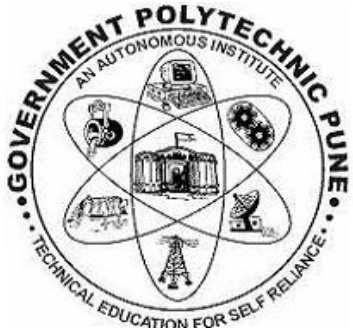
GOVERNMENT POLYTECHNIC, PUNE

(An Autonomous Institute of Government of Maharashtra)



DEPARTMENT OF COMPUTER ENGINEERING

ACADEMIC YEAR 2020-21

PROJECT REPORT ON

SOCIETY MANAGEMENT SYSTEM

Submitted By

1826211	**Siddharth Desai**
1826213	**Swapnil Dhamdhere**
1826204	**Smita Bansode**
1826206	**Prasad Bhavsar**

UNDER THE GUIDANCE

OF

Prof. N.R. Wagh

(DEPARTMENTOF COMPUTER ENGINEERING)

GOVERNMENT POLYTECHNIC, PUNE

(An Autonomous Institute of Government of Maharashtra)

CERTIFICATE

This is to certify that,

1826211	**Siddharth Desai**
1826213	**Swapnil Dhamdhere**
1826204	**Smita Bansode**
1826206	**Prasad Bhavsar**

of class Third Year (2020-21) have successfully completed Project on "**Society Management System**" under the guidance of **Prof. N. R. Wagh** in parallel fulfilment of the requirement for the award of Diploma in Computer Engineering from Government Polytechnic, Pune.

Prof. N. R. Wagh	**Prof. U.V. Kokate**	**Dr. V.S. Bandal**
(Project Guide)	(Head of Department)	(Principal)

ACKNOWLEDGMENT

The success of any project depends largely on the encouragement and guidelines of many others. This research project would not have been possible without their support. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

First and foremost, we wish to record our sincere gratitude to the Management of this college and to our Respected Principal Dr.V.S.Bandal, for his constant support and encouragement in the preparation of this report and for the availability of library facilities needed to prepare this report.

Our sincere thanks to Mr.U.V.Kokate, Head, Department of Computer, Govt. Polytechnic, Pune who was abundantly helpful and offered invaluable assistance, support and guidance.

We express our sincere gratitude to our Guide, **Prof. N.R. Wagh** for guiding us in investigations of this project and in carrying out experimental work. Our numerous discussions were extremely helpful. We are highly indebted to him for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project. We hold him in esteem for guidance, encouragement and inspiration received from him.

We wish to thank our parents for financing our studies and helping us throughout our life for achieving perfection and excellence. Their personal help in making this report and project worth presentation is gratefully acknowledged. Last but not the least we thank the Almighty for continuous strength we were bestowed for completion of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A housing society and its functions are a part of our everyday lives. Members of the managing committee are those who take charge of all the chores involved here. Residents often feel edgy regarding undone jobs and most even lack time to keep track of all these happenings. But, there's no avoiding the fact that these chores and all the housing society members are inseparable. This project presents the design and implementation of an Android based society management application. Daily life in city areas has important things to deal with society management. Our day to day needs such as Water supply, Electricity, Security for all of these maintenance and many more things which directly or indirectly plays the vital role in residential life, comes under Society Management. In most of the cases, Society management practices a traditional way of communication. This certainly has some limitations and disadvantages. Paying maintenance, cultural events, miscellaneous contacts, adding complaints for daily needs, security alerts, high priority communication and many others which may not be conveyed properly in current scenario as most of the things are getting handled manually. It lacks transparency. To overcome the problems occurring due to this time lagging manual system, an automated system needs to be developed to reduce the human efforts. Thus, our application will help users improve the management of their community and the tasks related to it.

Keywords: Society, Management, Maintenance, Time

Chapter 1
INTRODUCTION

1.1 INTRODUCTION

Generally, in Society all the work is decided in meetings and maintenance bills, contact no of members are recorded on the papers. There is no automated system for doing all the things that generally happens in society, so that members can come to know what is happening in society after admin add society related things to application through admin login. The Society Management System allows members to login with their own account and get updated with society happenings. Society Management System is the app to reduce conflicts among society members. The system has automated functionality for paying monthly maintenance bill from their account, Adding complaint related to our issues, Adding balance to your account in the login using debit card, Seeing upcoming events in the society all of these works can be done using one application which ultimately saves your time.

- This app allows user do society related work after login through mobile number and password.
- Connects members in one app and shows important announcements.
- This System is designed to used by the society secretary as admin and members as users with each other through the designed system.
- Provides every user separate login.

1.2 PROBLEM IDENTIFICATION

1.2.1 Problem Definition

- Our app will be the useful for the Society Management as members can do their work related to society using this app.
- They can also view what type of event going to be held in society in future.

1.2.2 Existing System

1.2.2.1 Study of existing system

- Complex and many bugs
- No backup facility
- No mail Security
- Not many apps like this are available

1.2.2.2 Limitations of Existing System

- They does not make user more Secure.
- Users cannot chat through this application.
- User cannot give reply to any notice made by admin.
- Admin login should be given first.

1.2.3 Need for New System

- Login to the system is Secure.
- User should be able to chat with other users through this application.
- To make main or mobile no authentication more secure.
- All users using the system are listed to the system.

1.3 PROJECT OBJECTIVE

- This system is design for a management purpose so that they can be able to share all the information safely between the society members and with other related nodes.
- This system will be usable for the Society Management only.
- No other unauthority person cannot access the data of admin login, the data in the system is secure by using database.
- The system provide more creative control to the Admin(Secretory).

1.4 PROPOSED SYSTEM AND METHODOLOGY

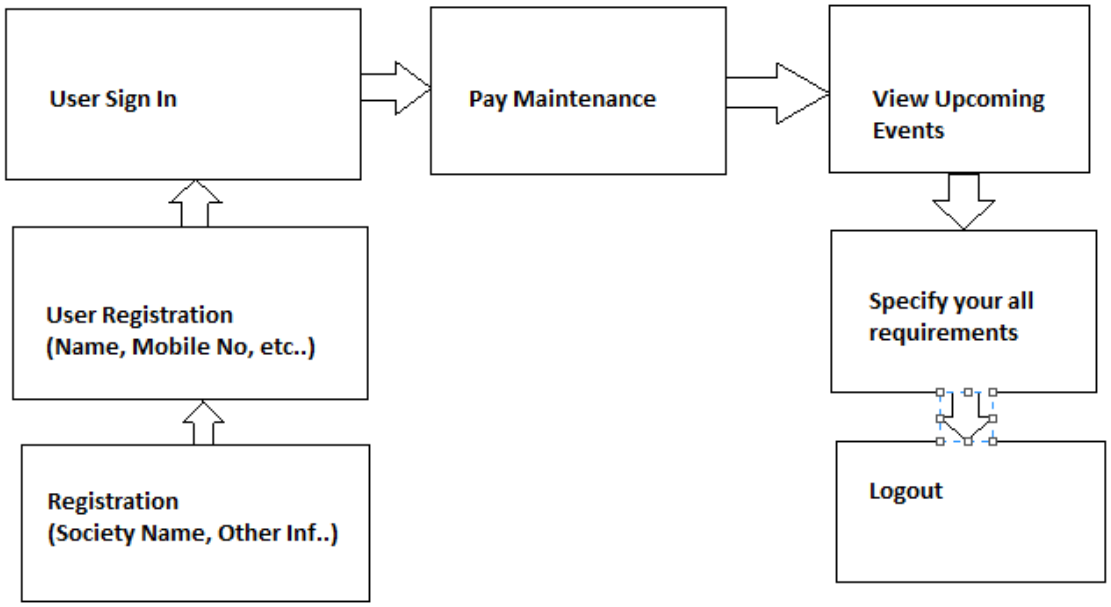


Figure 1.1 Modules of the System

1.5 APPLICABILITY

- This system will be usable for the working people to do their society related work within less time.
- This system is designed for a society management that the admin can be able to share the valuable information safely between the members and other related nodes .
- The system provide database where all data is stored securely.
- No other unauthority person cannot access the data, the data in the system is secure.

Chapter 2

PROJECT PLANNING & SCHEDULING

2.1 SOFTWARE MODEL

- For this project, we use iterative model.
- Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented.
- At each iteration design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through is to develop a system through repeated cycles and in smaller portions at a time.

2.2 APPROACH IN PROJECT DEVELOPMENT

The project is developed following the SMART approach.

- ❖ Specific
- ❖ Measurable
- ❖ Achievable
- ❖ Realistic
- ❖ Time-bound

2.3 GOALS

The following goals are achieved by this project plan:

1. Software risks are documented for use in planning and tracking the software project.
2. Software project activities and commitments are planned and documented.
3. Affected groups and individuals agree to their commitments related to the software project.
4. Project is scheduled and documented.
5. Gives the desired output.

2.4 PROJECTSCOPE

For the scope of the project, the project will be tested as the program is being developed. A database for the users registered will be developed and tested, a Login will be

developed and tested for the user's benefits. When the society management system application is near completion, more testing will be done in order to make it less buggy or more user friendly.

2.5TEAM STRUCTURE

Team structure addresses the issue of organization of the individual project teams. Our project team consists of **Four** members; the efforts assignment to each team member are given the project table, the role of each member is as below:

Table 2.1 Team Structure, Roles & Details

Sr. No.	Name of Team Member	Role in Project	Email ID
1.	Siddharth Desai	Coding, Testing, Documentation	siddharthdesai1710@gmail.com
2.	Swapnil Dhamdhere	Database, Coding,Testing, Documentation	dhamdhereswapnil31@gmail.com
3.	Smita Bansode	Design,Coding,Testing, Documentation	smitabansode80@gmail.com
4.	Prasad Bhavsar	Design,Coding,Testing, Documentation	prasadbhavsar567@gamil.com

2.6 PROJECT SCHEDULE

Project table is a tabular listing of all project tasks, their planned and actual start- and end-dates, and a variety of related information. Used in conjunction with the timeline chart, project tables enable the project manager to track progress.

