**SOCIETY MANAGEMENT SYSTEM**

###### A Project Report

Submitted in partial fulfillment of the Requirement for the award of the Degree of

#### BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

###### By

Amol Patil

Seat No.:

**Under the esteemed guidance of Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation: Course Co-Ordinator**

**Prof. \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Designation: Assistant Professor**

###### DEPARTMENT OF INFORMATION TECHNOLOGY

**JMF’s VANDE MATARAM DEGREE COLLEGE OF SCIENCE & COMMERCE**

***(Affiliated to University of Mumbai)***

###### Dombivali,421201 Maharashtra 2020-2021

**JMF’s VANDE MATARAM DEGREE COLLEGE OF SCIENCE & COMMERCE**

***(Affiliated to University of Mumbai)***

**DOMBIVALI-MAHARASHTRA-PINCODE**

#### DEPARTMENT OF INFORMATION TECHNOLOGY

**CERTIFICATE**

This is to certify that the project entitled, **“SOCIETY MANAGEMENT SYSTEM”,** is Bonafide work of **AMOL B. PATIL**

Bearing Seat no.: **( )** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

###### Internal Guide Co-Ordinator

**External Examiner**

**Date: College Seal**

**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

***(Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

PNR No.: ………………………………. Roll no.:

1. Name of the Student: Amol Patil
2. Title of the Project: Society Management Systems
3. Name of the Guide:
4. Teaching experience of the Guide:
5. Is this your first submission? Yes

Signature of the Student Signature of the Guide

Date ……………………… Date…………………

Signature of the Coordinator Date………………………

**DECLARATION**

I hereby declare that the project entitled the **“Society Management System”** done at “**Vande Mataram Degree College of Science &Commerce”** has not been in any case duplicated to submit to any other university for the award of any degree. According to me, yet nobody has been submitted this project in university.

The project is done in partial fulfilment of the requirements for the award of degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY** to be submitted as

Semester- 4 project as a part of curriculum.

###### Amol Patil

**ABSTRACT**

In this society management system, all the society categorize by the number of blocks and flats. The people who live in flats they may be an owner of flat or tenant of flat. In this era, people are very hectic schedule, so they do not have time for complain small problem related to flat. We have developed the system for society member they can make complain form anywhere any time and we resolve the Complain as soonas possible. In this system people can easily find address of the flat by providing member name.

## ACKNOWLEDGEMENT

A project is a creative work for each and everyone. A proper synchronization between the team members is must for completing the project successfully. I would like to extend my gratitude to, our **Principal Dr. Rajkumar Kolhe Sir** and all the staffs of our **Vande Mataram Degree College of Science and Commerce** for providing us moral support, conductive work environment which was needed to complete this project.

I would also like to thanks our Course Co-ordinator **Prof.\_\_\_\_\_\_\_\_** and all the faculties of IT Department for giving us the most needed guidance and continuous encouragement throughout the duration of the project and without them it would not have been possible to accomplish this project.

I am also extremely thankful to the **University of Mumbai** for having prescribed this project work to me as a part of the academic requirement in the final year of **Master of Science in Information Technology.**

Finally, I would like to thanks my group members and other friends who have been helped us to reach our project in a good state.

## TABLE OF CONTENTS

[CHAPTER 1: INTRODUCTION 10](#_TOC_250030)

* 1. [BACKGROUND 12](#_TOC_250029)
  2. [OBJECTIVES… 13](#_TOC_250028)
  3. PURPOSE, SCOPE & APLLICABILITY 14
     1. [PURPOSE 14](#_TOC_250027)
     2. [SCOPE 14](#_TOC_250026)
     3. [APPLICABILITY 14](#_TOC_250025)
  4. [ACHIEVEMENTS… 15](#_TOC_250024)

CHAPTER 2: SURVEY OF TECHNOLOGIES 16

* 1. [EXISTING SYSTEM 17](#_TOC_250023)
  2. [PROPOSED SYSTEM… 17](#_TOC_250022)

CHAPTER 3: REQUIREMENTS AND ANALYSIS… 18

* 1. [PROBLEM DEFINITION 19](#_TOC_250021)
  2. [REQUIREMENT SPECIFICATION 20](#_TOC_250020)
  3. [PLANNING AND SCHEDULING 21](#_TOC_250019)

[GANTT CHART](#_TOC_250018)

* 1. [SOFTWARE AND HARDWARE REQUIREMENTS… 22](#_TOC_250017)

[CHAPTER 4: SYSTEM DESIGN 23](#_TOC_250016)

* 1. SYSTEM MODULE 24
  2. UML Diagrams… 27
     1. Entity Relationship Diagram 27
     2. Class Diagram 30
     3. Use Case Diagram 32
     4. Sequence Diagram 34
     5. Activity Diagram 36
     6. DFD Diagram 38
  3. DATA SCHEMA… 40
  4. USER INTERFACE DESIGN 42

**CHAPTER 1**

#### INTRODUCTION

**BACKGR OUND 1.2OBJECTIVES**

**PURPOSE, SCOPE & APLLICABILITY**

* + 1. **1.3.3APPLICABILITY**
  1. **ACHIEVEMENTS**

## CHAPTER 1 INTRODUCTION

Society Management System is a platform that provide interface between member of society and society management. A society management system effectively manages and handles all the functioning of a society. The software system can store the data of various flat owners and their family members. The system also maintains and calculates the society maintenance as well as electricity bill automatically in individual flat bill. The system needs an administrator to input various flat owner data and billing amounts into it. The rest of the work is done by the system on its own. It calculates various associated costs, adds them up and provides a bill accordingly.

The two main users involved in this system are:

1.User (i.e., Member of society)

2. Admin

## Background:

As the title of our project itself “Society Management System” says that our system has to manages the administration of a society. So, to keep the information correct, it will manage the status/information about the procedures and proceedings.

In the existing system they required to maintain all records on paper, thus when we have to search any information related to events or the registered tenants it takes a lot of time and even human errors possibilities is more. Security was not maintained in the existing one.

## Objectives:

Our project is basically aimed in providing latest technologies and tendencies to the cooperative housing societies, so that all societies can update their administration and operation to each tenant through this system. To provide all the services required for organizing administration of the society.

Our main aim of this project is to make and provide some features to register users, sign in, access dashboard to view bills, lodge complaints, view complain status, view visitor, search visitor, generate reports, update own profile etc.

## Purpose, Scope and Applicability:

### Purpose:

The purpose of our project is to shorten the procedure of managing society by providing a web interface of secretary and the tenants. Another main purpose is that it's going save time of secretary and tenants both in terms like the tenants are not needed to visit to society office and in this way, they will not miss their society meetings and will also stay updated at the same time without any hassle or hurry.

It can lead to error-free, secure, reliable, fast management system. The society can maintain computerized records without redundant entries. It also maintains financial records for the maintenance bills and other similar expenses.

### Scope:

This web application includes the IT fields. Society tenants are the participants. They have to register for the respective society. Tenants could easily register the events of their choice. Easy to access the system anywhere and anytime. User will have the easy communication with the manager and gets instant message about any change updates easily.

This application leads to provide the user especially the tenants the reliability in finding the schedule of meetings at one place rather than to go for each of the meeting's venues.

### 1.3.3. Applicability:

Our project is going to be useful to tenants and managers both. As managers will not need to do any paper work like making the list of the tenants who have paid the maintenance or not.

This system saves the time and energy of tenants and managers so that the they won't have to go to each tenant's house to remind them or cal them for some urgent update making it a solution to the traditional but faulty system.

### Achievements

While I was performing this project, I had got a lot of knowledge about the MySQL database and PHP backend development which we have been used for the project and also about its versions and features.

