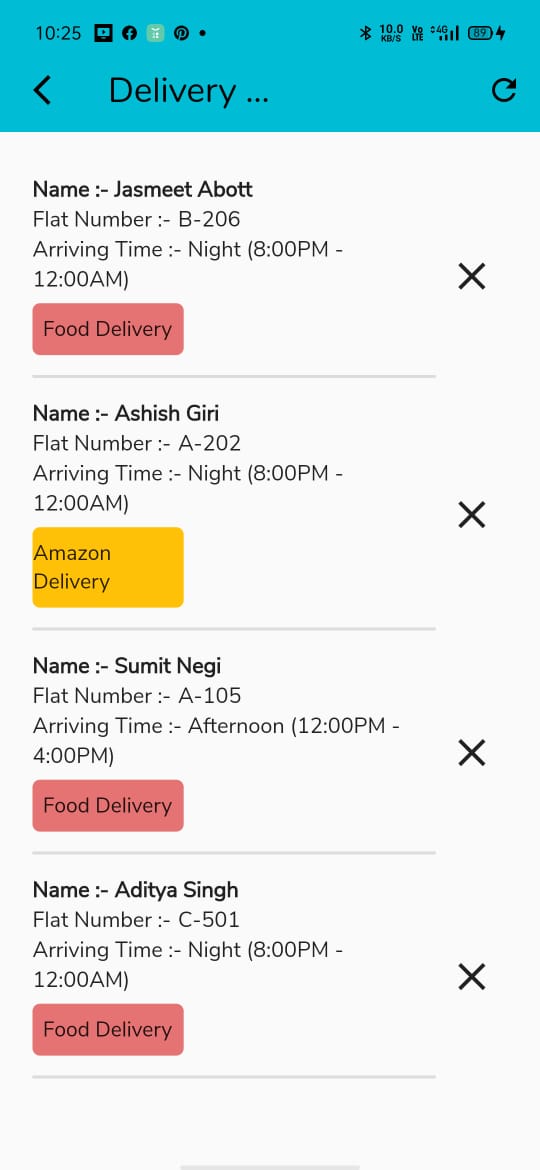
****

****

Explanation of the Above Screenshot: -

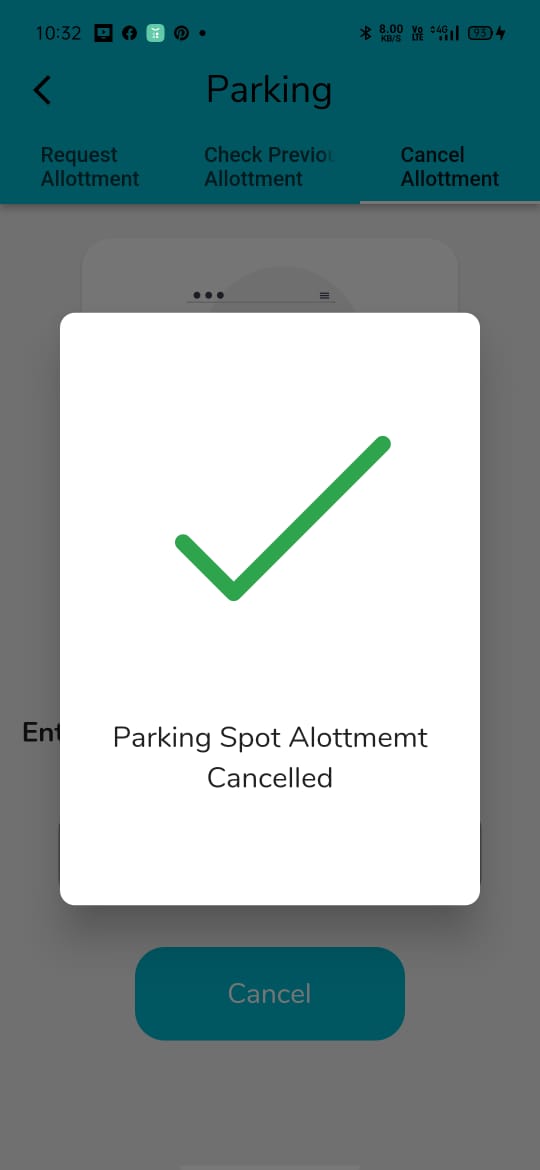
This is a visitor management system in which a user can create a visiting card for a non-society member and send it to them, after that a staff person can verify if the code is legitimate or not.