Table 2.2 Project Table

Event Name	Start Date	Actual Start Date	End Date	Actual End Date	Effort Assignment
Feasibility Study 1. Study of project domain/ project area. 2. Evaluation and analysis of the proposed project. 3. Scope of a proposed project.	17/04/ 2021	**17/04/ 2021**	20/04/ 2021	**20/04/ 2021**	All
Requirement Gathering and Analysis 1. Gather Customer / user Requirements 2. Analyze user requirement 3. Decide System Requiements(consider Hardware, OS, DBMS and other developer's requirements)	21/04/ 2021	**21/04/ 2021**	29/04/ 2021	**29/04/ 2021**	All
Design 1. Design structure of proposed system 2. Decide input, processing, and output of the proposed system. 3. Design modules required and interfaces between modules 4. Identify third party interfaces, if required 5. Design User Interface 6. Design database structure 7. E-R Diagram	01/05/ 2021	**01/05/ 2021**	06/05/ 2021	**06/05/ 2021**	All
Coding 1. Develop GUI, various Interfaces, and 2. Programs / code 3. Create database and database connectivity, if required	07/05/ 2021	**07/05/ 2021**	22/06/ 2021	**22/06/ 2021**	All
Testing 1. Installation of product 2. Dummy data entry 3. Testing 4. Bug reporting and rectification 5. Regression testing	18/02/ 2021	**18/02/ 2021**	23/06/ 2021	**23/06/ 2021**	All
Project- Documentation 1. Prepare Estimates, Results, Update Report 2. Writing User Manual for the system	18/06/ 2021	18/06/ 2021	23/06/ 2021	**23/06/ 2021**	All

Chapter 3

REQUIREMENT ANALYSIS

3.1 MINIMUM HARDWARE REQUIREMENTS

Table 3.1 Minimum Hardware Requirements

Hardware	Minimum Requirement
Processor	3.20 GHz
Primary Memory	8 GB RAM
Secondary Memory	HardDisk (128 GB)
Internet Connection	1-2 Mbps
Other Hardware	Smartphone ,USB Cable

3.2 MINIMUM SOFTWARE REQUIREMENTS

Table 3.2 Minimum Software Requirements

Role	Software	Minimum Requirement
Development	**Platform (OS)**	Windows 8.1 and any other.
	Front End (Prog. Lang.)	Java
	Backend (DB)	My SQL
	Development Tool (IDE)	Android Studio 4.2
	Testing Tool	Selenium IDE
Deployment	**Execution Environment**	Android OS (6.0 onwards)
	Browser	Chrome for viewing database
	Server (Application / Database Server)	XAMPP Server
Design	**UML Design**	Draw.io

	DFD, ER, Flows	Draw.io

3.3 ANDROID

Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets

3.4 MYSQL Database :

The MySQL Database lets you build rich, collaborative applications by allowing secure access to the database by creating tables from client-side code. Data is persisted locally, real-time events continue to fire, giving the end user a responsive experience

Chapter 4
SYSTEM DESIGN

4.1 INTRODUCTION

Design uses a combination of text and diagrammatic forms to depict the requirements for data, function and behavior in a way that is relatively easy to understand and more important, straightforward to review for correctness, completeness and consistency.

A diagram is the graphical presentation of a set of elements most often rendered as a connected graph of vertices (things) and arcs (relationship). These diagrams are drawn to visualize a system from different perspectives so a diagram into a system.

4.2 UML MODELING

The unified modeling language (UML) is a Graphical Language for visualization, Specifying, construction and documenting the artifacts of a software intensive system. The UML gives a standard was to write system‟s blue prints, covering conceptual thing, such as Business Processes & system functions, As well as concrete things, such as classes written in a specific programming language, database schemas, and reusable software components.

4.2.1 Use Case Diagram

A use case defines behavioral features of a system. Each use case is named using a verb phase expresses a goal of the system. A use case diagram shows a set of use cases and actors &their relationships. Use case diagrams address the static use case view of a system. These diagrams are especially important in organizing and modeling the behaviors of a system. It shows the graphical overview of functionality provided by the system intents actor.

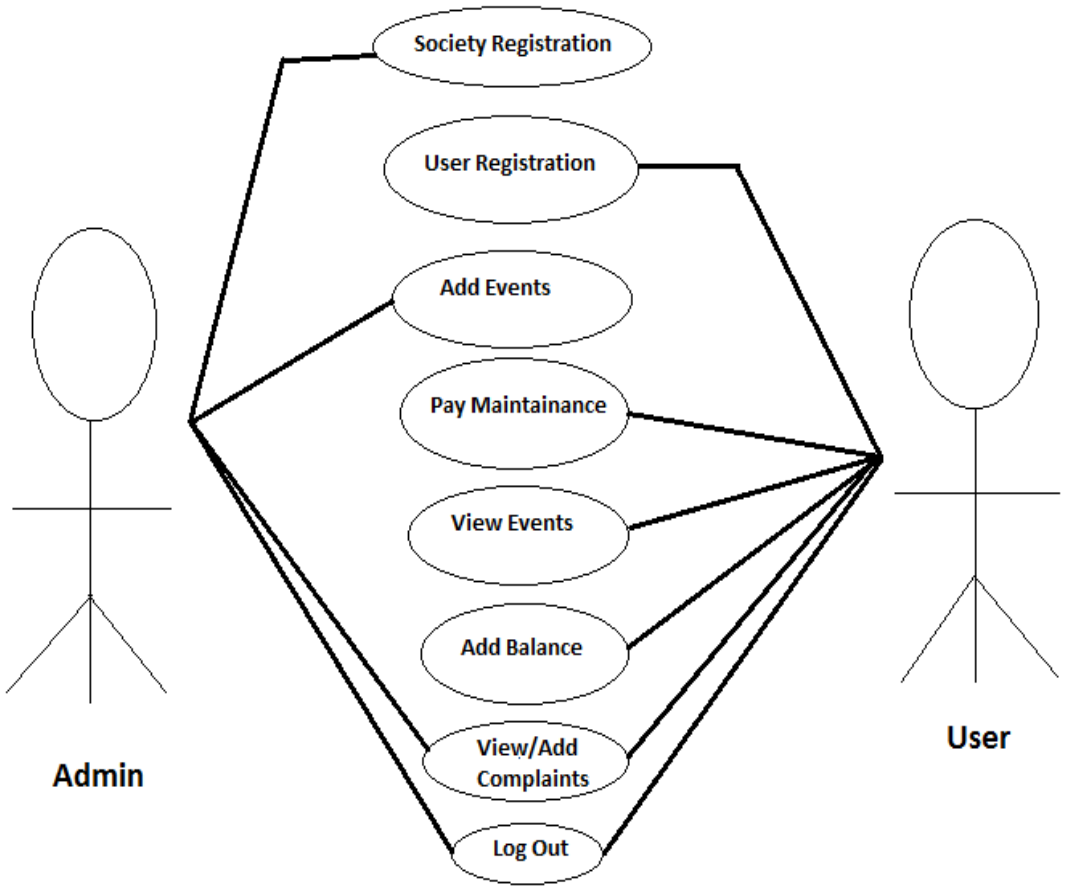


Figure 4.1 Use case Diagram

4.2.2 Activity Diagram

An activity diagram of a special kind of a state chart diagram that shows the flow from activity within a system. An activity addresses the dynamic view of a system. The activity diagram is often seen as part of the functional view of a system because it describes logical processes, or functions. Each process describes a sequence of tasks and the decisions that govern when and they are performed. The flow in an activity diagram is driven by the completion of an action.