Hence, this was the knowledge we had gained while performing this project.

###### Goals Achieved:

- The goals have been partially achieved.

# CHAPTER 2

#### SURVEY OF TECHNOLOGIES

* 1. **EXISTING SYSTEM**
  2. **PROPOSED SYSTEM**

**CHAPTER 2**

**SURVEY OF TECHNOLOGIES**

* 1. **Existing system**

Existing technology used is manual process. Manual process requires more man power. Manual process requires many records to maintain. Manager and authorities needs to take care to store each and every record of the tenants. Here, Data Security is not provided and even integrating a data is also a problem in the existing system.

* 1. **Proposed system**

Our Proposed system is a web application. In this application tenant details are maintained efficiently admin has a facility to view the tenant details like name, from which tenant, their contact number, and the details about the operations in which the tenants are involved. This has an enhanced facility. It is a fast, reliable system.

# CHAPTER 3

#### REQUIREMENTS AND ANALYSIS

* 1. **PROBLEM DEFINITION**
  2. **REQUIREMENT SPECIFICATION**
  3. **PLANNING AND SCHEDULING**
  4. **GANTT CHART**
  5. **SOFTWARE AND HARDWARE REQUIREMENTS**

**CHAPTER 3 REQUIREMENT & ANALYSIS**

**Problem Definition**

Before coming new technology, we used to face many problems like wastage of time and energy and many other things. So now here finally we have a new technology with us, the work has been becoming easier. Previously, Society Management System were very difficult and complicated to handle and also to maintain the data was very difficult and it was used to consume a lot time and man power.

So, the invented technologies provide us ease for completing any project. As if we know that the existing system was very complicated and time consuming because it is done by manual process, we are trying to build this project in such a way that it should reduce the man power and time.

The problems are:

* Data security is not provided in existing system whereas here this project is providing a data security as if only authenticated users can only access the system not others.
* Difficult to insert, modify and delete the records because there can happen some human errors
* The major problem was faced by tenants and managers because the old system consumed a lot time.
* Paper work is more and it becomes tedious to handle it.
  1. **Requirements Specification**

A software requirements specification is a description of a software system to be developed and also it describes the nature of project, software, or application. It rests out functional and non-functional requirements, and may include a set of use cases that describe user interaction that the software must provide. In addition to this, it also covers the information about safety and security requirements, software quality attributes of the project.

This system is capable of storing the data of events and all the tenants (users). Admin can add the operation. Admin will give the access to the tenants to register for the complaint or disposal of a dispute. This system is portable i.e., it can run on any operating system whether it is Windows 7/8/10 and even it is reliable and performance of the speed is high. It provides the Security so that our data will not leak.

### Planning and scheduling

This phase describes the tasks and steps necessary to plan and schedule the activities that we have should perform to achieve a successful completion of a project. A Project plan is prepared in such a manner which defines it as summary document that describes everything about project in terms of its objectives and how the objectives are to be achieved.

We have chosen this topic to overcome the problems like – previous system does not run efficiently, it was having a need of man power and difficult to response every user within short period of time, thus we got to know that there is aa important need to make it automated.

**Step 1:** The task of SMS is to reduce the man power and to do work paperless. In this we create data about events and users in this system. We have also created a login id and password for the users.

**Step 2: The** major task of our project is to complete the work easily and paperless.

**Step 3: Topic** selection: 4 days have been taken to select the topic

**Discuss:** 4 days have been taken to discuss about the topic

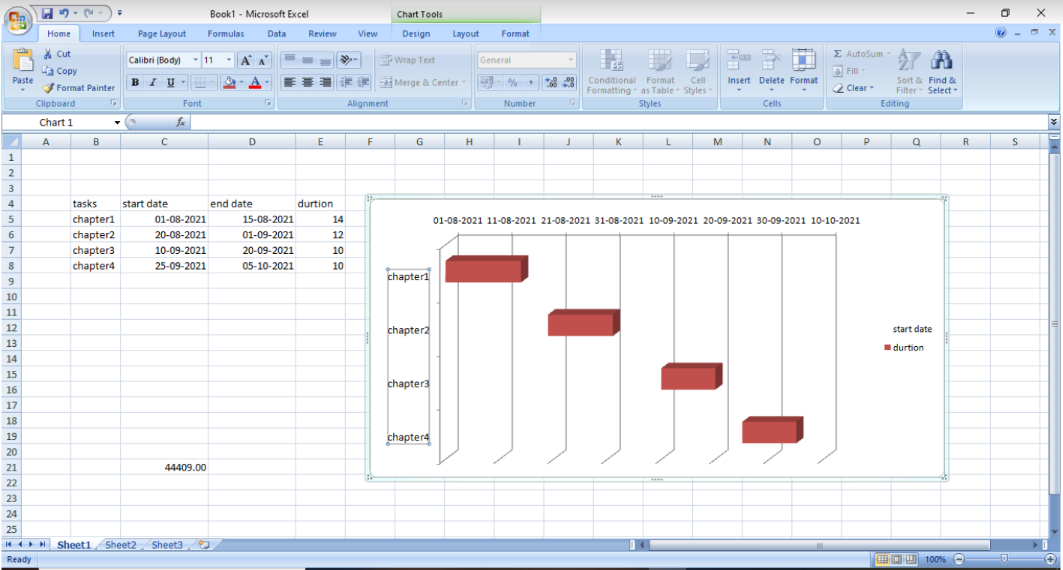
**Chapter 1:** 6 days have been taken for chapter 1 in which we have shown in introduction, Objectives, Purpose, Scope, Applicability, Achievements.

**Chapter 2:** 2 days have been taken for survey of technologies.

**Chapter 3:** Appearing

###### Gantt Chart

A Gantt Chart is a horizontal bar chart developed as a production tool in 1917 by Henry L. Gantt, an American engineer and social scientist. Frequently, used in project management, a Gantt chart provides a graphical illustration of a schedule that helps to complete a specific task in a project.



### Software and Hardware Requirements

###### Software Requirement:

* + - Database: MySQL
    - Server: Apache
    - Frontend: HTML
    - Scripting Language: JavaScript
    - IDE: VS Code (insiders)
    - Backend Technology: PHP

###### Hardware Requirements:

* + - * Any processor after Pentium 4.
      * Any version of Windows XP or later.
      * Processor speed: 2.0 GHz
      * RAM: 2-4GB
      * Hard disk: 40GB to 80 GB

###### Features:

* Protect the database of the firm by requiring a correct and registered username and password.
* Provide an option for users to update information.
* Create new events/updates.
* Provide an easy function where you can navigate the forms whenever necessary.
* User friendly.

# CHAPTER 4

#### SYSTEM MODULE

* 1. **UML Diagrams**
     1. **Entity Relationship Diagram**
     2. **Class Diagram**
     3. **Use Case Diagram**
     4. **Sequence Diagram**
     5. **Activity Diagram**
     6. **DFD Diagram**
  2. **DATA SCHEMA**
  3. **USER INTERFACE DESIGN**

## CHAPTER 4 SYSTEM DESIGN

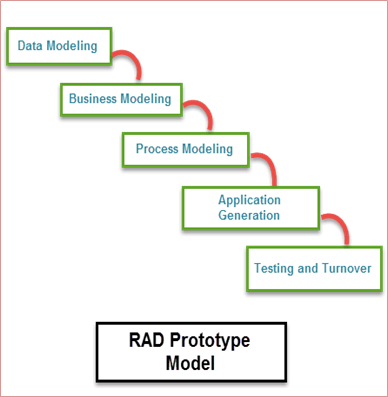
* 1. **System Model**

**RAPID APPLICATION DEVELOPMENT MODEL**

RAD or Rapid Application Development process is an adoption of the waterfall model; it targets at developing software in a short span of time. RAD follow the iterative

SDLC RAD model has following phases

* Business Modeling
* Data Modeling
* Process Modeling
* Application Generation
* Testing and Turnover



It focuses on input-output source and destination of the information. It emphasizes on delivering projects in small pieces; the larger projects are divided into a series of smaller projects. The main features of RAD model are that it focuses on the reuse of templates, tools, processes, and code.



