**A**

**Project Report**

**On**

**"** **SOCIETY MANAGEMENT SYSTEM"**

(CE352 – Software Group Project -III)



**Prepared by**

PRATIK DHORIYANI (18CE024)

DHARMESH GORANIYA (18CE026)

HARSH HIRPARA (18CE030)

TUFEL JIRAWALA (18CE033)

ADIL KESHWANI (18CE042)

**Under the Supervision of**

**Prof. Khushboo Patel**

**Submitted to**

Charotar University of Science & Technology (CHARUSAT)

for the Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology (B.Tech.)

in Computer Engineering (CE)

for 5th semester B.Tech.

**Submitted at**

****

**Accredited with Grade A by NAAC**

**Accredited with Grade A by KCG**

****

**U & P U. PATEL DEPARTMENT OF COMPUTER ENGINEERING**

**(NBA Accredited)**

**Chandubhai S. Patel Institute of Technology (CSPIT)**

**Faculty of Technology & Engineering (FTE), CHARUSAT**

**At: Changa, Dist: Anand, Pin: 388421.**

**November, 2020**

**DECLARATION BY THE CANDIDATE/CANDIDATES**

We hereby declare that the project report entitled “Society Management System” submitted by us to Chandubhai S. Patel Institute of Technology, Changa in partial fulfilment of the requirement for the award of the degree of **B.Tech** in Computer Engineering, from U & P U. Patel Department of Computer Engineering, CSPIT/FTE, is a record of bonafide CE352 Software Group Project -III (project work) carried out by us under the guidance of **Prof. Khushboo Patel** We further declare that the work carried out and documented in this project report has not been submitted anywhere else either in part or in full and it is the original work, for the award of any other degree or diploma in this institute or any other institute or university.

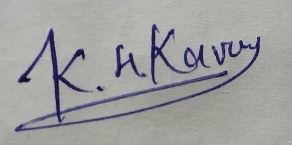
PRATIK DHORIYANI (18CE024)

DHARMESH GORANIYA (18CE026)

HARSH HIRPARA (18CE030)

TUFEL JIRAWALA (18CE033)

ADIL KESHWANI (18CE042)



Asst. Prof. Khushboo Patel

Assistant Professor

U & P U. Patel Department of Computer Engineering,

CSPIT/FTE, CHARUSAT-Changa.

****

**Accredited with Grade A by NAAC**

**Accredited with Grade A by KCG**

**CERTIFICATE**

This is to certify that the report entitled “Society Management System” is a bonafied work carried out by Pratik Dhoriyani (18ce024), Dharmesh Goraniya (18ce026), Harsh Hirpara (18ce030), Tufel Jirawala (18ce033) And Adil Keshwani (18ce042) under the guidance and supervision of **Prof. Khushboo Patel** for the subject **Software Group Project -III (CE352)** of 5th Semester of Bachelor of Technology in **Computer Engineering** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate themselves, has duly been completed, and fulfills the Partial requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

|  |  |
| --- | --- |
| Under the supervision of,    **Prof. Khushboo Patel**  U & P U. Patel Dept. of Computer Engineering.  CSPIT/FTE, CHARUSAT, Changa, Gujarat |  |
| Dr. Ritesh Patel  Head - U & P U. Patel Department of Computer Engineering,  CHARUSAT, Changa, Gujarat. | |

**Chandubhai S. Patel Institute of Technology (CSPIT)**

**Faculty of Technology & Engineering (FTE), CHARUSAT**

At: Changa, Ta. Petlad, Dist. Anand, Pin:388421. Gujarat.

# ABSTRACT

This system is developed to manage day-to-day activities of any co-operative housing society. Generally, in society all the work is manually. Contact no of members are noted on the papers. As all work is done on paper so it is very difficult to manage and keep track of all the work expenses in the society. This co-operative society management system will computerize all day to day operation in the society. This System also keep the details of service providers who provide different services. Members can download an invoice of maintenance of their flat. This system is of maintaining a society is made in such a way, so that the most common problem faced in residential societies are solved.

**ACKNOWLEDGEMENT**

We are privileged to have this opportunity to express our gratitude and acknowledge everyone’s never ending support and valuable contributions for our project.

We would like to express my sincere gratitude to our advisor Asst. Prof. Khushboo Patel for the continuous support of our project study and related research, for her patience, motivation, and immense knowledge.

Our sincere thanks also goes to Principal Sir Dr. A.D Patel, Head Dr. Ritesh Patel who provided us an opportunity to work on a project and to be able to present the same.

Last but not the least, We would like to thank our friends and families for supporting us spiritually throughout this project and for always being a constant source of inspiration.

We also place on record, our sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

# TABLE OF CONTENTS

[Abstract v](#_TOC_250053)

[Acknowledgement vi](#_TOC_250052)

List of figures ….ix

List of tables x

[Chapter 1 Introduction 1](#_TOC_250051)

* 1. [Project Summary 1](#_TOC_250050)
  2. Purpose ...1
  3. [Objective 1](#_TOC_250049)
  4. [Scope 1](#_TOC_250048)

[Chapter 2 Project Management. 2](#_TOC_250046)

* 1. [Project Planning. 2](#_TOC_250045)
     1. [Project Development Approach and Justification. 2](#_TOC_250044)
     2. Project Effort and Time, Cost Estimation. 2
     3. [Roles and Responsibilities 5](#_TOC_250043)
  2. [Project Scheduling. 5](#_TOC_250042)
     1. [Gantt Chart Representation. 5](#_TOC_250041)
     2. [Task Network Representation. 6](#_TOC_250040)
     3. [Pert Chart Representation. 7](#_TOC_250039)

[Chapter 3 System Requirements Study 8](#_TOC_250038)

* 1. [User Characteristics 8](#_TOC_250037)
  2. [Hardware and Software Requirements 8](#_TOC_250036)

3.2.1  [Software Requirements. 8](#_TOC_250044)

3.2.2  [Hardware Requirements. 8](#_TOC_250044)

* 1. Assumptions and Dependencies 8

[Chapter 4 System Analysis 9](#_TOC_250035)

* 1. [Study of Current System 9](#_TOC_250034)
  2. [Problem and Weaknesses of Current System 9](#_TOC_250033)
  3. [Requirements of New System 9](#_TOC_250032)
     1. [Functional Requirements 9](#_TOC_250031)

4.3.1.1 [F1: Usability 9](#_TOC_250031)

4.3.1.2 [F2: Efficiency 9](#_TOC_250031)

4.3.1.3 [F3: Reliability 9](#_TOC_250031)

4.3.1.4 [F4: System attributes 10](#_TOC_250031)

4.3.1.5 [F5: Software requirements 10](#_TOC_250031)

4.3.1.6 [F6: Hardware requirements 10](#_TOC_250031)

* + 1. [Non Functional Requirements 10](#_TOC_250030)

4.3.1.1 [Usability 10](#_TOC_250031)