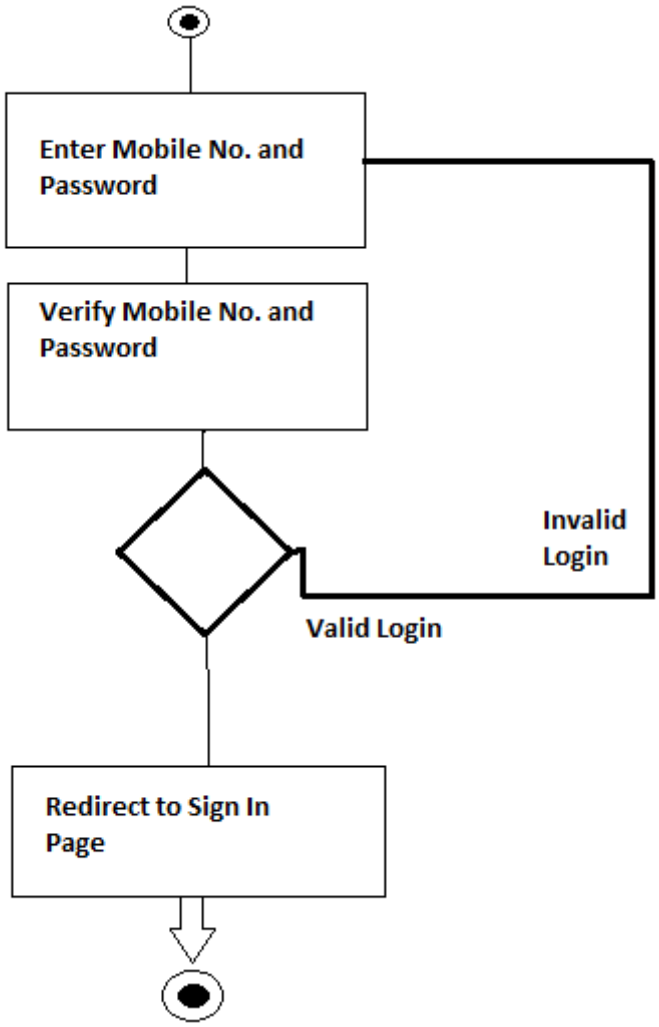


Figure 4.2 Activity Diagram For Sign In

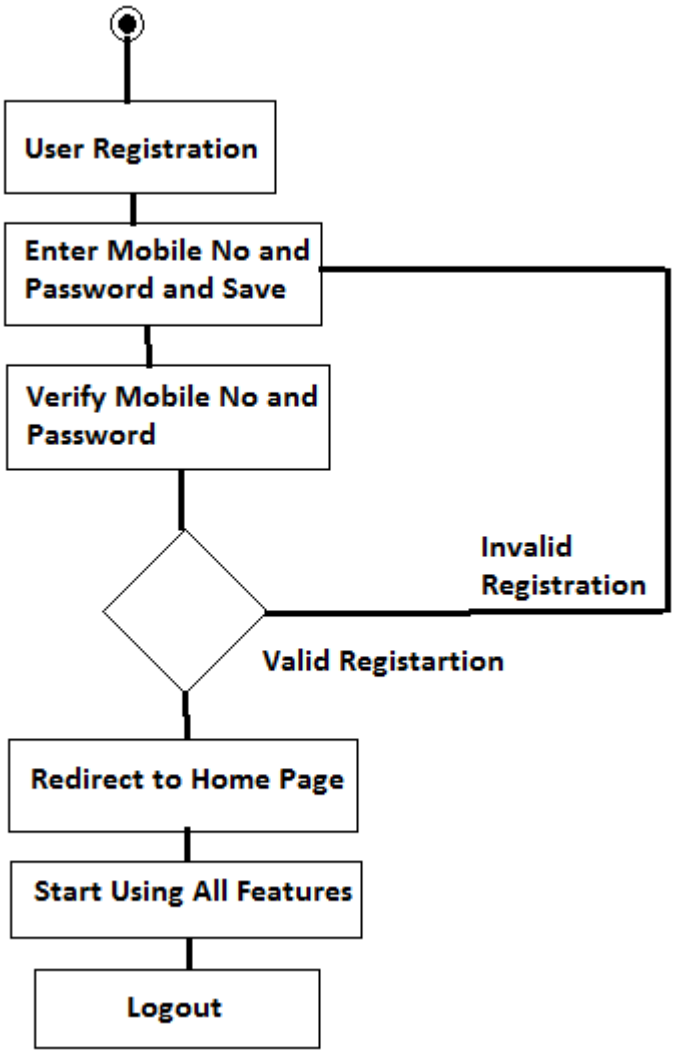


Figure 4.3 Activity Diagram For User Registration and Use.

4.3 ANALYSIS MODELING

4.3.1 Data Modeling(Entity - Relationship Diagram)

Entity-Relationship diagram (ERD) is a graphical technique, which is used to represent entities present in the system, and relationship those are applied between these entities.

The entity-relationship (E-R) data model is based on a perception of a real world that consists of a collection of basic objects, called **entities**, and of **relationships** among these objects.

An entity is a “thing” or “object” in the real world that is distinguishable from other objects. For example, each person is an entity, and bank accounts can be considered as entities. Entities are described in a database by a set of **attributes**. A **relationship** is an association among several entities.

The overall logical structure (schema) of a database can be expressed graphically by an E-R diagram, which is built up from the following components:

- **Rectangles,** which represent entity sets
- **Ellipses,** which represent attributes
- **Diamonds,** which represent relationships among entity sets
- **Lines,** which link attributes to entity sets and entity sets to relationships
- Each component is labeled with the entity or relationship that it represents.

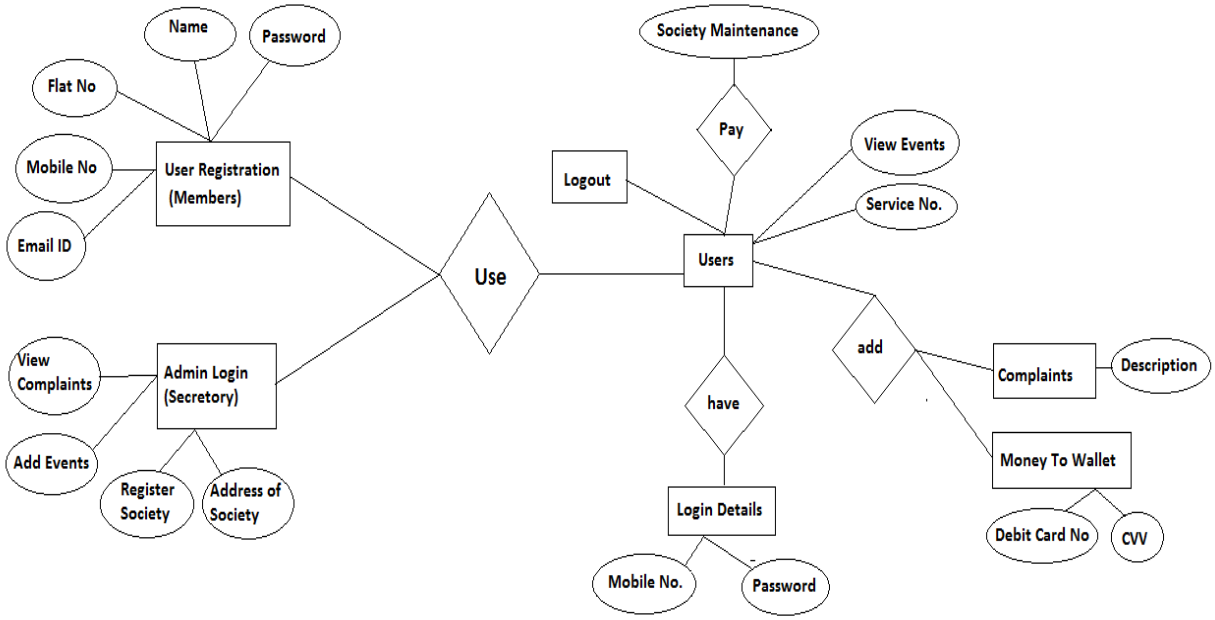


Figure 4.5 Entity-Relationship Diagram

4.3.2 Functional Modeling(Data Flow Diagram)

Data flow diagram (DFD), also called as „Bubble chart" is a hierarchical (or leveled) set of diagrams, used to represent the flow of data elements into and out of the functional units of the program, data stores, environmental sources and sinks.

The data flow diagram (DFD) serves two purposes:(1) to provide an indication of how data are transformed as they move through the system and **(2)** to depict the functions (and sub-functions) that transform the data flow.

The data flow diagram may be used to represent a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail.

A level 0 DFD, also called a fundamental system model or a context model, represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. Additional processes (bubbles)

and information flow paths are represented as the level 0 DFD is partitioned to reveal more detail. For example, a level 1 DFD might contain five or six bubbles with interconnecting arrows. Each of the processes represented at level 1 is a sub-function of the overall system depicted in the context model.