**Delivery Management**

****

Explanation of the Above Screenshot: -

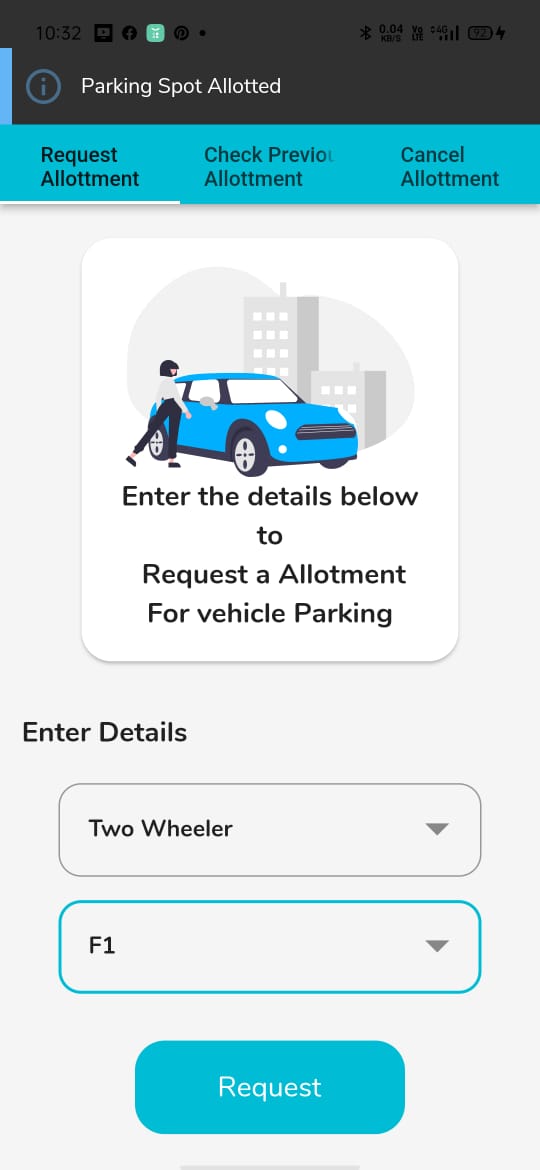
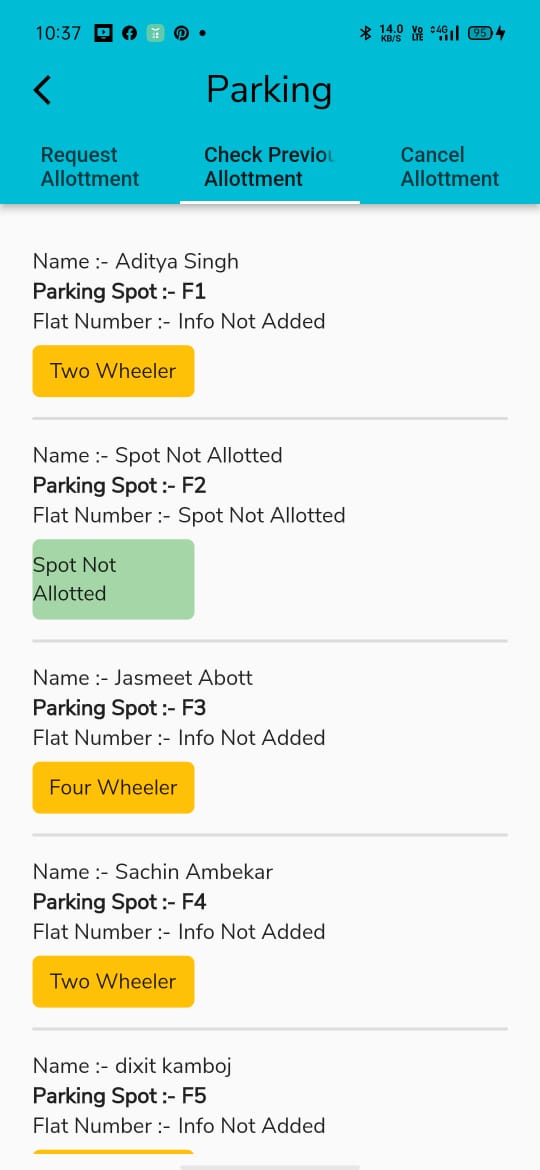
This is a Delivery Management system in which a user can submit a parcel delivery request, and once their items have been delivered, a staff member can delete the request.