###### RAD Model in Software Engineering

**Different phases of RAD model includes:**

**Business Modeling:** On basis of the flow of information and distribution between various business channels, the product is designed.

**Data Modeling:** The information collected from business modeling is refined into a set of data objects that are significant for the business.

**Process Modeling:** The data object that is declared in the data modeling phase is transformed to achieve the information flow necessary to implement a business function.

**Application Generation:** Automated tools are used for the construction of the software, to convert process and data models into prototypes.

**Testing and Turnover:** As prototypes are individually verified during every iteration, the overall testing time is reduced in RAD.

**When to use RAD Methodology:**

* When a system needs to be produced in a short span of time (2-3 months)
* When the requirements are known
* When the user will be involved all through the life cycle
* When technical risk is less
* When there is a necessity to create a system that can be modularized in 2-3 months of time
* When a budget is high enough to afford designers for modeling along with the cost of automated tools for code generation

#### Advantages-

Flexible and adaptable to changes

It is useful when you have to reduce the overall project risk It is adaptable and flexible to changes

It is easier to transfer deliverables as scripts, high-level abstractions and intermediate codes are used

Due to code generators and code reuse, there is a reduction of manual coding. Due to prototyping in nature, there is a possibility of lesser defects.

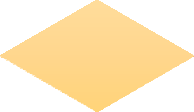
Each phase in RAD delivers highest priority functionality to client With less people, productivity can be increased in short time **Disadvantages-**

It can't be used for smaller projects

When technical risk is high, it is not suitable Requires highly skilled designers or developers

Reduced scalability occurs because a RAD developed application begins as a prototype and evolves into a finished application.

* 1. **UML DIAGRAMS ER-DIAGRAM**



An Entity Relationship diagram shows the relationships of entity sets stored in a database. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.

###### Symbols:

* + 1. **Entity -**Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.



* + 1. **Attributes** -Attributes are the properties of entities. Attributes are represented by means of ellipses. Attributes are of 2 types:
       1. **Composite Attributes:** They are further divided in tree like structure. Every node is then connected to its attribute.



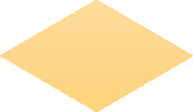
* + - 1. **Derived Attributes:** It is depicted by double ellipse.



* + 1. **Relationships-** Relationships are represented by diamond shapes. All the entities Participating in a relationship, are connected to it by a line.

###### --Binary Relationship and Cardinality:

1. **One-to-One: When** only one instance of a entity is associated with a relationship.

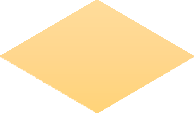


~~1~~   ~~1~~

1. **One -to-many:** When more than one instance of an entity is associated with a relationship.

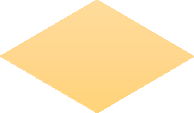
1 N

1. **Many -to-one:** When more than one instance of an entity is associated with a relationship.



N 1

1. **Many- to -Many: When** more than one instance of an entity on the left and more than one instance of an entity on the right can be associated.



N

N

###### --Participation Constraints:

1. **Total Participation:** Each entity is involved in a relationship. It is represented by double lines.
2. **Partial Participation:** Not all entities are involved in a relationship. It is

represented by single line.

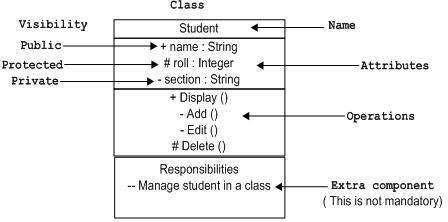


### CLASS DIAGRAM

###### Class Notation:

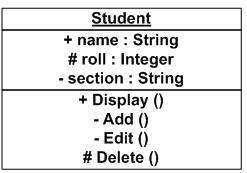
A class represented by the following figure. This diagram is divided into 4 parts:

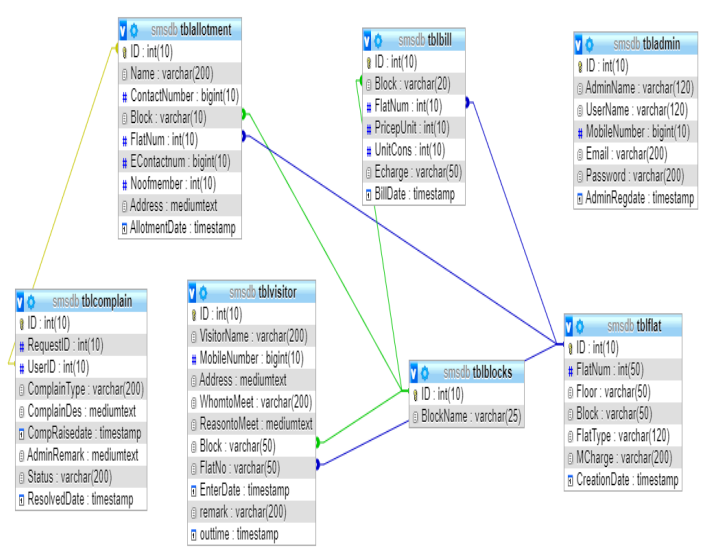
* The top section is used to name the class.
* The second one is used to show the attributes of the class.
* The third section is used to describe the operations performed by the class.
* The fourth section is optional to show any additional components.



###### Object Notation:

The *object* is represented in the same way as class. The only difference is the *name* which is underlined as shown in the figure. As the object is an actual implementation of a class, which is known as the instance of the class. Hence, it has the same usage as class.





### USE CASE DIAGRAM

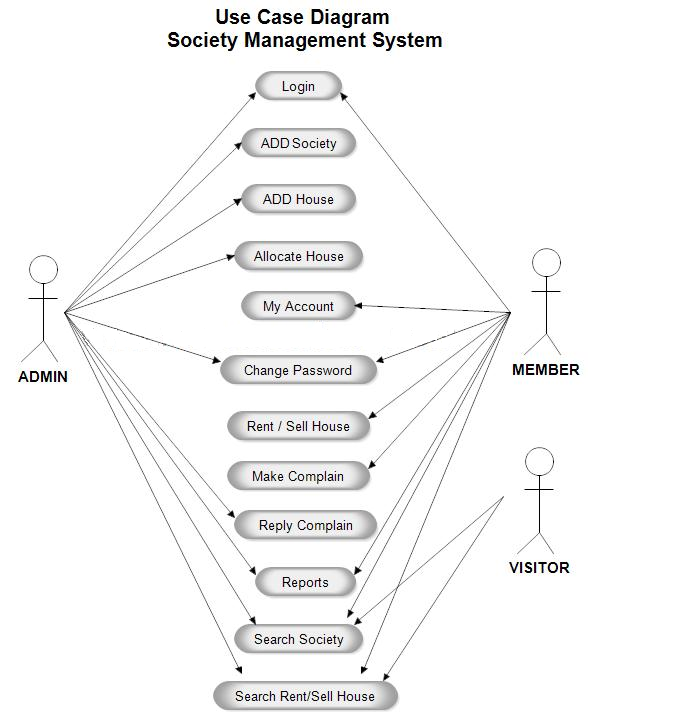
A use case diagram as its simplest is a representation of a user’s interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It is a requirement analysis concept. Describes the actions from a point of view of user. A sequence of events is involving. Interactions of a user with the system specifies one aspect of the behavior of a system, without specifying the structure of the system.