4.3.1.1 [Design 11](#_TOC_250031)

4.3.1.1 [Performance 11](#_TOC_250031)

* 1. [Use case Diagram 12](#_TOC_250029)
     1. [User Use-Case 12](#_TOC_250030)
     2. [Admin Use-Case 13](#_TOC_250030)
  2. [Class Diagram 14](#_TOC_250026)
  3. [Sequence Diagram 14](#_TOC_250025)
  4. [Context Diagram(level-0 DFD) 15](#_TOC_250024)
  5. [Data Flow Diagram(1 and 2 level) 16](#_TOC_250023)
  6. [Data Modeling 19](#_TOC_250022)
     1. [Data Dictionary 19](#_TOC_250031)
     2. ER Diagram 22

[Chapter 5 System Design 23](#_TOC_250020)

* 1. [System Application Design. 23](#_TOC_250019)
     1. [State Transition Diagram 23](#_TOC_250018)
  2. [Input/Output and Interface Design. 24](#_TOC_250017)
     1. [Samples of Forms, Reports and Interface 24](#_TOC_250016)

[Chapter 6 Testing. 26](#_TOC_250015)

* 1. [Testing Plan 26](#_TOC_250014)
  2. [Testing Strategy. 26](#_TOC_250012)
  3. [Test Suites Design 26](#_TOC_250011)
     1. [Test Cases 26](#_TOC_250006)

[Chapter 7 Conclusion and Discussion. 30](#_TOC_250010)

* 1. [Self Analysis of Project Viabilities. 30](#_TOC_250009)
  2. [Problem Encountered and Possible Solutions 30](#_TOC_250008)
  3. [Summary of Project work. 30](#_TOC_250007)

[Chapter 8 Limitation and Future Enhancement 31](#_TOC_250005)

* 1. [Limitation 13](#_TOC_250004)
  2. [Future Enhancement 31](#_TOC_250003)

LIST OF FIGURES

Fig 2.1 Gantt Chart Representation. .5

Fig 2.2 Network Representation. ..7

Fig 2.3 Pert Chart. ..7

Fig 4.1 User Use-Case 12

Fig 4.2 Admin Use-Case 13

Fig 4.3 Class Diagram 14

Fig 4.4 Sequence Diagram 14

Fig 4.5 Context Diagram(Level-0 DFD) ..15

Fig 4.6 DFD Level-1 (A) ..16

Fig 4.7 DFD Level-1 (B) ...17

Fig 4.8 DFD Level-2 (A) ..17

Fig 4.9 DFD Level-2 (B) ..18

Fig 4.10 ER Diagram. 22

Fig 5.1 State Transition Diagram. .23

Fig 5.2 Main Page. .24

Fig 5.3 Login Page 24

Fig 5.4 Signup Page 25

Fig 5.5 Selection Page 25

Fig 5.6 List of Properties for Buy 25

Fig 5.7 List of Properties for Rent .25

# LIST OF TABLES

Table 2.1 Roles and Responsibilities…………………….…………………………3

Table 2.2 Complexity Adjustment Value 4

Table 2.3 Role and Responsibilities 5

Table 4.1: Postproperty\_sell. 20

Table 4.2: Postproperty\_rent 21

Table 4.3 :Postrequirement\_buy 21

Table 4.4: Postrequirement\_sell. 22

Table 6.1 Black Box Testing(Login) 27

Table 6.2 Black Box Testing(Sell Own Property) 28

Table 6.3 Black Box Testing(Sell Rental Property). 29

Table 6.4 Black Box Testing(Buy Own Property) 29

**CHAPTER 1 INTRODUCTION**

## Introduction

## This system is developed to manage day-to-day activities of any co-operative housing society. Generally, in society all the work is manually. Contact no of members are noted on the papers. As all work is done on paper so it is very difficult to manage and keep track of all the work expenses in the society. This society co-operative society management system will computerize all day to day operation in the society. This System also keep the details of service providers who provide different services. Members can download an invoice of maintenance of their flat. This system of maintaining a society is made in such a way, so that the most common problem faced in residential societies are solved.

### Purpose

Society Management System is the android application designed to reduce conflicts among society members. The system has automated functionality for calculating monthly maintenance and member can view their bill status on their account. The main functionality of this project is that to overcome the method of old society management systems which was maintained manually and to handle the society digitally and smartly. The Society Management System allows members to login with their own account and get updated with society happenings.

### Scope

**CHAPTER 2 PROJECT MANAGEMENT**

**2.1 Project Planning**

**2.1.1 Project Development Approach and Justification**

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate. Product definition is stable. Requirement can change easily in any phase.

Front-end development: XML (Android) Back-end development: Java

* + 1. **Project Effort and Time, Cost Estimation**

Basic COCOMO Model **​**

It gives only an approximate estimation.

Effort = a1 \* (KLOC)^a2 ​

Tdev = b1 \* (Effort)^b2​

KLOC is the estimated kilo lines of source code,

a1, a2, b1, b2 are constants for different categories of software products, ​

Tdev is the estimated time to develop the software in months, ​

Effort estimation is obtained in terms of person months (PMs). ​

Development Effort & Time Estimation

Organic:

Effort = 2.4 (KLOC)^1.05 PM​

Tdev = 2.5 (Effort)^0.38 Months​

Semi-detached: ​

Effort = 3.0(KLOC)^1.12 PM​

Tdev = 2.5 (Effort)^0.35 Months ​

Embedded: ​

Effort = 3.6 (KLOC)^1.20PM​

Tdev = 2.5 (Effort)^0.32 Months​

Effort and Nominal Development time of our Project

Line of Code of our project = 3876 line of source code

Line of Code of our project = 3.876 KLOC

Effort = 2.4\*(3.876)^1.05 = 9.74 PM​ (Person Month)

Nominal development time = 2.5\*(9.74)^0.38

Suppose the average monthly salary of each software developer is Rs. 20,000.