****

**Parking Allotment**

Explanation of the Above Screenshot: -

This is a parking allotment where the user can inspect, request, and cancel a vehicle parking allotment.

****

Explanation of the Above Screenshot: -

This is a parking allotment where the user can inspect, request, and cancel a vehicle parking allotment.

**5)Cost and benefits analysis and Software parameter Estimation**

1. **Cost Estimation Model**

Software cost estimation is important for making good management decisions. It is

also connected to determining how much effort and time a software project requires.

The COnstructive COst MOdel (COCOMO) is an example of regression models used for estimating software cost and effort.

* **The Development Mode**

There are several modes of software development. These different software

development modes have cost-estimating relationships which are similar in form, but which yield significantly different cost estimates for software products of the same size. In the COCOMO Model, one of the most important factors contributing to a project's duration and cost is the Development mode. Every project is considered to be developed in one of three modes:

• Organic Mode.

• Semidetached Mode

• Embedded Mode

To estimate the effort and development time, COCOMO use the same equations but with

different coefficients (a, b, c, d in the effort and schedule equations) for each development

mode. Therefore, before using the COCOMO model, one must be able to recognize the

development mode of our project.

**1. Organic Mode**:

In the organic mode the project is developed in a familiar, stable environment and the

product is similar to previously developed products. The product is relatively small, and

requires little innovation. Most people connected with the project have extensive experience

in working with related systems within the organization and therefore can usefully contribute

to the project in its early stages, without generating a great deal of project communication

overhead.

**2. Semidetached Mode:**

In this mode project's characteristics are intermediate between Organic and

Embedded. "Intermediate" may mean either of two things:

1. An intermediate level of project characteristics.

2. A mixture of the organic and embedded mode characteristics.

The size of a Semidetached mode product generally extends up to 300 KDSI.

**3. Embedded Mode:**

In this development mode Project is characterized by tight, inflexible constraints and

interface requirements. The product must operate within a strongly coupled complex of

hardware, software, regulations, and operational procedures.

The Basic COCOMO equations take the form

Effort (SM) = a \* (KDSI)

[Staff-Months]

Schedule (TDEV) = c \* (SM)

[Months]

Average Staffing = SM / TDEV [FSP]

The coefficients a, b, c, d is given below:

Software project a b c d

Organic 2.4 1.05 2.5 0.38

Semi-detached 3.0 1.12 2.5 0.35

Embedded 3.6 1.20 2.5 0.32

Effort Applied = 2.4(3)1.05

= 2.4(3.169)

= 7.6

Development time (D) = 2.5(7.6)0.38

= 2.5(2.16)

= 5.4 months

Total time required to develop the project

= 5.4

= 5.4 x 30

= 162 days

Cost per month = 3000

No of Developers = 3

Total Cost = 5.4 x 30 x 30 x 100

= 16,200

**6) Future Work**

The Future Work which will enhance and improve our project will be as follows:

a) Adding more and more modules like Rent room System.

b) Making the interface attractive and simple so that everyone can use it without any hassles.

c) Adding new technologies without adding more complexity in the project.

**7) Conclusion**

Mobile apps provide a hassle free, user friendly affordable communication platform to develop healthy relations between housing society members and a smarter way of interconnectivity. It manages housing society which assists the society members to play their roles effectively. In this digital era, this app will be a well-tailored app to meet the needs of a Housing Society. It really helps all the people in a residential society on a daily basis. The app helps to effectively manage every aspects of the Society Operations.

**8) Bibliography/References**

**Websites**

* [**https://www.youtube.com/**](https://www.youtube.com/)
* [**https://stackoverflow.com/**](https://stackoverflow.com/)
* [**https://github.com/**](https://github.com/)
* [**https://flutter.dev/**](https://flutter.dev/)
* [**https://pub.dev/**](https://pub.dev/)