A use case diagram contains 4 components:

* **Actor:** Actor represents roles that take on when they use the IT system. Actors can be person, another system or organization.
* **Use case:** Use case describes the interactions that take place between actors and Systems during the execution.
* **Association:** An association is a connection between an actor and a use case. An association indicates that an actor can carry out a use case. Several actors at one use case mean that each actor can carry out the use case.
* **Include Relationships:** An include relationship is a relationship between two cases:

<<include>>

It indicates that the use case to which the arrow points is included in the use case on the other side of the arrow. This makes it possible to reuse a use case in another use case.



### SEQUENCE DIAGRAM

Sequence diagram is an “interaction diagram” that models a single scenario executing in a system.

###### Key Parts:

* **Participant –** An object or an entity, the sequence diagram actor.
* **Sequence –** Diagram starts with an unattached “found message” arrow.
* **Message –** Communication between objects.

###### Syntax and Semantics:

**An object:** a box with an underlined label that specifies the object type and optionally the object name.

* + Write the object’s name if it clarifies the diagram.
* An object’s “life line” is represented by a dashed vertical line.
* Represents the life span of the object during the scenario being modelled.

###### Representing messages between objects:

A message: an horizontal arrow to the receiving object. Write message name and arguments above the arrow.

###### Different types of messages:

Type of arrow indicates types of messages:

* Synchronous message: Solid arrow with solid head
* Asynchronous message: Solid arrow with stick head.
* Return message: Dashed arrow with stick head.

Diagram

Description automatically generated

### ACTIVITY DIAGRAM

The activity diagram is used to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as the operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagram deals with all type of flow controls by using different elements such as fork, join etc.

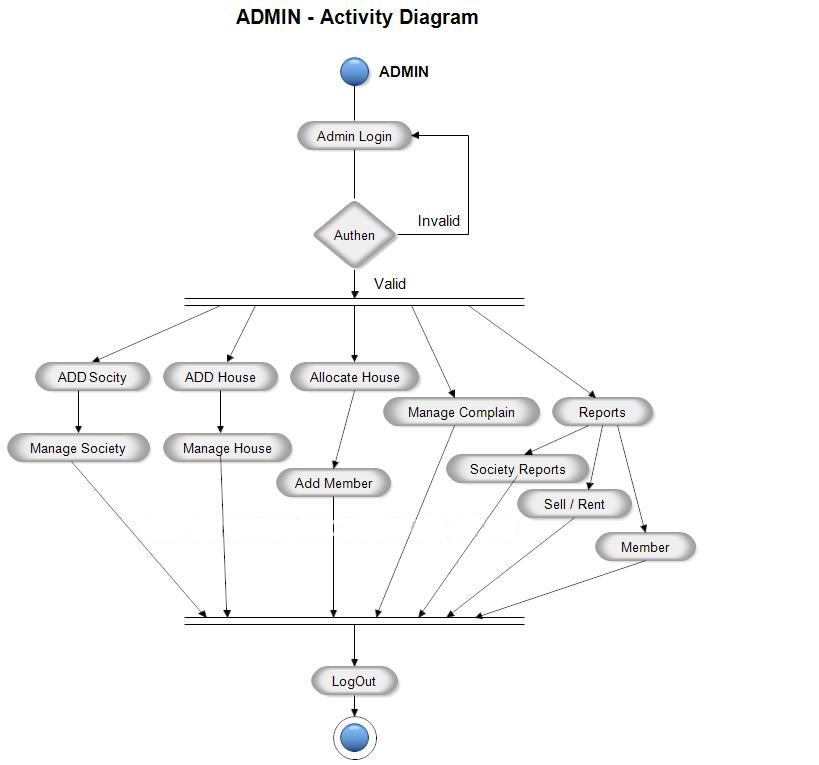
###### Purpose:

* Draw the activity flow of a system
* Describe the sequence from one activity to another.
* Describe the parallel, branched, concurrent flow of the system.

Before drawing an activity diagram, we need to identify the following elements:

* Activities
* Association
* Conditions
* Constraints

Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow.





### DFD DIAGRAM

A DFD diagram illustrates how data is processed by a system in terms of inputs and outputs.As it name indicates its focus is on the flow of information, where data comes from, where data goes, and how it gets stored.

###### Symbols:

There are essentially 2 different types

###### Process Notations:

A process transforms incoming data flow into outgoing data flow.



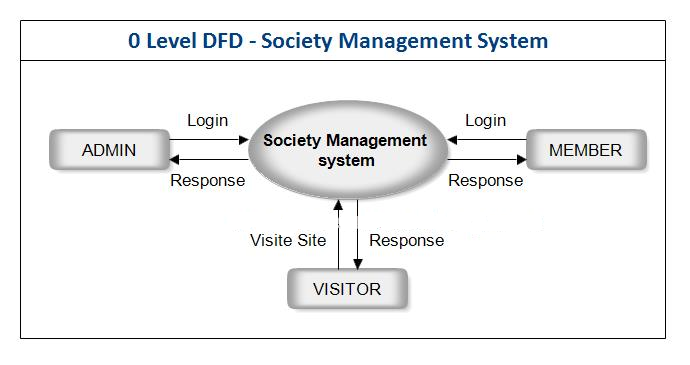
###### Datastores Notations:

Datastores are repositories of data in the system. They are sometimes referred to as

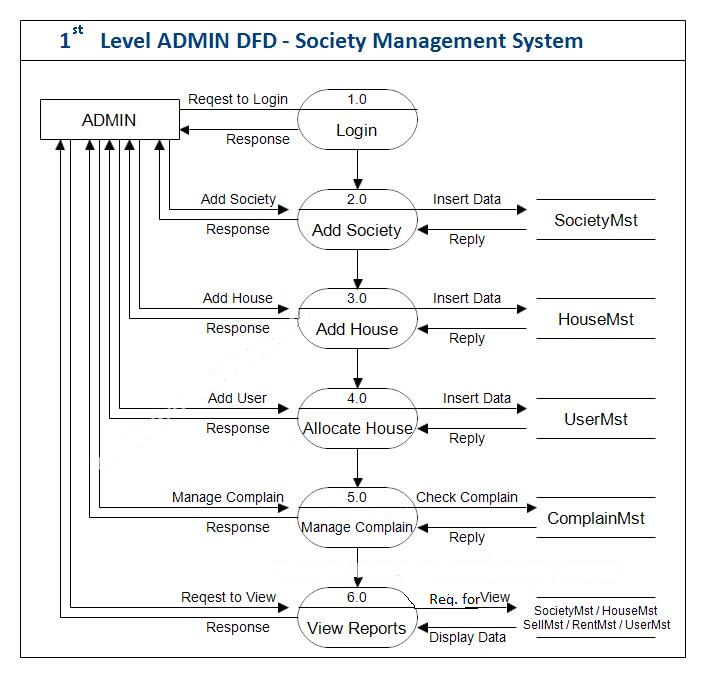
files.