Total Cost of the project is = 20000 \* 5 People \* 5.93 Months

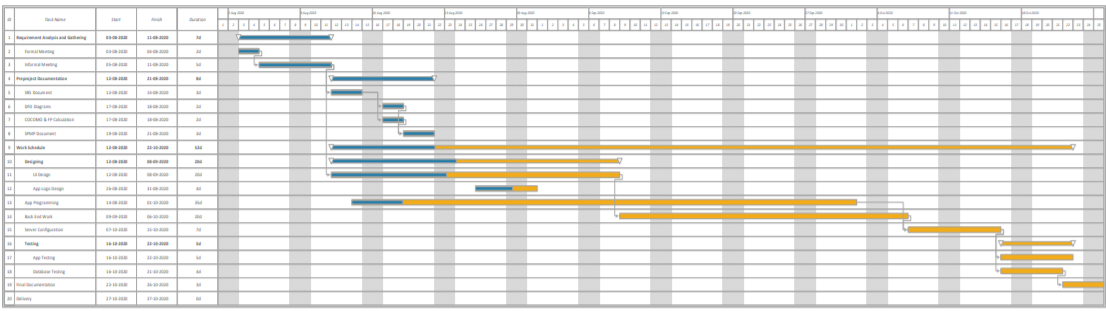
          =Rs 593,734

* + 1. **Roles and Responsibilities**

|  |  |  |
| --- | --- | --- |
| **Member Name** | **Roles** | **Responsibilities** |
| Harsh Hirpara | Project Leader,  UI Design | Lead the project development and also design UI. |
| Pratik Dhoriyani | Database Adminitrator, Programmer | Designing Database and Coding. |
| Tufel Jirawala | UI Designer | Design UI. |
| Dharmesh Goraniya | UI Designer,  Documentation | Design UI and also documentation. |
| Adil Keshwani | Tester | Testing |

2.1 Roles and Responsibilities

* 1. **Project Scheduling (Gantt Chart/PERT/Network Chart)**



2.2 Gantt Chart

**CHAPTER 3 SYSTEM REQUIREMENTS STUDY**

**3.1 User Characteristics**

• User can easily view his maintenance bill.

• User can complain regarding any problem.

• User can provide the funds for festival celebration.

**3.2 Hardware and Software Requirements**

**Hardware requirements**

Your development workstation should meet or exceed these hardware requirements:

* A 64-bit environment is required for Android 2.3.x (Gingerbread) and higher versions, including the master branch. You can compile older versions on 32-bit systems.
* At least 250GB of free disk space to check out the code and an extra 150 GB to build it. If you conduct multiple builds, you need additional space.

**Software requirements**

* Android Studio
  1. **Assumptions and Dependencies**
* The users have basic knowledge of applications.
* The information of all the users must be stored in a Firebase database that is accessible by the system.

**CHAPTER 4 SYSTEM ANALYSIS**

**4.0 System Analysis :-**

* System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. Analysis specifies what the system should do.

**4.1 Study of Current System :-**

* The Current system of society management is a traditional way of communication is used which include a common notice board system operated by responsible society member. The data is stored in the files and the processing of the data is done manually and the report generation is slow. Maintenance is generated on paper. It creates dependability with the specific person. Sometimes the person has to compromise with his own time schedule for these common activities. No one give attention on parking System.
  1. **Problem and Weaknesses of Current System**
* Guest Register.
* Parking module.
* Service Center.
* Event module.

**4.3 Requirements of New System**

* Only few amounts of data can be visible at a time
* Data like notice generate, maintenance, Event details, complaint box details are limited to see we increase size of visible details.
* Only limited to Android users
* We develop this application in android studio so it’s only limited to android user in future we add this app to iOS user also.
* Can’t make online call and meeting.
* We are not any online service to this app in future we add a different section for online call and meetings.
* Can’t make society Accounting.
* We also add a section for calculating Accounting for societies.

* Can’t display society information.
* When multiple society are connected then we also add section for display the information about societies.

**4.3.1 Functional Requirements: -**

A **Functional Requirement** (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements are also called **Functional Specification**.

These are the main functionality We add in our application: -

1. Notice Board
2. My flat
3. Maintenance
4. Parking
5. Event
6. Guest Register
7. Complain Box
8. Society Expenses
9. Flat Directory
10. Service Center
11. Rules
    * 1. **Non Functional Requirements :-**
12. **Security –**

The system uses SSL (secured socket layer) in all transactions that include any confidential customer information.

The system must automatically log out all customers after a period of inactivity.

The system should not leave any cookies on the customer’s computer containing the user’s password.

The system’s back-end servers shall only be accessible to authenticated administrators.

Sensitive data will be encrypted before being sent over insecure connections like the internet.

1. **Availability –**

The system should be available at all times, meaning the user can access it using a application restricted by the downtime of the server on which the system runs. In case of an of a hardware failure or database corruption, a replacement page will be shown. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24 X 7 availability.

1. **Maintainability –**

A commercial database is used for maintaining the database and the application server takes care of the application. In case of a failure, a re-initialization of the program will be done.

1. **Portability –**

An end-user is using this system only on android OS.

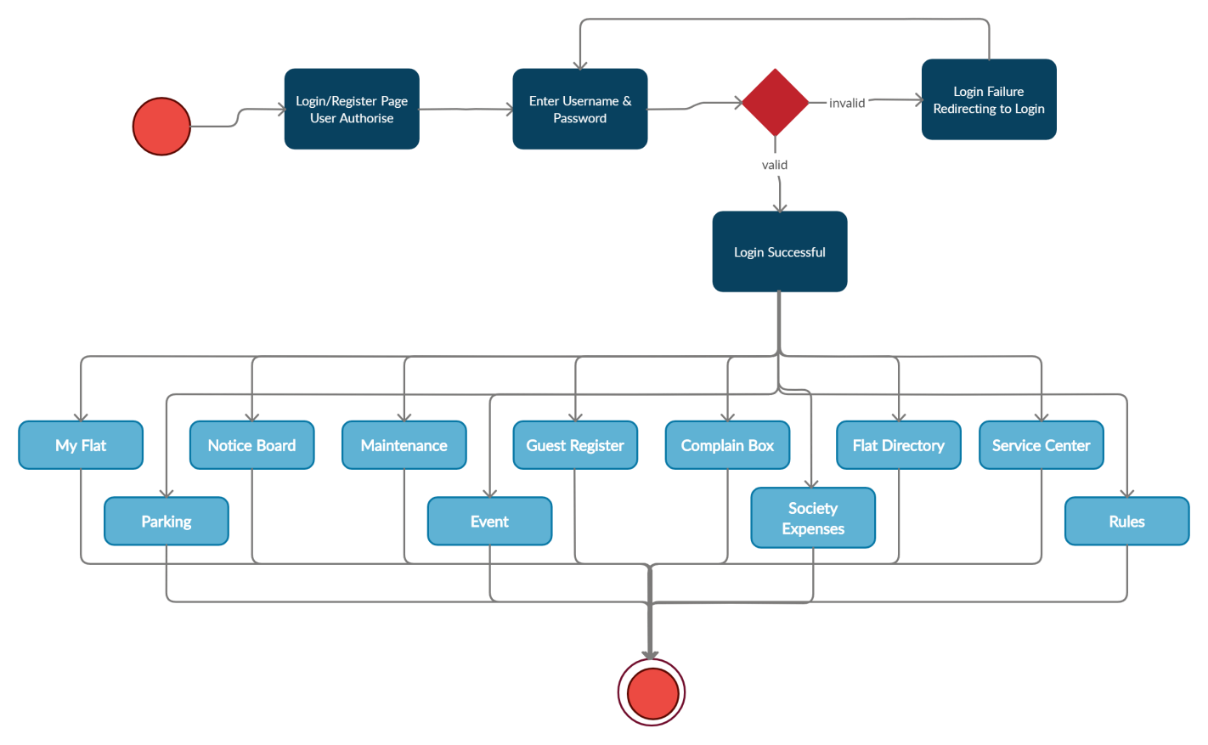
The system shall run on Mobile Phone,Laptops using emulators.