4.3.2.1 Data Flow Diagram - Level 0

Figure 4.6 Data Flow Diagram level-0

4.3.2.3 Data Flow Diagram - Level 1

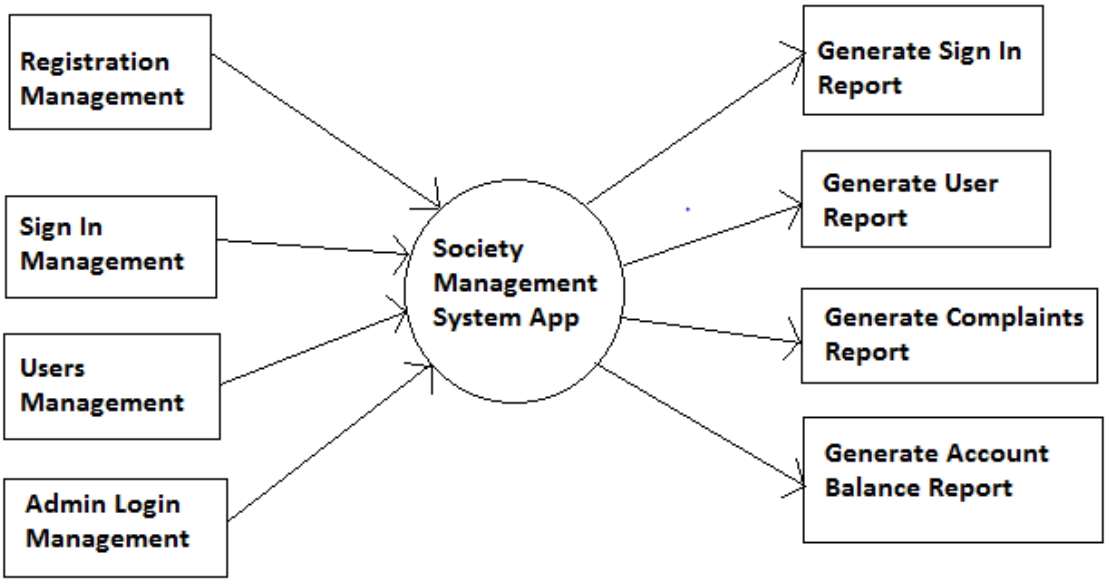


Figure 4.7 Data Flow Diagram level-1

4.3.2.3 Data Flow Diagram - Level 2

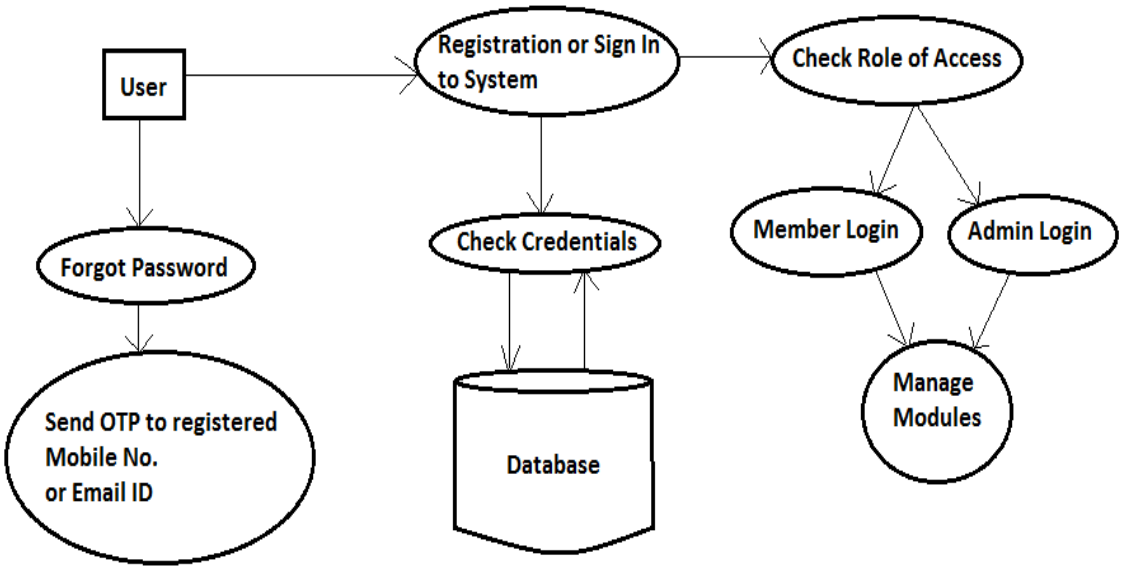


Figure 4.8 Data Flow Diagram level-2

Chapter 5

IMPLEMENTATION DETAILS

5.1 IMPLEMENTATION LANGUAGE – Java

Java

Java is a set of computer software and specications developed by Sun Microsystems, which was later acquired by the Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. Java applets, which are less common than standalone Java applications, run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

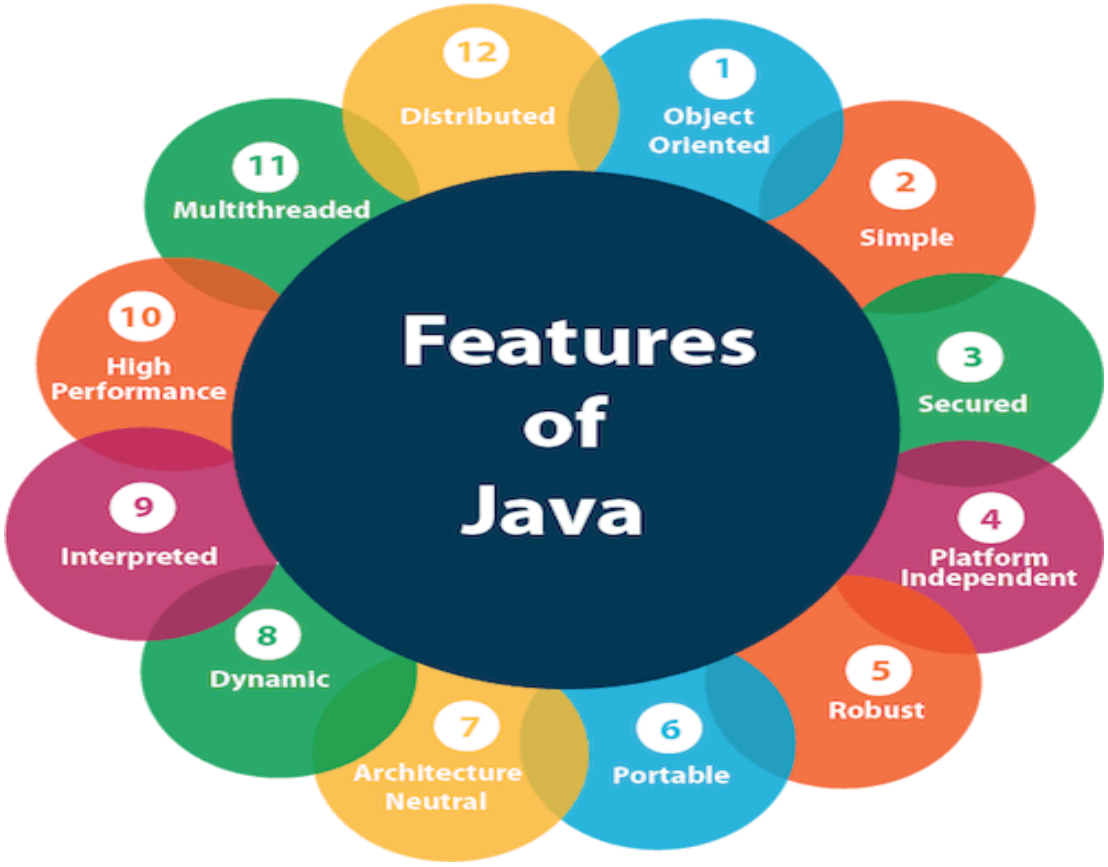
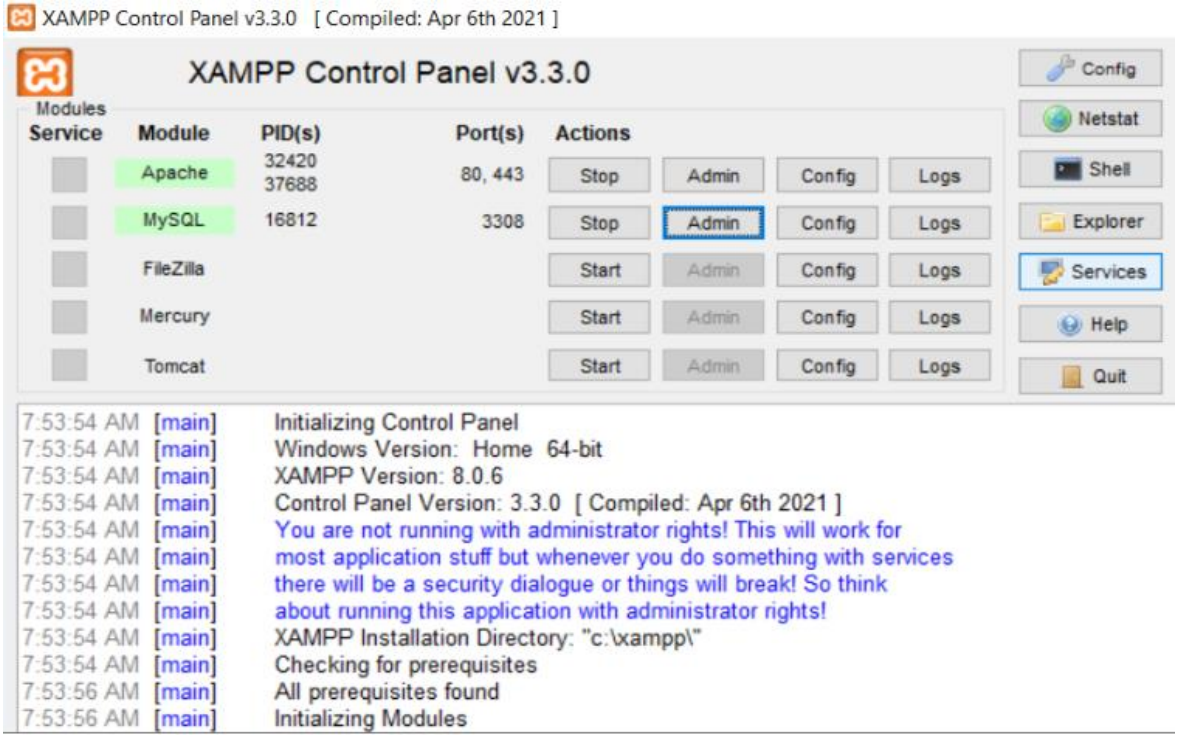


Figure 5.1 Features of Java

5.2 DATABASE – MySQL using XAMPP

The system uses MySQL to store the database required for the Society Management System. When input is given by the user, values are stored in the appropriate tables and then retrieved as it needs the data from the table. The system uses the following tables to store the data..