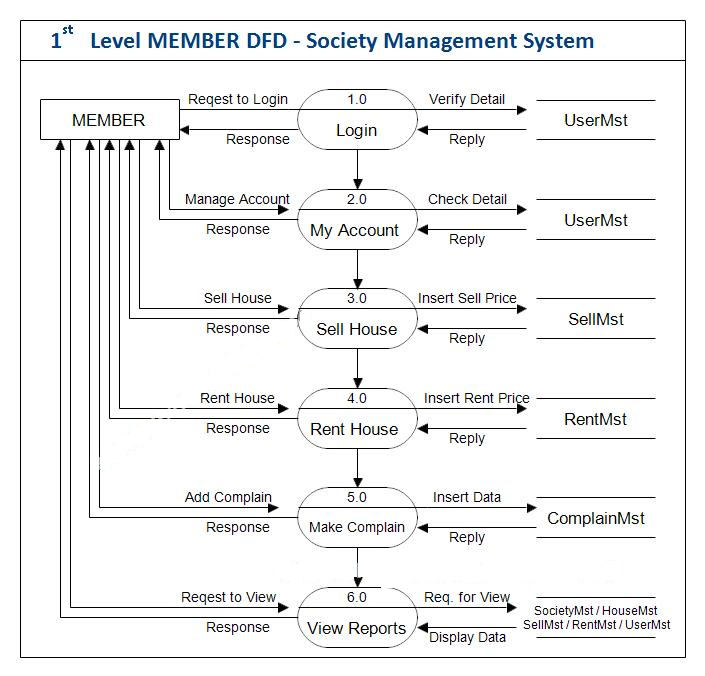
###### DFD Level 0



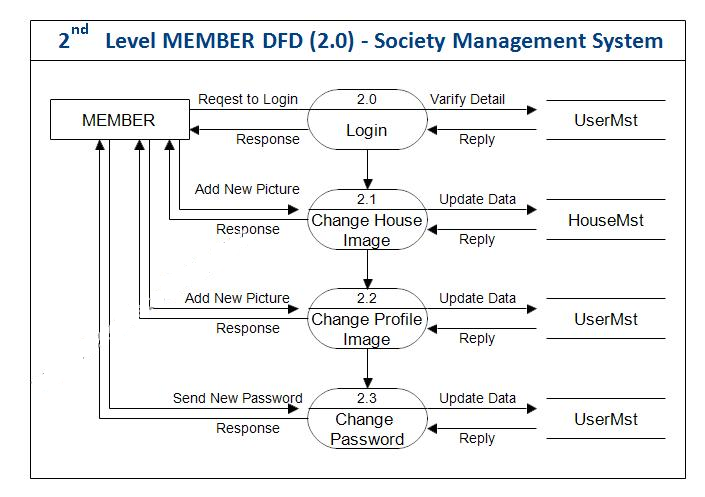
###### DFD Level 1 (ADMIN)



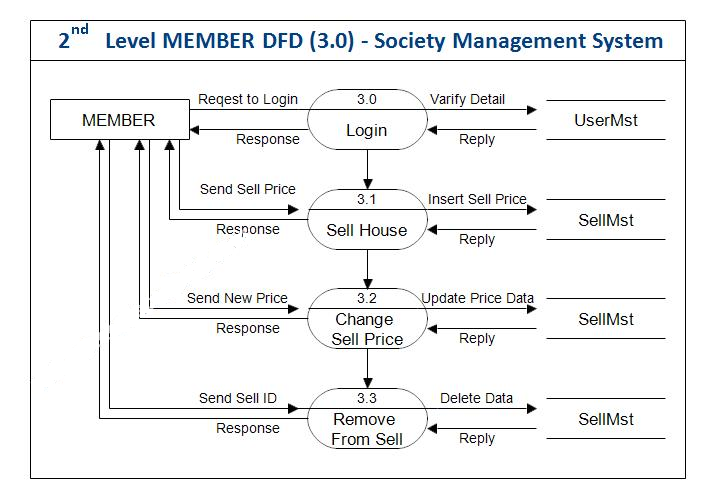
###### DFD Level 1 (MEMBER)



###### DFD Level 2 (MEMBER)



###### DFD Level 2 (MEMBER 3.0)



###### DFD Level 2 (MEMBER 4.0)

###### 

###### DFD Level 2 (MEMBER 5.0)

###### 

###### DFD Level 2 (ADMIN 5.0)

###### 

###### DFD Level 2 (MEMBER 6.0)

###### 

###### DFD Level 2 (MEMBER 6.0)

###### 

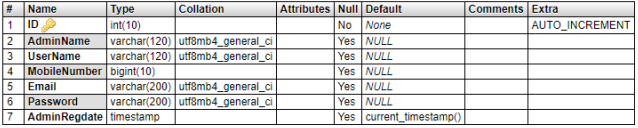
**DATABASE DESIGN**

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system. A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases

**Society Management System (SMS) contains 7 MySQL tables:**

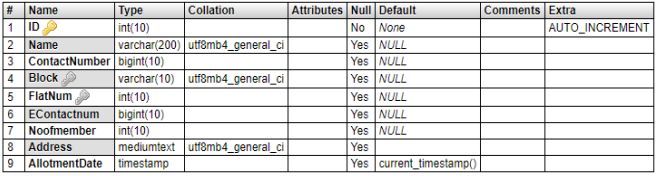
**tbladmin table Structure :**

**This table store the admin login and personal Details.**



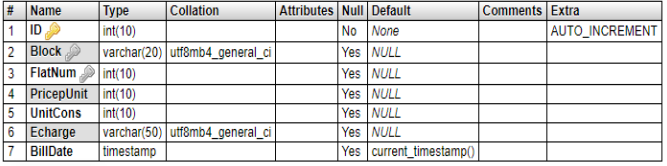
**tblallotment table Structure :**

**This table store the allotment detail of flats**



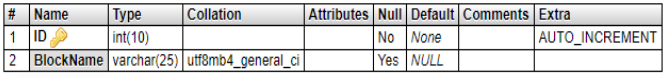
**tblbill table Structure :**

**This table store the bill detail of flats.**



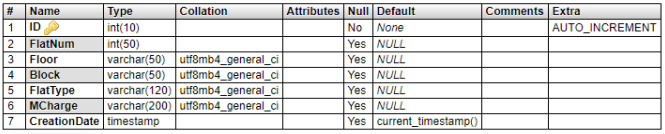
**tblblocks table Structure :**

**This table store the blocks of society.**



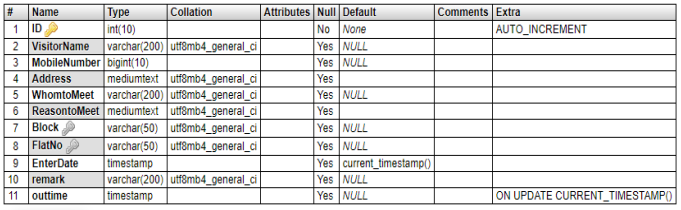
**tblflat table Structure :**

**This table store the flats details of society.**



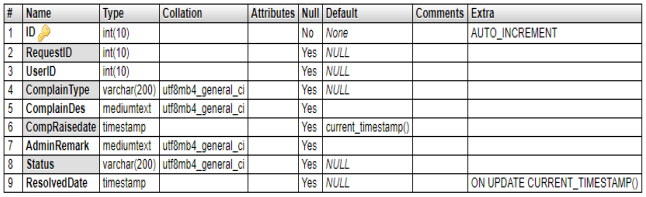
**tblvisitor table Structure :**

**This table store the details of visitors who visit in flats .**



**tblcomplain table Structure :**

**This table store the details of complains.**



* 1. **USER INTERFACE**

#### Homepage

#### 

#### 

#### **Admin Login**

#### 

#### **Forgot Password**

#### 

#### **Dashboard**

#### 

#### **Admin Profile**

#### 

#### **Change Password**

#### 

#### **Add Flats**

#### 

#### **Manage Flat**

#### 

#### **Update Flat**

#### 

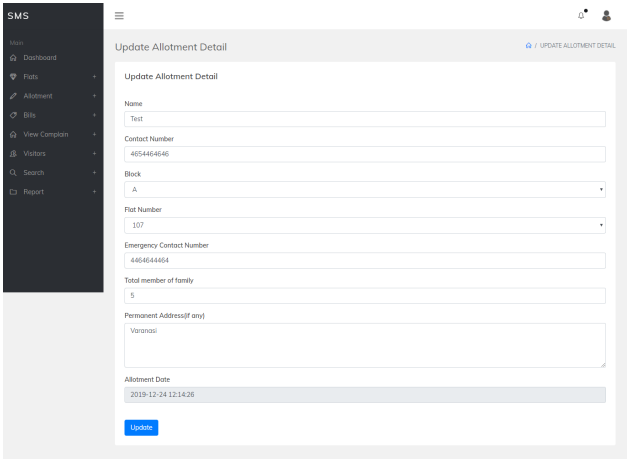
#### **Add Allotment**

#### 

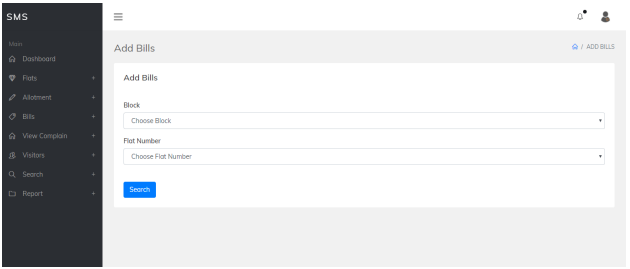