1. **Accessibility**

The system will be a mobile based application it is going to be accessible on the mobile.

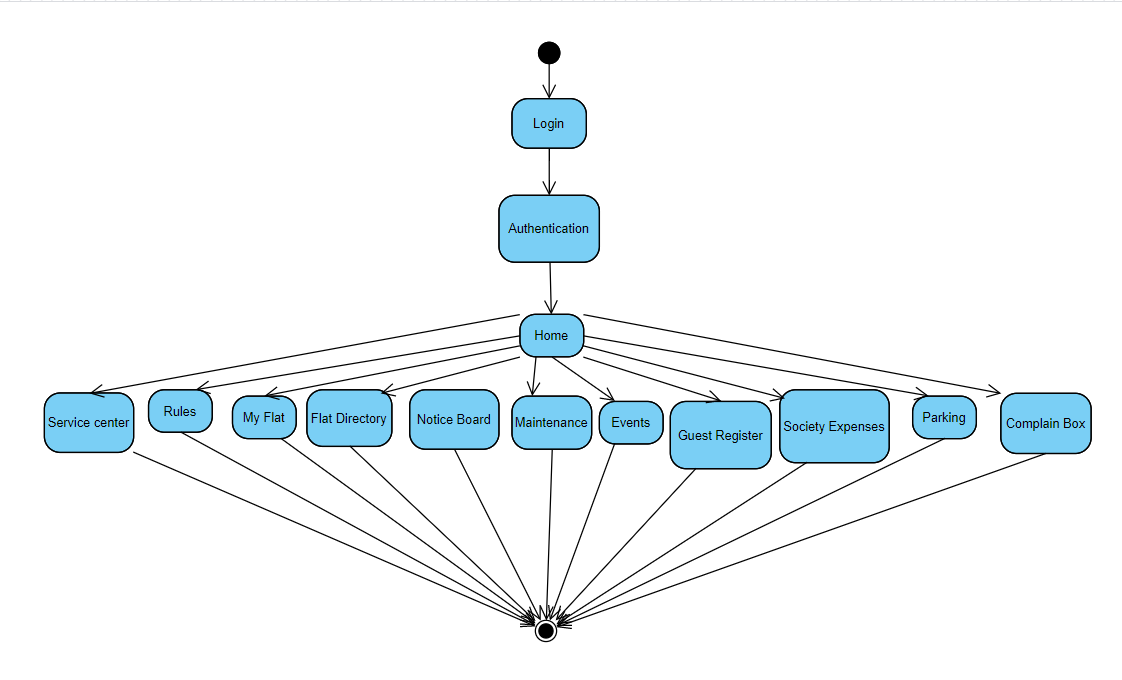
**For OO Approach**

* State diagram:-



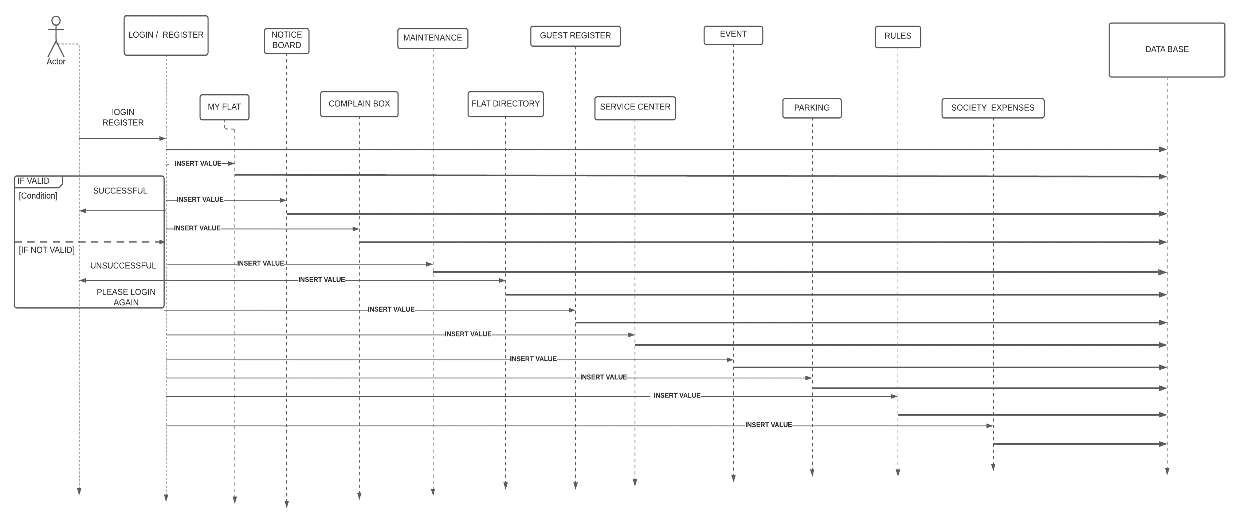
4.11

* Activity Diagram:-



4.12

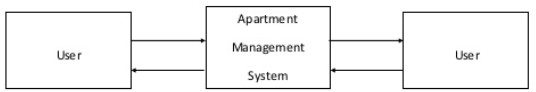