5.2.1 XAMPP Control Panel

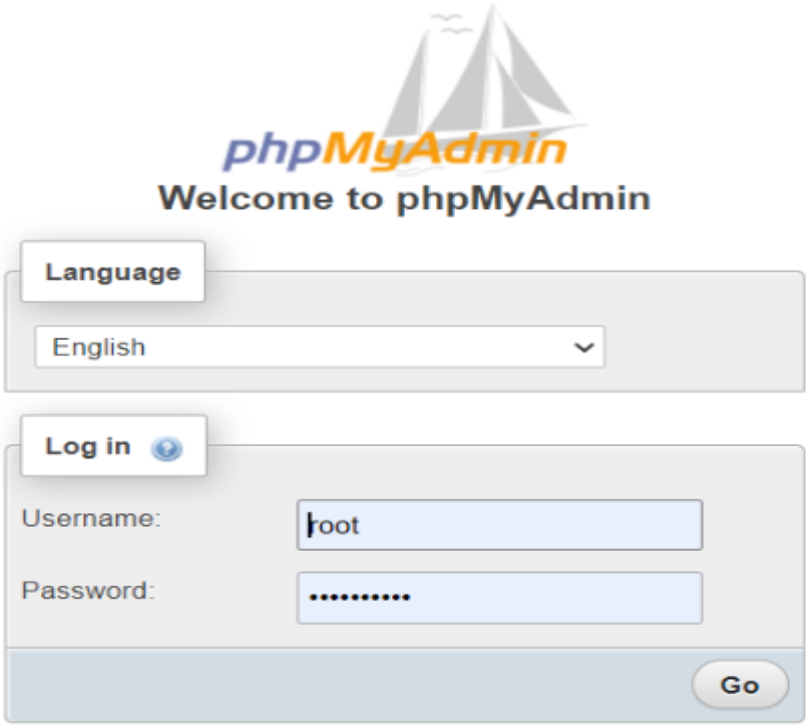
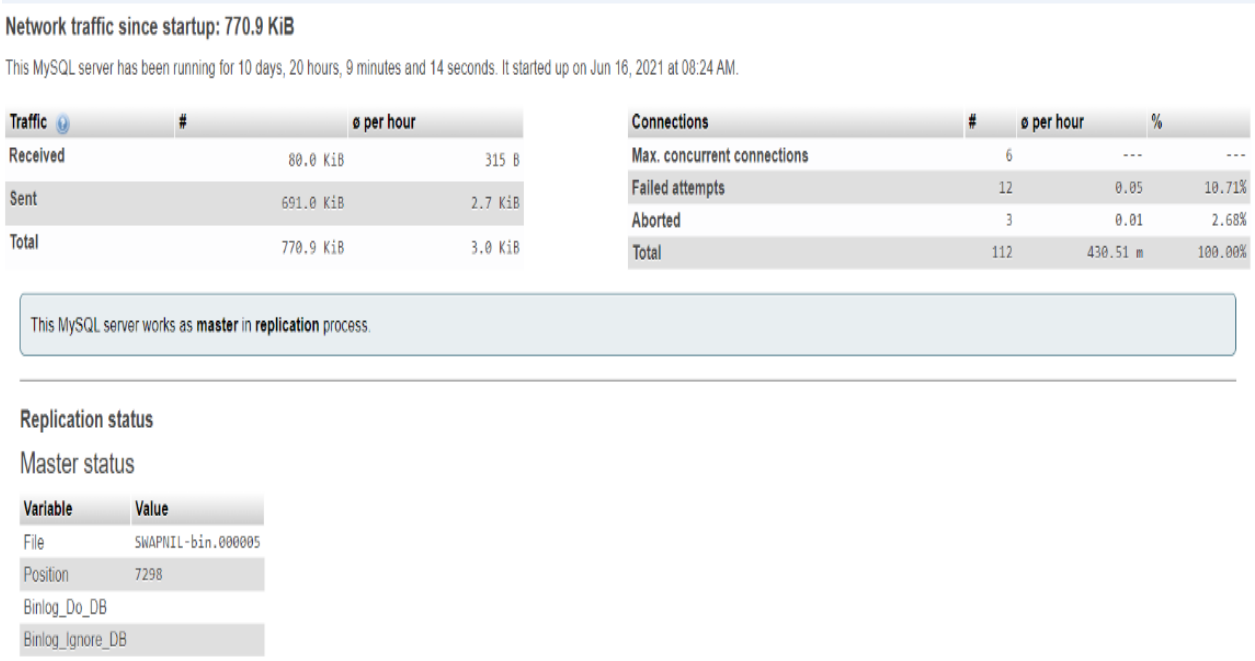


Figure5.2.2 Php MyAdmin Login



Network traffic since startup: 770.9 KiB

This MySQL server has been running for 10 days, 20 hours, 9 minutes and 14 seconds. It started up on Jun 16, 2021 at 08:24 AM.

Traffic	#	ø per hour
Received	80.0 KiB	315 B
Sent	691.0 KiB	2.7 KiB
Total	770.9 KiB	3.0 KiB

Connections	#	ø per hour	%
Max. concurrent connections	6	---	---
Failed attempts	12	0.05	10.71%
Aborted	3	0.01	2.68%
Total	112	430.51 m	100.00%

This MySQL server works as **master** in **replication** process.

Replication status

Master status

Variable	Value
File	SWAPNIL-bin.000005
Position	7298
Binlog_Do_DB	
Binlog_Ignore_DB	

Figure5.2.3 Server Status

Table structure | Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Id	int			No	None		AUTO_INCREMENT	Change Drop More
2	User_full_name	varchar(30)	utf8mb4_0900_ai_ci		No	None			Change Drop More
3	User_Pass	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More
4	Mobile	bigint			No	None			Change Drop More
5	User_Add	text	utf8mb4_0900_ai_ci		No	None			Change Drop More
6	City	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More
7	Pincode	int			No	None			Change Drop More
8	Email	text	utf8mb4_0900_ai_ci		No	None			Change Drop More
9	Cardno	bigint			No	None			Change Drop More
10	Amount	int			No	0			Change Drop More
11	Emergencyno	bigint			No	None			Change Drop More
12	Society	varchar(20)	utf8mb4_0900_ai_ci		No	None			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

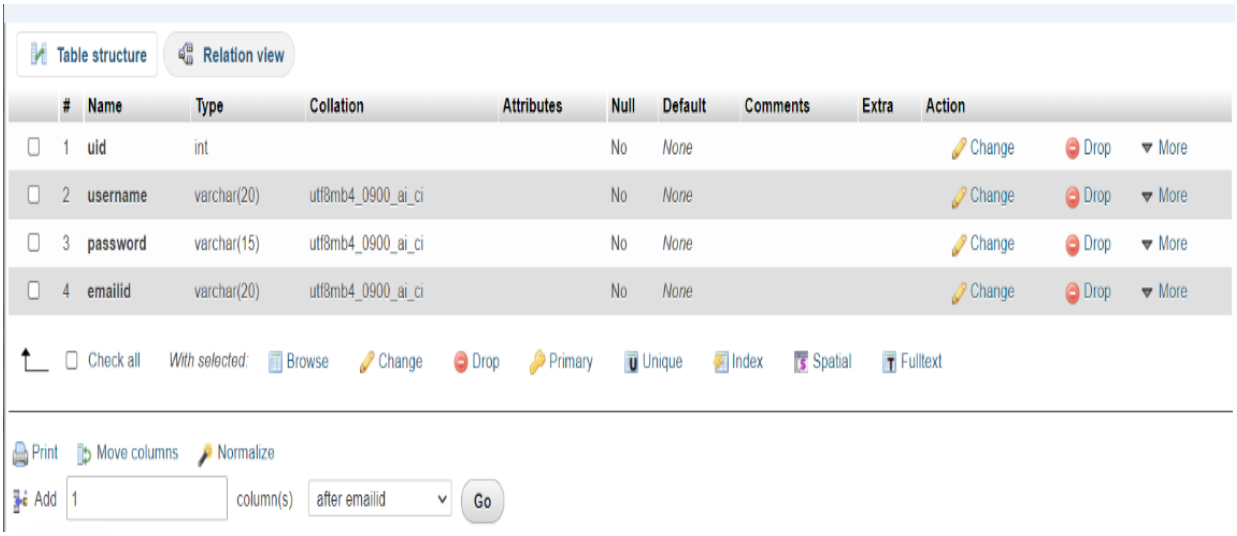


Figure5.2.4 Database Table Structure

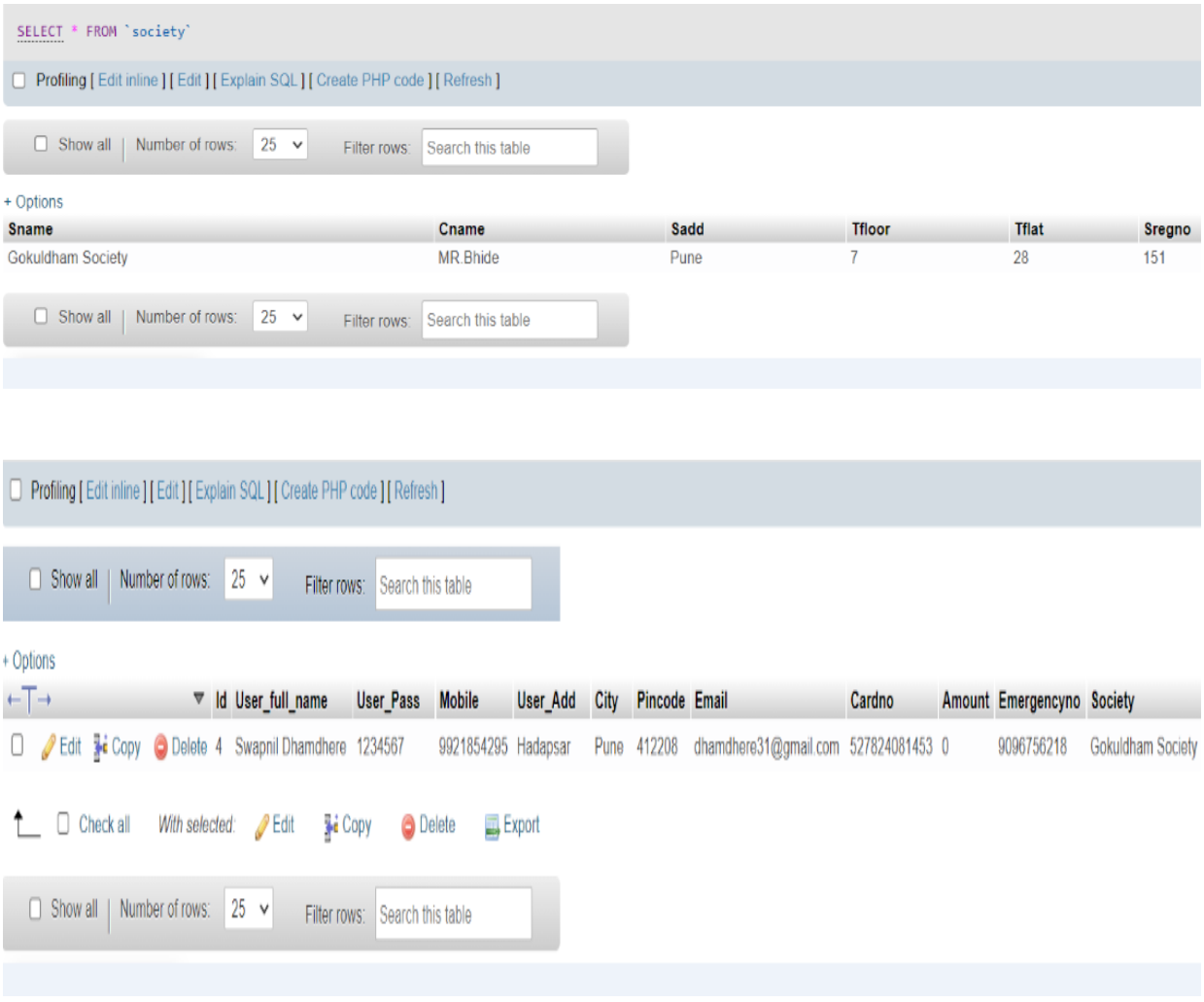


Figure5.2.5 Storage

5.2.1 Database Schema

Databases change over time as information is inserted and deleted. The collection of information stored in the database at a particular moment is called an ***instance*** of the database. The overall design of the database is called the ***database schema***. Schemas are changed rarely, if at all.