#### **Manage Allotment**

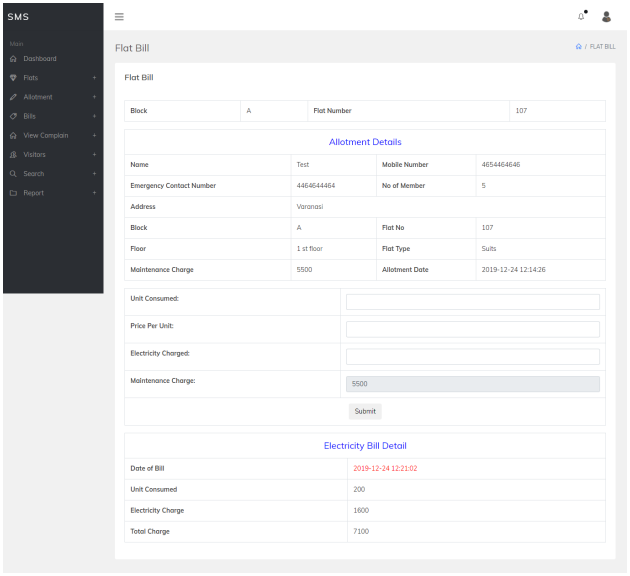
#### 

#### **Update Allotment**

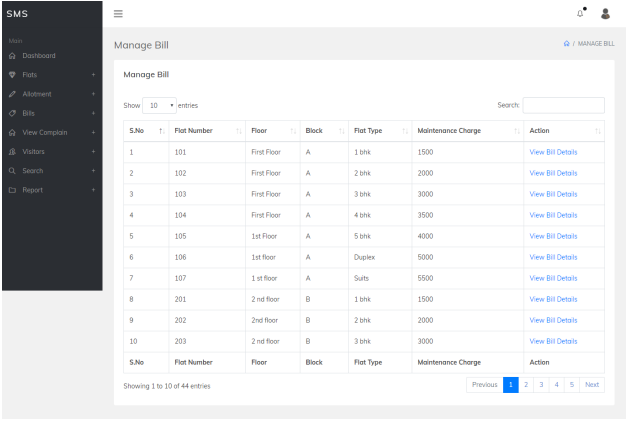


**Add Bills**

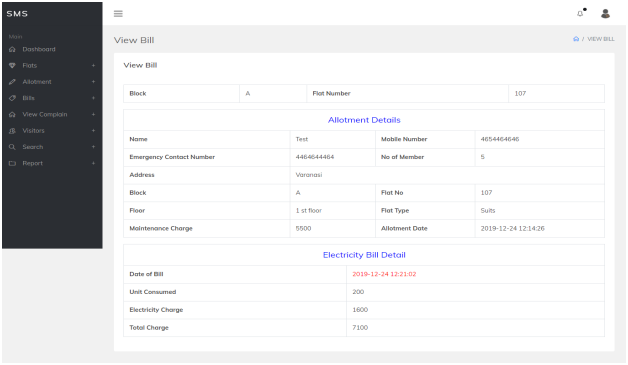




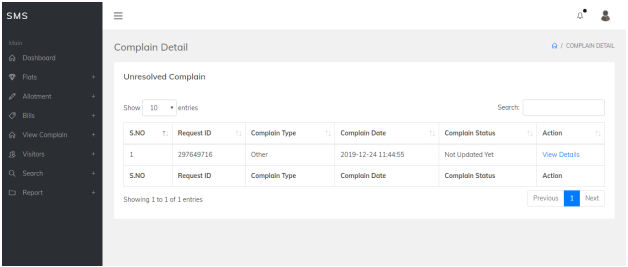
**Manage Bills**



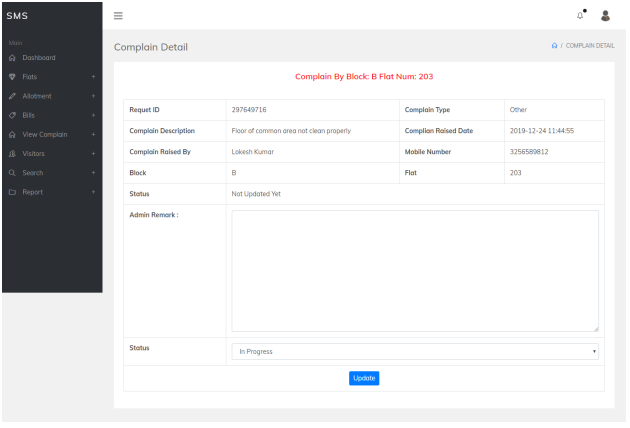
**View Bills**



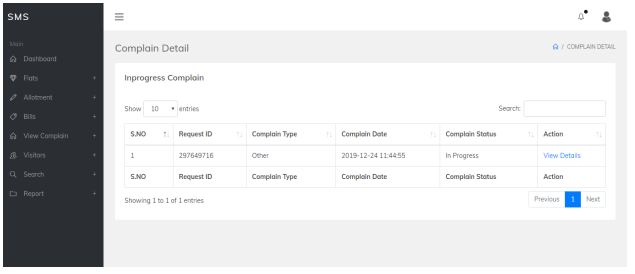
**Unresolved Complaint**



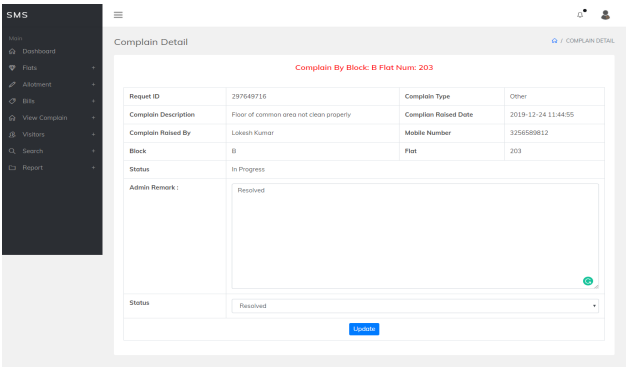
**View Unresolved complaint**



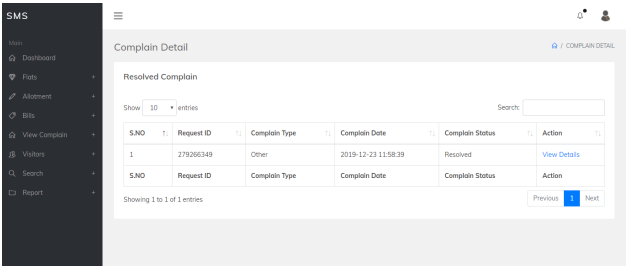
**In progress complaint**



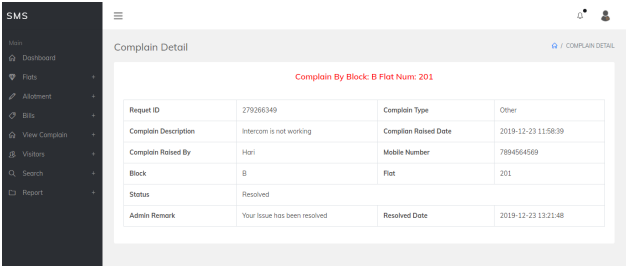
**View in progress complaint**



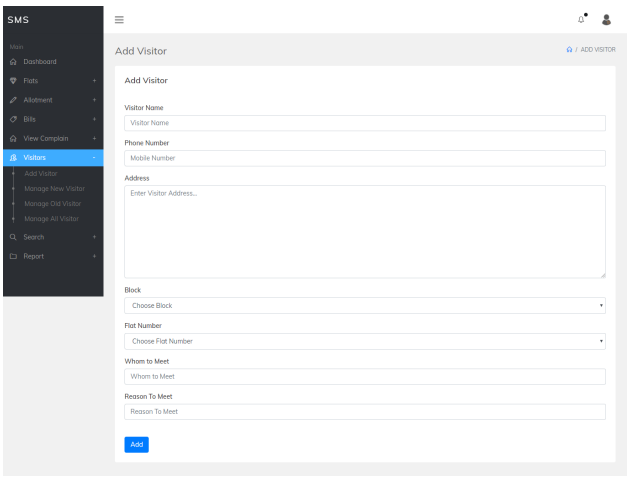
**Resolved Complaint**



**View Resolved Complain**



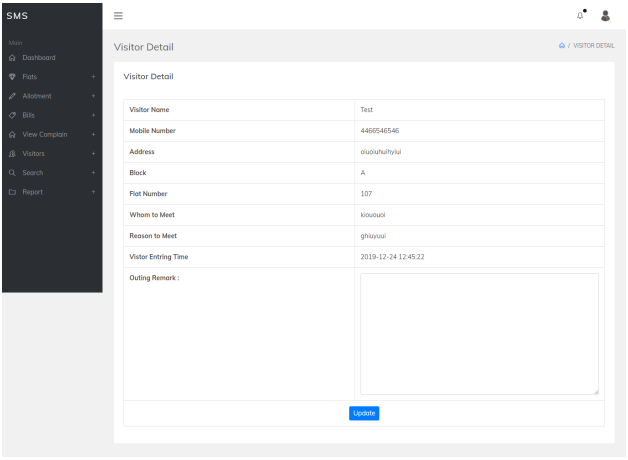
**Add Visitor**



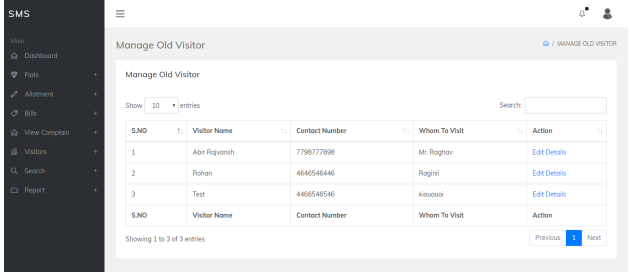
**Manage new visitor**



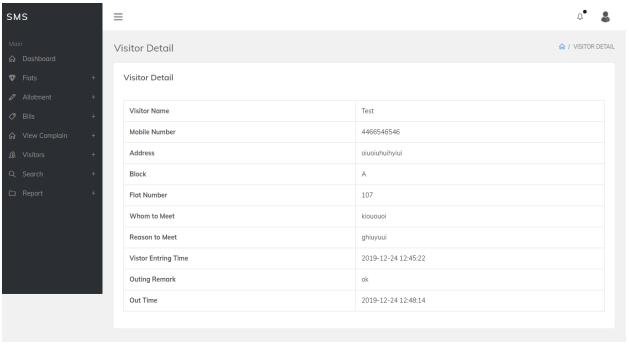
**Update New Visitor**



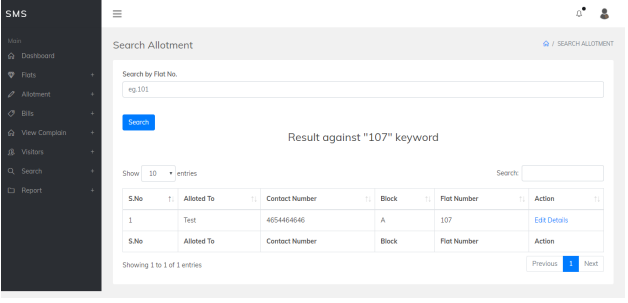
**Manage old visitor**



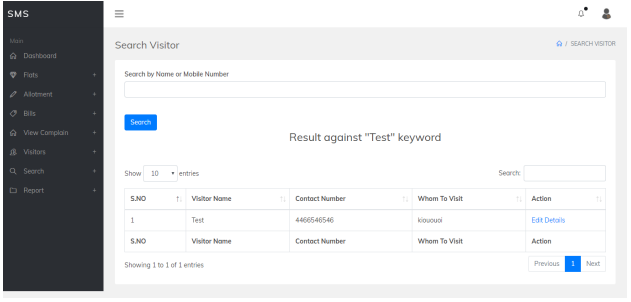
**View Old Visitor**



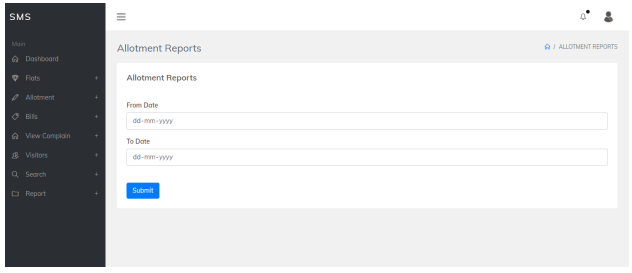
**Search Allotment**



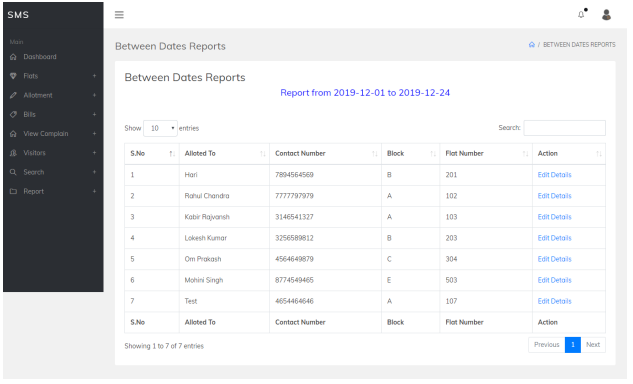
**Search Visitor**



**Allotment Reports**



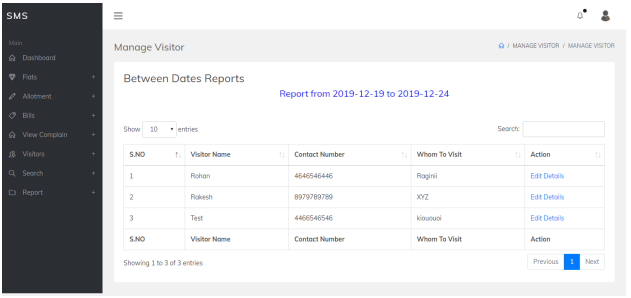
**View Between date reports of the allotment**