* Sequence Diagram :-



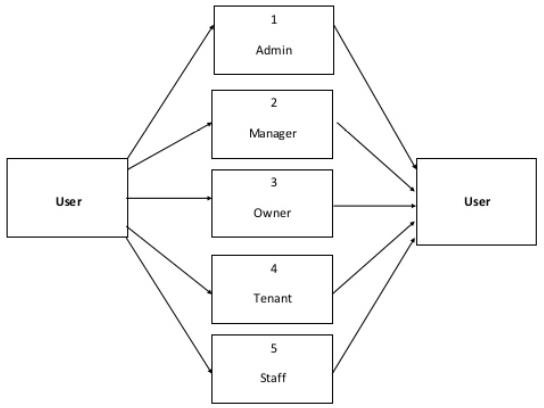
4.13

**For Conventional Approach**

* Context Diagram (level-0 DFD) :-

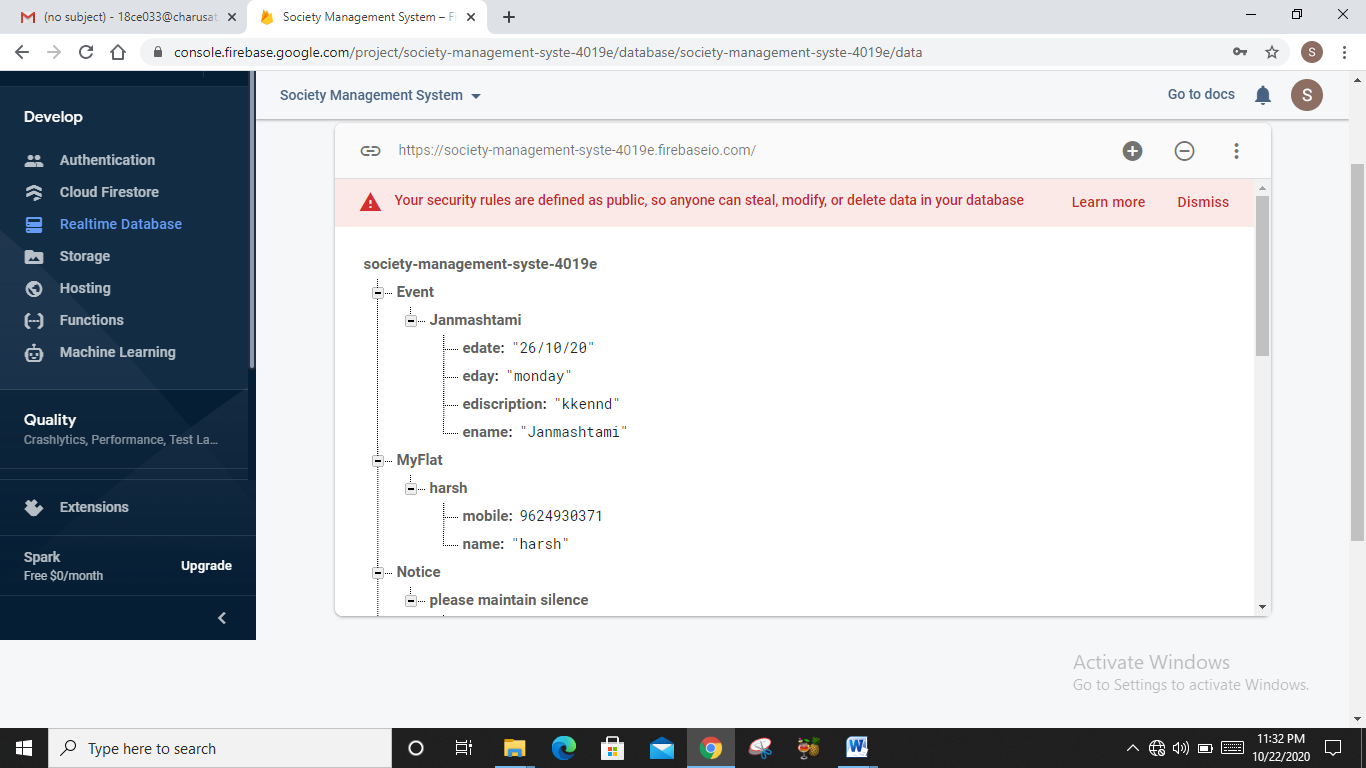


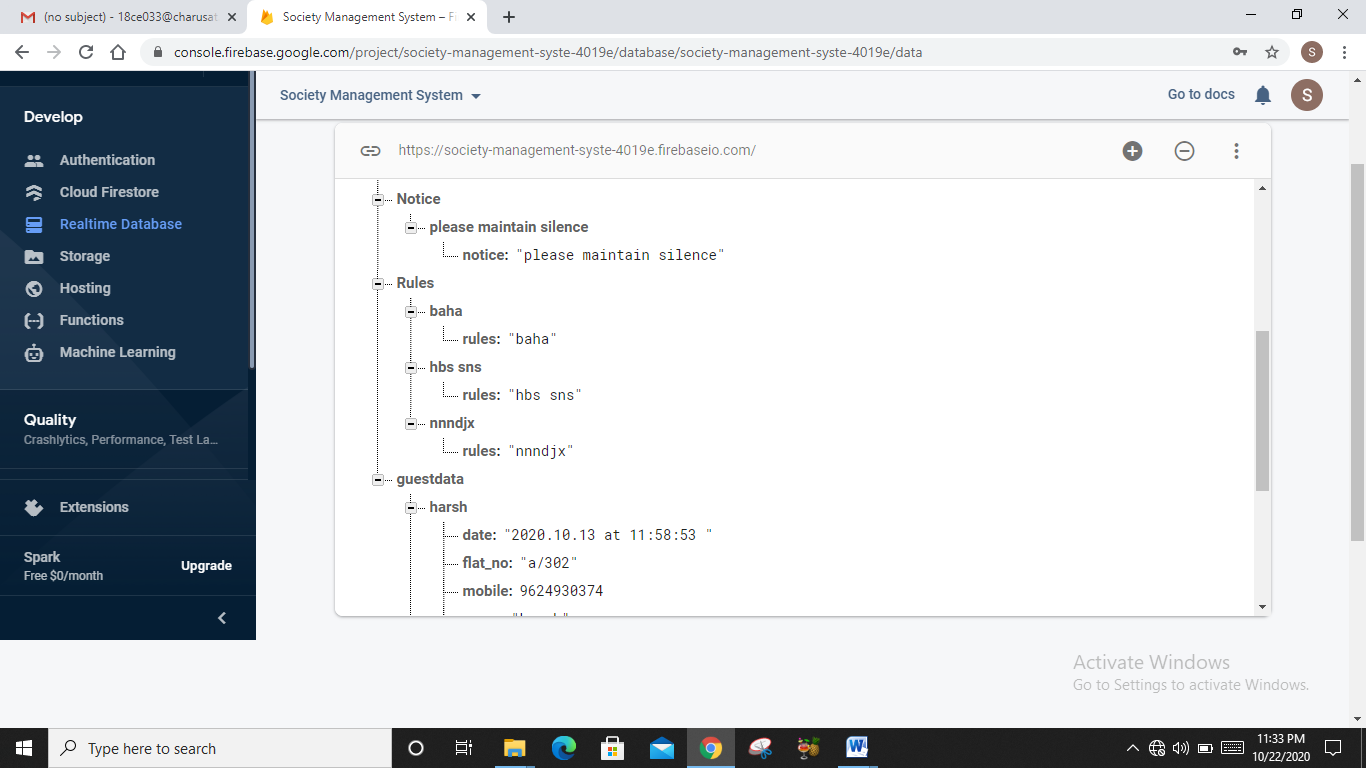
4.1

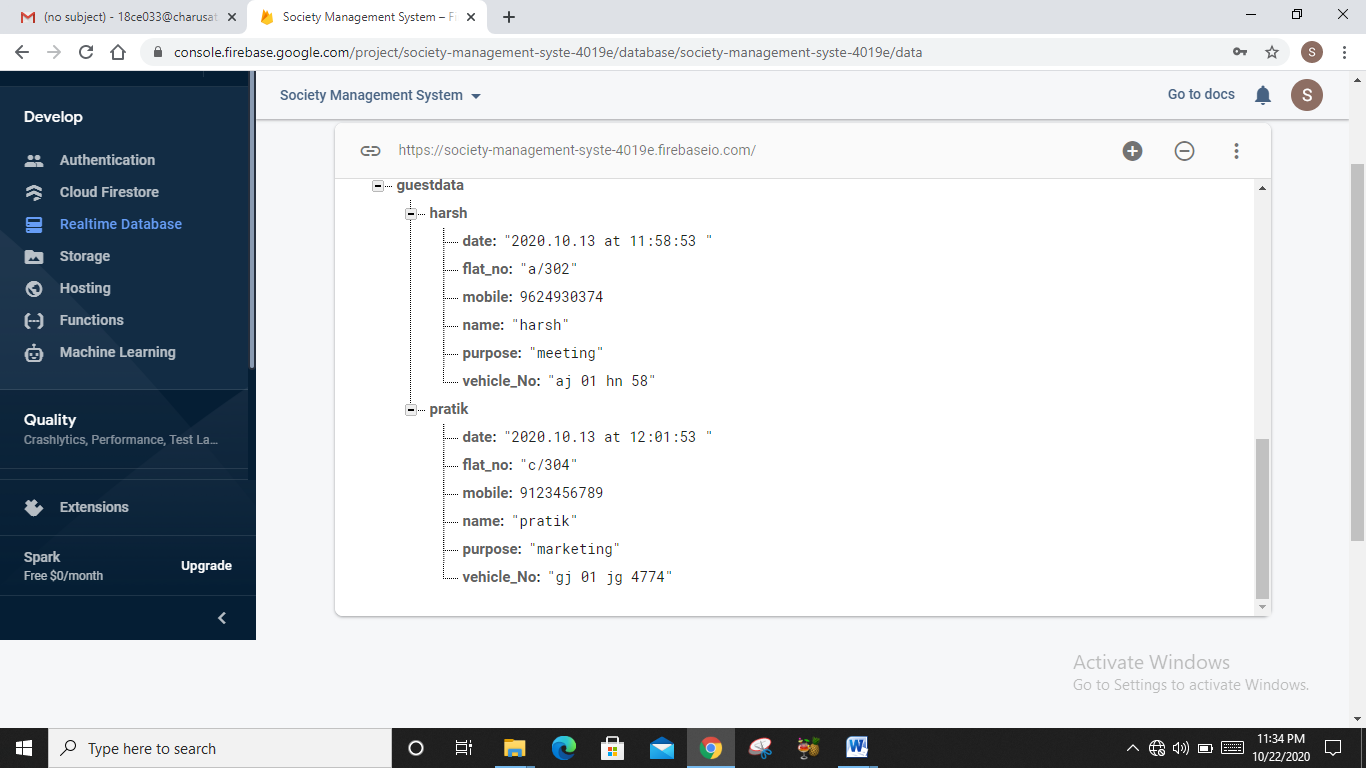
* Data Flow Diagram (1 and 2 level or higher) :-