The concept of database schemas and instances can be understood by analogy to a program written in a programming language. A **database schema** corresponds to the variable declarations (along with associated type definitions) in a program. Each variable has a particular value at a given instant. The values of the variables in a program at a point in time correspond to an instance of a database schema. Schema is the **logical structure** of the database (e.g., set of customers and accounts and the relationship between them). The schema displays the structure of each record type but not the actual instances of records.

A. Table: Users Table

Table 5. 5.9 Users Table

Sr No	Field_Name	Data_type	Description
1.	User Id	Int	Stores the unique id for each user.
2.	Name	Text	Stores the name of the user.
3.	Society Name	Text	Stores the name of society.
4.	User Email Id	Varchar	Stores the email id of user.
5.	User password	Varchar	Stores the password for particular user.
6.	User type	Text	Stores the type of user.

B.Table: Admin Table

Table 5. 5.10 Admin Table

Sr No	Field_Name	Data_type	Description
1.	Admin Id	Int	Stores the Email id of admin.
2.	Admin Password	Varchar	Stores the password of admin.
3.	User and complaints	Text	Stores the user name and complaints by that user.
4.	Society name	Text	Stores the society name of user which make complaint.

C.Table: Payments Table

Table 5. 5.11 Payments Table

Sr No	Field_Name	Data_type	Description
1.	User Name	Text	Stores the name of user who pays maintenance.
2.	Maintenance to pay	Int	Stores money to be paid by user.
3.	Add balance to wallet	Int	Stores the money added to wallet account.

D.Table:Fields

Table 5. 5.12 Fields

Field	Type
Mobile No	Int
Add Complaint	String
Add Balance	Int
Pay Maintenance	Int
Password	Varchar

5.3 IMPLEMENTATION TOOL – Android Studio 4.2

Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at theGoogle I/Oconference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0.The current stable version is 4.2, which was released in April 2021.

.