+

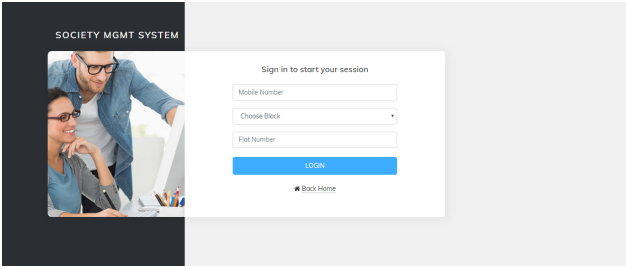
**Visitor Report**



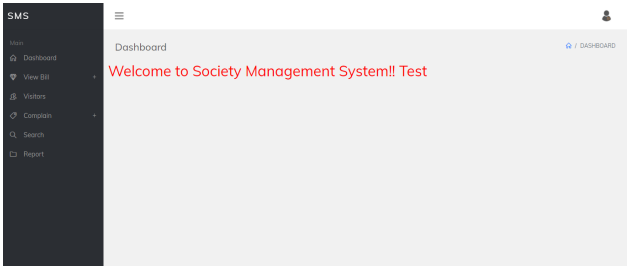
**View Between dates reports of the visitors**



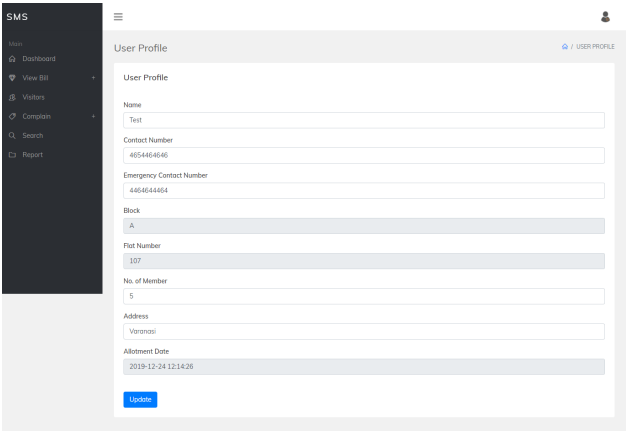
**User Login**



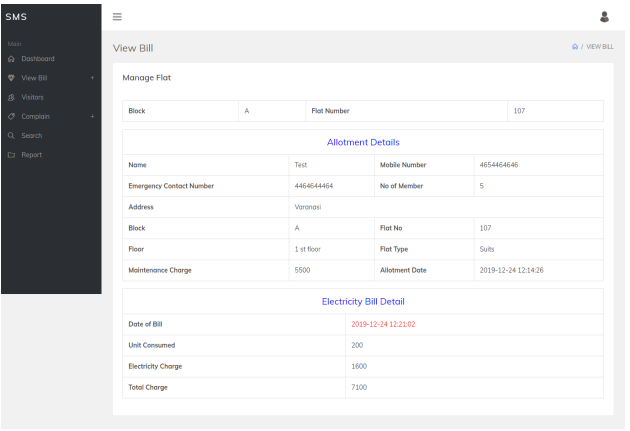
**Dashboard**



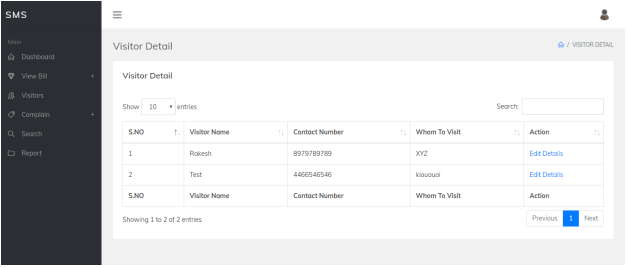
**User Profile**



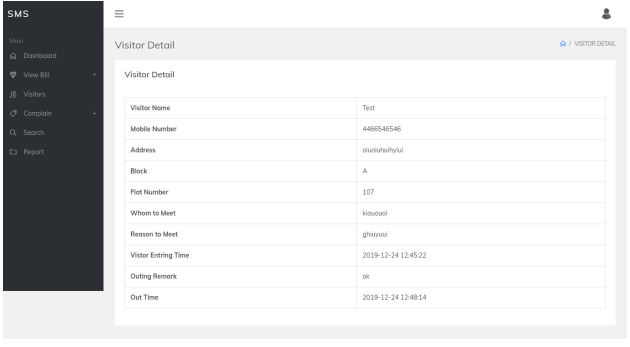
**View Bill**



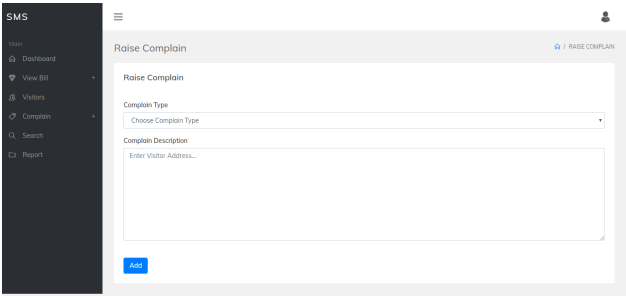
**View Visitor List**



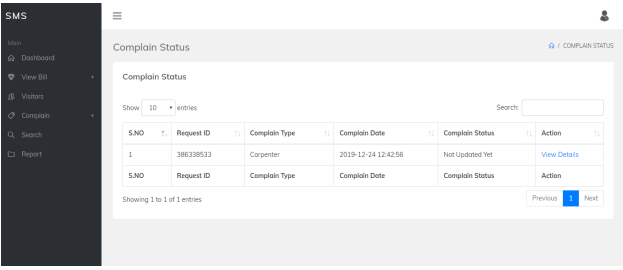
**Details of Visitor**



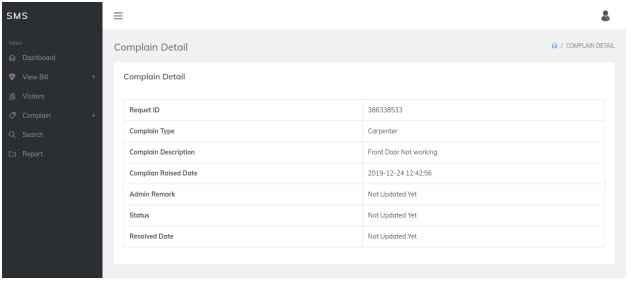
**Raise Complaint**



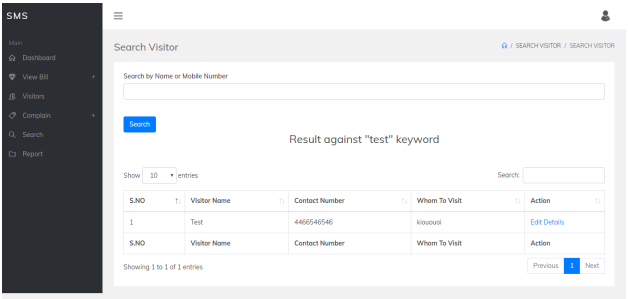
**Complaint Status**



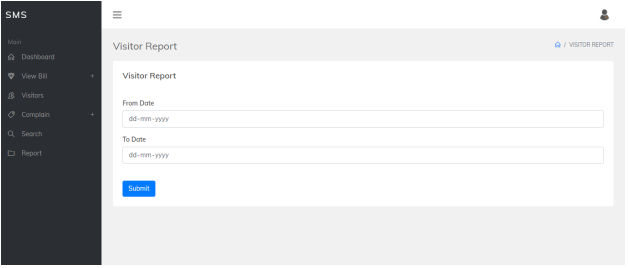
**View Detail of complaint**



**Search Visitor**



**Visitor Report**



**View Between dates and reports of visitors**