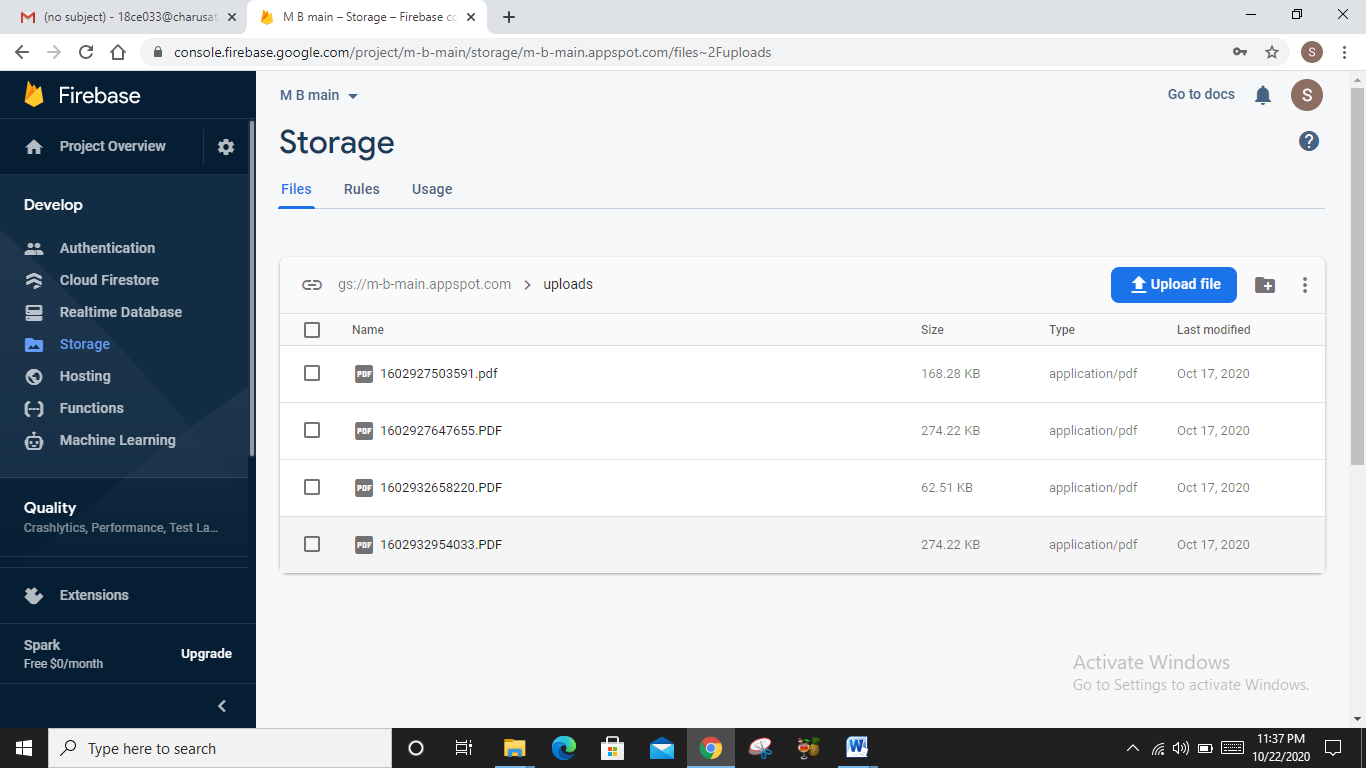
4.15

* Data Dictionary :-



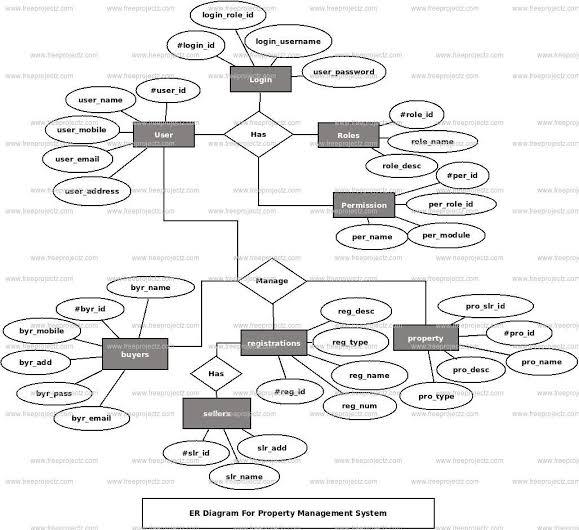






4.16

* ER Diagram



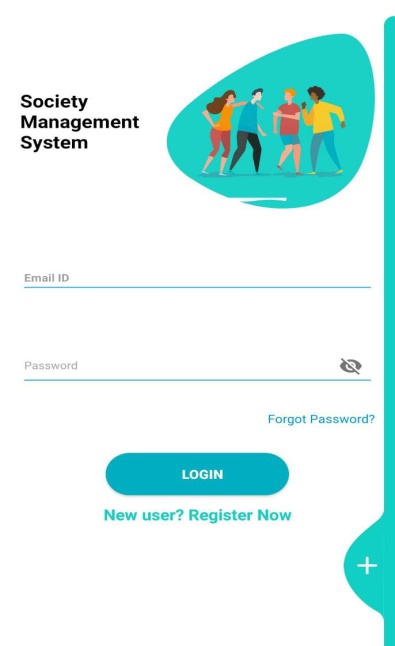
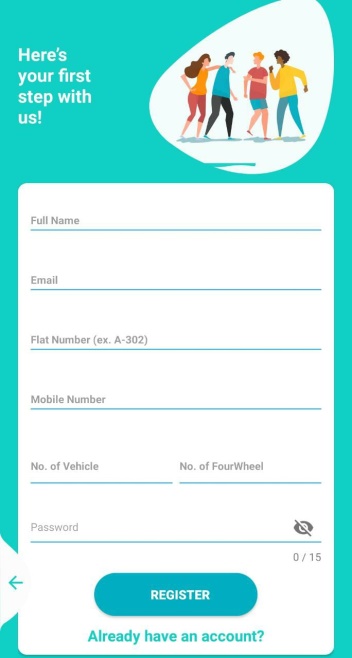
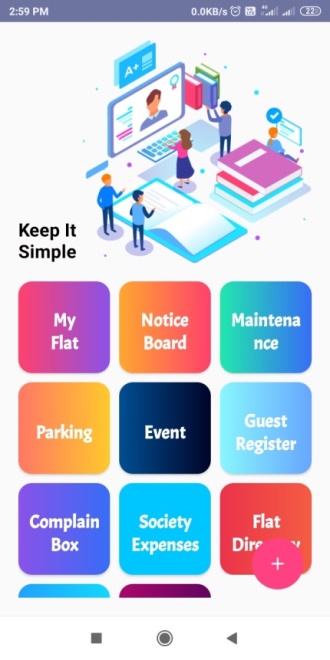
4.17

**CHAPTER 5 SYSTEM DESIGN**

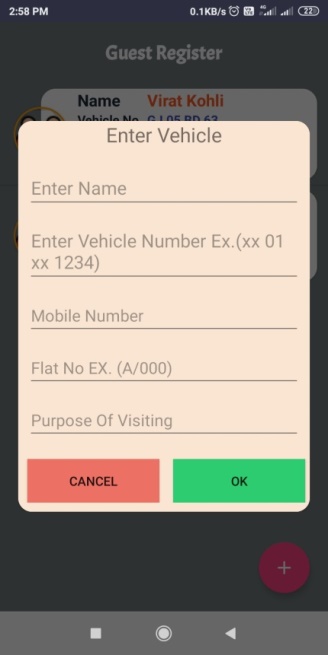