5.5 FORM DESIGN

5.5.1 Snapshots

Figure5.5.1. Home Page

5.5.1.1 Sign Up and Admin Login Module

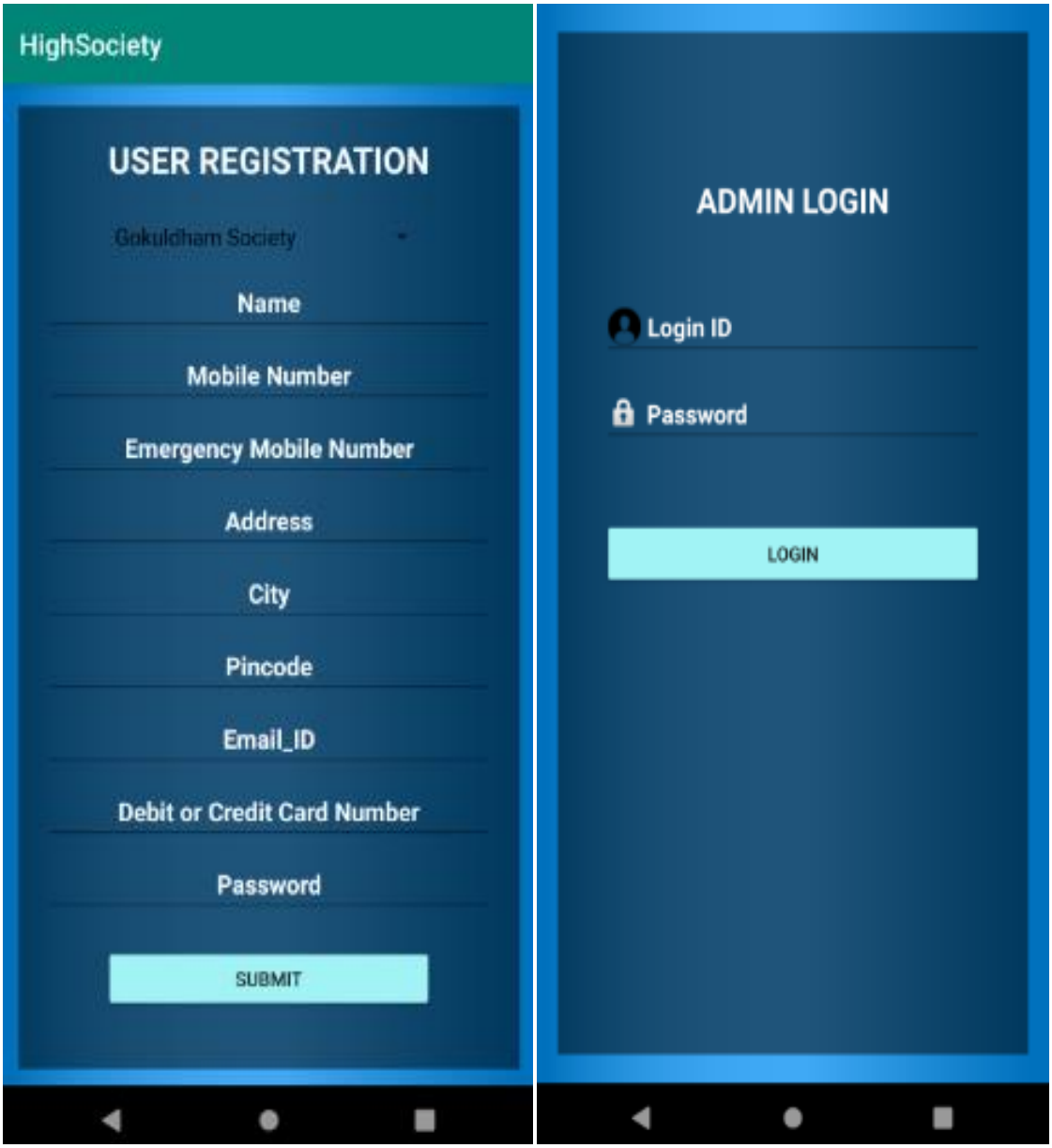


Figure5.5.1.1 Sign Up and Admin Login Module

5.5.1.2 Forgot Password

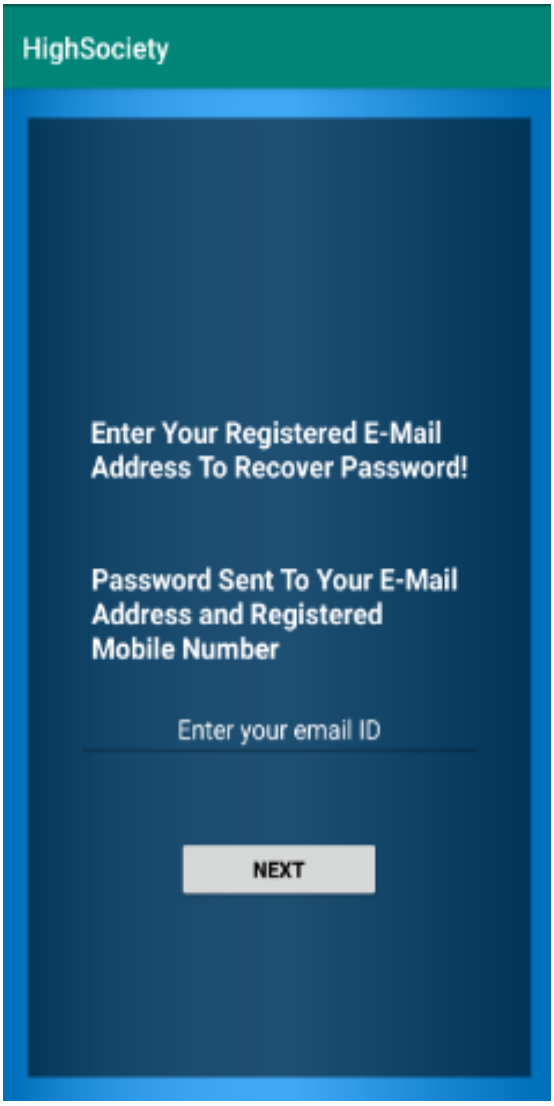


Figure5.5.1.2 Forgot Password

5.5.1.3 User Interface Module

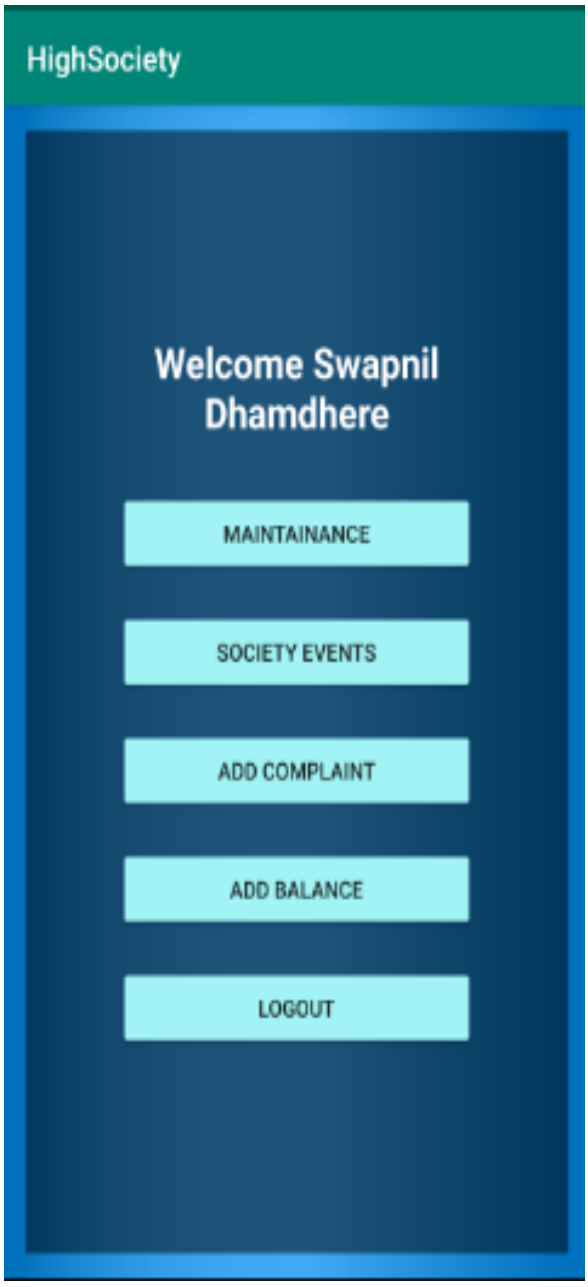


Figure5.5.1.3 User Interface Module

5.5.1.4 Pay Maintenance and Add Balance Module

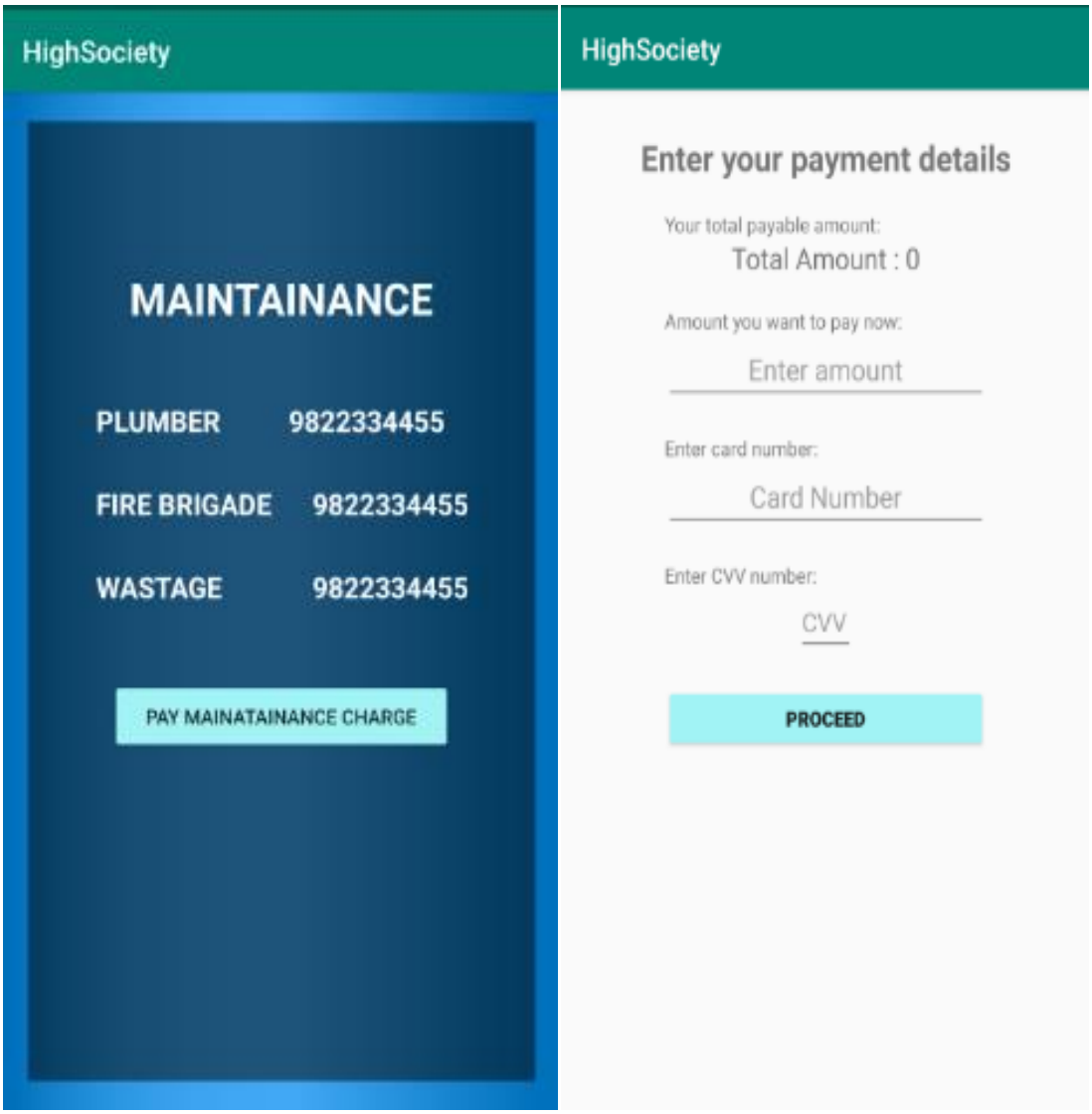


Figure5.5.1.4 Pay Maintenance and Add Balance Module

5.5.1.5 Add Complaint and View Events Module

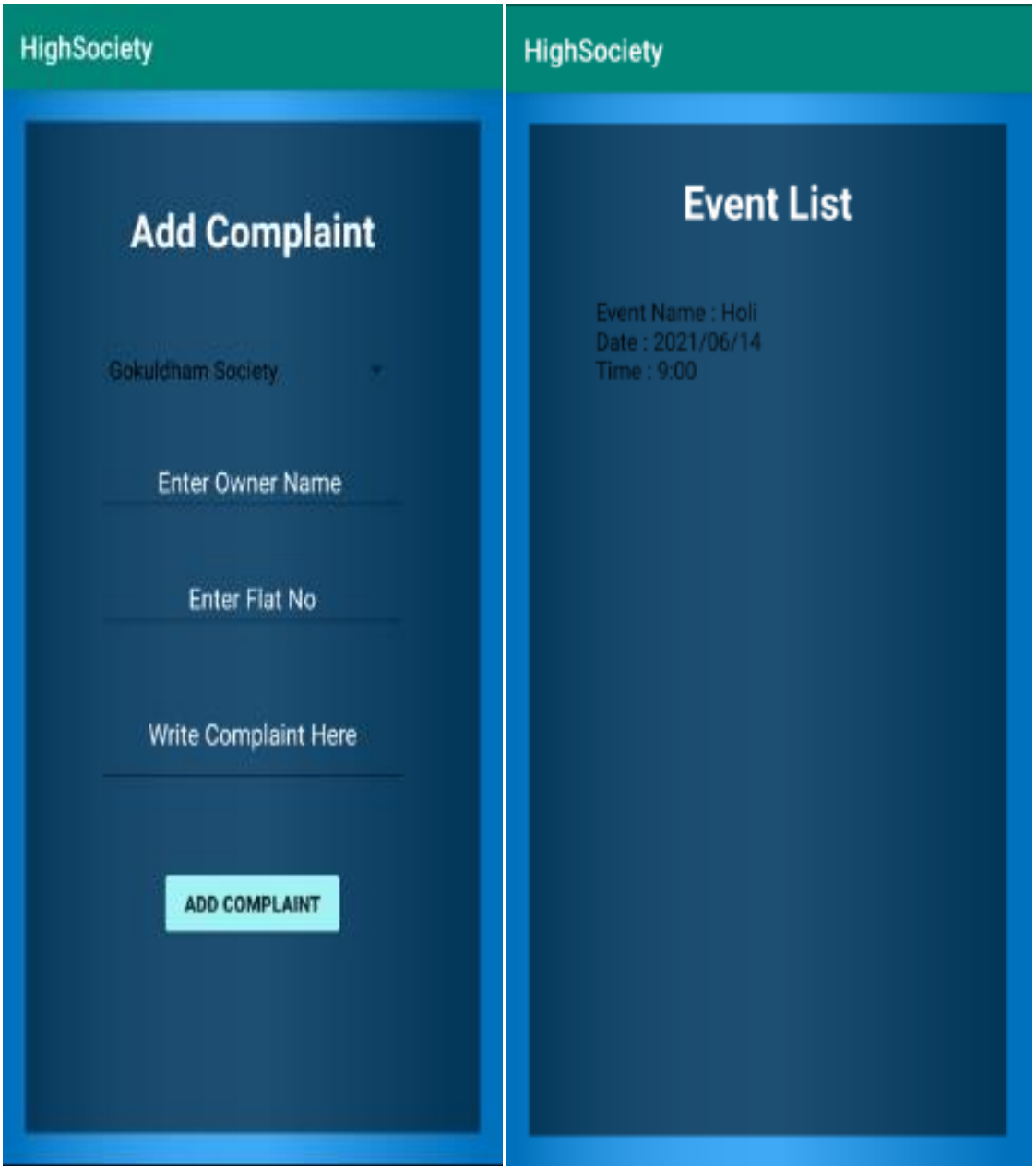


Figure5.5.1.5 Add Complaint and View Events Module

Chapter 6

TESTING

Testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing depending on the testing method employed can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the Software Development methodology adopted.

6.1 TESTING TOOL – Selenium IDE

The Selenium-IDE (Integrated Development Environment) is the tool you use to develop your

Selenium test cases. It‟s an easy-to-use Chrome and Firefox extension and is generally the most efficient way to develop test cases. It records the users actions in the browser for you, using existing Selenium commands, with parameters defined by the context of that element. This is not only a timesaver, but also an excellent way of learning Selenium script syntax.

6.2 TEST PLAN

A **test plan** documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from Test Engineers.

Test plan document formats can be as varied as the products and organizations to which they apply. There are three major elements that should be described in the test plan: **Test Coverage, Test Methods, and Test Responsibilities.** These are also used in a formal test strategy.

Test coverage in the test plan states what requirements will be verified during what stages of the product life.

Test methods in the test plan state how test coverage will be implemented. Test methods also specify test equipment to be used in the performance of the tests and establish pass/fail criteria.

Test responsibilities include what organizations will perform the test methods and at each stage of the product life. Test responsibilities also includes, what data will be collected, and how that data will be stored and reported (often referred to as "deliverables").

Table 6.1 Test Plan

Name of Tester	Name Test Item / Module / Function	Date of Testing
Swapnil Dhamdhere	User Registration Module and Admin Login	**19/06/2021**
Siddharth Desai	Sign In and Change password module with Gmail or Mobile No	**20/06/2021**
Smita Bansode	Pay Maintenance And Add Balance Module	**22/06/2021**
Prasad Bhavsar	Add Complaint And View Events Module	**23/06/2021**

6.3 TEST CASES

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. It may take many test cases to determine that a software program or system is functioning correctly. Test cases are often referred to as test scripts, particularly when written. Written test cases are usually collected into test suites.

A test case is a detailed procedure that fully tests a feature or an aspect of a feature. Whereas the test plan describes what to test, a test case describes how to perform a particular test. You need to develop a test case for each test listed in the test plan. A test case includes:

- The purpose of the test.
- Special hardware requirements, such as a modem.

- Special software requirements, such as a tool.
- Specific setup or configuration requirements.
- A description of how to perform the test.
- The expected results or success criteria for the test.

Test case priority:**High**	Test Designed date:**18/06/2021**
Module Name:**User Registration**	Test Executed by:**Swapnil Dhamdhere**
Test Title:**Society Management**	Test Execution date:**19/06/2021**
Pre-Condition:**Android OS & SMS.apk**	

Test Cases	Test Case Objectives	Steps	Expected Result	Actual Result	Status
TC-1	Open the Society Management System app.	Click on logo	App will be open	App is opened	Pass
TC-2	Check if the User Registration screen UI is Correct	Open the Registration (Sign Up) screen	Registration screen will be opened and correct	Registration screen is opened and correct	Pass
TC-3	Enter Invalid Name, Email ID,Mobile No, Password, Debit Card No & click on Submit button.	1.Enter Invalid Name 2.Enter Invalid Email ID 3.Enter Invalid Mobile No 4.Click on Submit Button	User can't be able to register his/her data	User would not be able process further.	Fail
TC-4	Enter Valid Name, Email ID,Mobile No, Password, Debit Card No & click on Submit button.	1.Enter Valid Name 2.Enter valid Email ID 3.Enter Valid Mobile No 4.Click on Submit Button	User can be able to register his/her data.	User would be able process further.	Pass

Table 6.2 User Registration test-cases

Test case priority:**High**	Test Designed date:**18/06/2021**
Module Name:**Admin Login**	Test Executed by:**Swapnil Dhamdhere**
Test Title:**Society Management**	Test Execution date:**19/06/2021**
Pre-Condition:**Android OS & SMS.apk**	

Test Cases	Test Case Objectives	Steps	Expected Result	Actual Result	Status
TC-1	Open the Society Management System app.	Click on logo	App will be open	App is opened	Pass
TC-2	Check if the Admin Login screen UI is Correct	Open the Admin Login screen by clicking on button	Admin login screen will be opened and correct	Admin login screen is opened and correct	Pass
TC-3	Enter Invalid ID and Password	1.Enter Invalid ID 2.Enter Invalid Password 3.Click on Login Button	User can't be able to login in Admin module	User would not be able process further.	Fail
TC-4	Enter valid ID and Password	1.Enter valid ID 2.Enter valid Password 3.Click on Login Button	User will be able to login in Admin module	User would be able process further.	Pass

Table 6.3 Admin Login test-cases

Test case priority: **High**	Test Designed date: **19/06/2021**
Module Name: **Sign In & change Password**	Test Executed by: **Siddharth Desai**
Test Title: **Society Management**	Test Execution date: **20/06/2021**
Pre-Condition: **Android OS & SMS.apk**	

Test Cases	Test Case Objectives	Steps	Expected Result	Actual Result	Status
TC-1	Open the Society Management System app.	Click on logo	App will be open	App is opened	Pass
TC-2	Enter Invalid Mobile No and Password & Click on Sign In button.	1..Enter Invalid Mobile No. 2.Enter Invalid Password 3.Click on Sign In Button	User ca't be able to Sign In his/her data	User would not able to Sign In his/her data	Fail
TC-2	Enter Valid Mobile No and Password & Click on Sign In button.	1..Enter Valid Mobile No. 2.Enter valid Password 3.Click on Sign In Button	User can be able to Sign In his/her data	User would able process further	Pass
TC-3	Enter a Invalid password (Forget Password)	User will get to know password with help of Email-Id.	User can be able to get his/her Password through sms or email.	User is able to see Password	Pass

Table 6.4 Sign In and Change Password test-cases

Test case priority: **High**	Test Designed date:**21/06/2021**
Module Name:**Pay Maintenance & Add Balance**	Test Executed by:**Smita Bansode**
Test Title:**Society Management**	Test Execution date:**22/06/2021**
Pre-Condition:**Android OS & SMS.apk**	

Test Cases	Test Case Objectives	Steps	Expected Result	Actual Result	Status
TC-1	User should be able to Pay Maintenance.	1.Sign In and click on Maintenance button. 2.See all the details. 3.Click on Pay Maintenance charge button	Maintenance Paid Successfully message shown.	Paid message shown successfully.	Pass
TC-2	User Should be able to Add Balance to wallet Successfully	1.Sign In and click on Add Balance Button. 2.Enter Invalid Amount, Card No and CVV. 3.Click on Submit button.	Further Process should not been shown, Shows error.	Error Message shown, Not able to proceed further.	Fail
TC-3	User Should be able to Add Balance to wallet Successfully	1.Sign In and click on Add Balance Button. 2.Enter Valid Amount, Card No and CVV. 3.Click on Submit button	User will be able Proceed Further.	Proceed Further And Amount Added Successfully.	Pass

Table 6.5 Pay Maintenance & Add Balance Module test-cases

Test case priority: **Medium**		Test Designed date: **22/06/2021**			
Module Name: **Add Complaint & View Events Module**		Test Executed by: **Prasad Bhavsar**			
Test Title: **Society Management**		Test Execution date: **22/06/2021**			
Pre-Condition: **Android OS & SMS.apk**					
Test Cases	**Test Case Objectives**	**Steps**	**Expected Result**	**Actual Result**	**Status**
TC-1	To Use Add Complaint Function in the App	1.Sign In and go to Add Complaint Button And click on it. 2.Enter essential details and Write your complaint. 3.Click on Add Complaint Button.	User will able be to Add Complaint.	User Complaint added successfully and will be visible to Admin.	Pass
TC-2	User should able to View Events	1.Sign In into system 2.Click on Society Events.	User will able to see upcoming Society Events.	User is able to see Events.	Pass

Table 6.6 Add Complaint & View Events Module test-cases

Chapter 7

STRENGTH & LIMITATIONS

STRENGTHS

1. Easy to understand by the user.
2. Easy to operate.
3. Good User Interface..
4. Satisfies the User Requirements.
5. Robust database back-end.
6. Better storage capacity.
7. Admin have Creativity Control
8. Time Saving.

LIMITATIONS

1. Before running the application, make sure that the server application is running. If server is not in the running condition then you can can't be able to use the App.
2. Does not support chat conversations.
3. Users cannot interact with other users through this app.
4. Admin login should be handled by only Society Secretary.

Chapter 8
FUTURE SCOPE

The functions of a society management system are an inevitable part of our lives. There are chores here which unknowingly take up a considerable amount time of our lives. Society secretory often gets tired of maintaining multiple email groups; excel sheets containing members' contact information, taking maintenance checks from owners and at the same time addressing the grievances of the housing society. Worse is the case where technology remains unused. Also, residents get restless that issues are not getting resolved despite reminders and no one knows the status of the complaint raised. Such issues and many more are common in most housing societies. With no appropriate tools, managing a residential complex takes too much of time, effort and money with a lot of inefficiency. As times have changed, most of us have strived to combine technology with our daily chores irrespective of the field. Thus, changing the way of maintaining the society information will also prove to be beneficial, improve efficiency and save us time

- Adding **Chat** feature so members can interact with **each other personally**.
- Adding **Group Chat** feature so that **Admin** can **Announce Notices** directly to everyone at one time.
- Adding **Payment Receipt** feature.
- Adding more about Society Details with location link and with photos/videos.

Chapter 9

CONCLUSION & REFERENCES

CONCLUSION

Although this project is made for small scale of society management but if we implement new technologies & features in the system, it will be able to fullfill all the requirements of users or say members in the society, Like In this project user can communicate with other users using this system. This project hopes to develop a Society Management System Android app with high Quality user interface and have all features for the society members and its nodes.

REFERENCES

[1] ***Lauren Darcey , "Android App Development"***, *Second Edition, August 2011 dated: 13-12-2018*

[2] ***E.Balagurusamy, " Programming with Java"***, *Seventh Edition,2011*

[3] ***"Android Studio Download",*** *https://developer.android.com/studio, dated:14-12-2018*

[4] ***http://en.wikipedia.org*** */wiki/MySQL*

[5] ***"Git Hub Tutorial"***https://www.tutorialspoint.com/git/

APPENDIX

USER MANUAL

I. Required Hardware

1. Android Phone

II. Required Network

1. Internet Availability (1-2 Mbps)
2. Connected to Server.

Project Execution Steps

1. Get the APK file in the phone and Install the Application.
2. As the installation is complete the Home Page is available.
3. Then First Click on Admin Login button and fill the information required below and Register the Society.
4. Then Click on Sign In button and register user first with all necessary information then you can login to the system using your mobile no and password.
5. If you have already Registered then Click on Sign In button and you will be able to use all the features of this application.