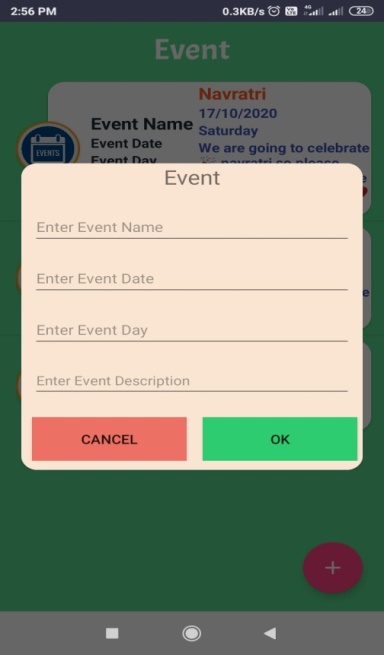
* 1. [**SYSTEM**](#_TOC_250019) **APPLICATION DESIGN**

[**INPUT/OUTPUT**](#_TOC_250017) **AND INTERFACE DESIGN**

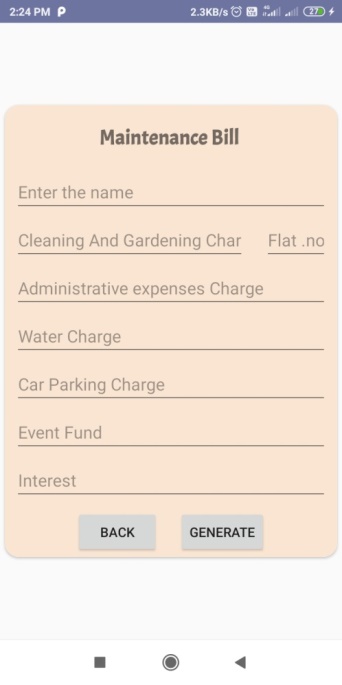
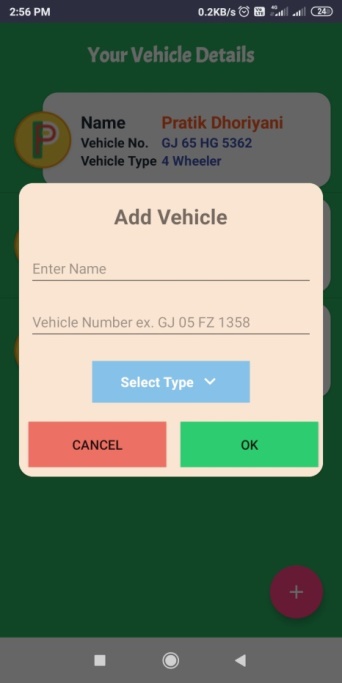
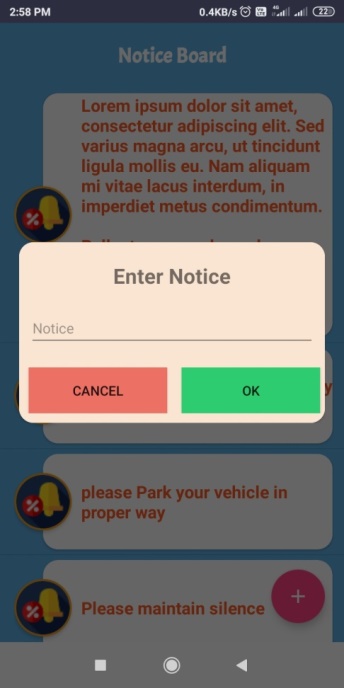
* + 1. [**SAMPLES**](#_TOC_250016) **OF FORMS, REPORTS AND INTERFACE**

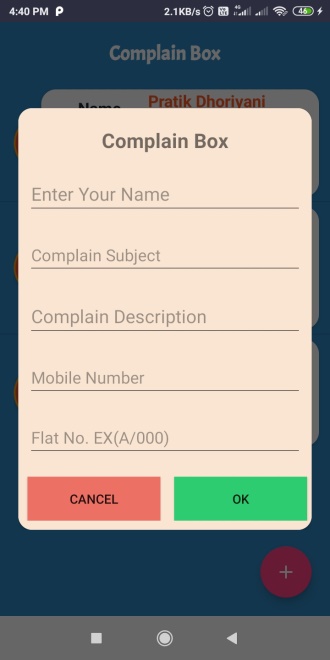
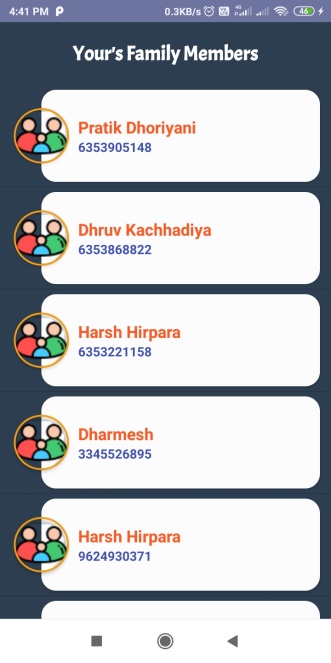
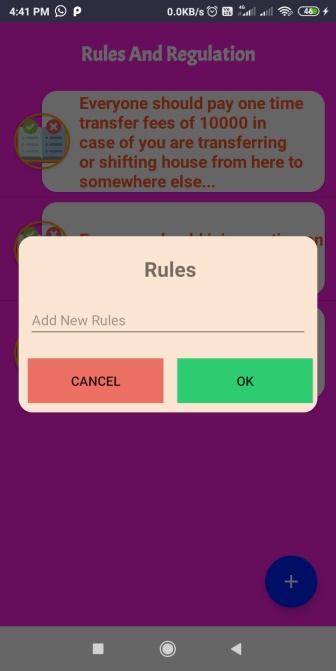
5.1 5.2 5.3

5.4 5.5 5.6

5.7 5.8 5.9

5.10 5.11 5.12

**CHAPTER 6 TESTING**

* 1. [**TESTING**](#_TOC_250014) **PLAN**

For project, I have used unit testing plan. There are different tasks we have to consider during testing with this type plan are as follows:

**Unit Testing Tasks and Steps:**

Step 1: Create a Test Plan.

Step 2: Create Test Cases and Test Data.

Step 3: If applicable create scripts to run test cases.

Step 4: Once the code is ready execute the test cases.

Step 5: Fix the bugs if any and re test the code.

Step 6: Repeat the test cycle until the “unit” is free of all bugs.

* 1. [**TESTING**](#_TOC_250012) **STRATEGY**

For testing this android application, we have tested each and every module by comparing its expected result with actual result. Each and every module has been tested.

* 1. **TESTING SUITES DESIGN**

**6.3.1 Test Cases**

Software Testing is evaluation of the software against requirements gathered from users and system specifications. It is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.

**Black-box** testing alludes to tests that are conducted at the software interface. They are used to determine that software functional are operational, that input is properly accepted and output is correctly produced, and that the integrity of external information (e.g. database) is maintained. It does not consider the internal logic structure that importance.

**White-box** testing of software is predicated on close examination of procedural detail. Logical paths through the software are tested by providing test cases that exercise specific sets of conditions and/or loops. The status of the program may be examined at various points to determine if the expected or asserted status corresponds to the actual status.

# Black Box Testing

Project Name: Society Management System Module Name: Login

Date of Creation: 22/10/2020

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case**  **No** | **Test Scenario** | **Pre- Condition** | **Test Steps** | **Test Case** | **Test Data** | **Expected Result** | **Actual Result** | **Post Condition** | **Status** |
| 1 | Verify Login | Account needs to be registered first | Enter user name, Enter password, Click on login button | Enter valid username and valid password | <Valid  Username> | Successful Login | Successful Login | Home Page is shown | Pass |
| <Valid  Password> |
| Enter valid username and invalid password | <Valid Username> | Invalid username or password | Invalid username or password | Login page is shown | Pass |
| <Invalid  Password> |
| Enter invalid username and valid  password | <Invalid  Username> | Invalid username or password | Invalid username or password | Login page is shown | Pass |
| <Valid Password> |
| Enter invalid username and invalid  password | <Invalid  Username> | Invalid username or password | Invalid username or password | Login page is shown | Pass |
| <Invalid Password> |  |

Project Name: Society Management System Module Name: Create Details

Date of Creation: 22/10/2020

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case No** | **Test Scenario** | **Pre- Condition** | **Test Steps** | **Test Case** | **Test Data** | **Expected Result** | **Actual Result** | **Post Condition** | **Status** |
|  |  |  | When |  |  | App is  displayed where the user can create Ad. | App is  displayed where the user can create Ad. |  |  |
|  |  |  | clicked |  |  |  |  |
|  |  |  | on |  |  |  |  |
|  |  |  | “Create |  |  | Fill up the |  |
|  |  |  | Detail”, it should be  redirected | - | - | details and create Ad. | Pass |
|  |  |  | to Ads |  |  |  |  |
|  |  |  | webpage. |  |  |  |  |
|  |  |  | Enter mobile number | Enter Positive 10digit  Number | <Valid Number> | Value is accepted. | Value is accepted. | Fill up other details. | Pass |
| Enter Negative Number /  Number | <Invalid Number> | Invalid Number Alert message displayed | Invalid Number | Re-enter Number | Pass |
|  |  |  |  | less than |  |  |  |
| 1 | Create Ad | User must be logged in with his valid credentials. |  | 10 digits |  |  |  |
| Select Sell / Rent Option | Select one  of the choice | <Valid Choice> | Value is accepted. | Value is accepted. | Data  Stored in Database. | Pass |
| Does not select any of  the given  choice | <No Input> | Alert message displayed | Please select one of the choices. | Select Choice to Sell / Rent | Pass |
|  |  |  | Enter | Enter the  name of the product | <Valid Input Name> | Value Accepted | Value Accepted | Fill Other Details | Pass |
|  |  |  | Product |  |  | Alert Message Displayed | Alert Message Displayed | Enter |  |
|  |  |  | Name | No Name  given | Empty  Field | name of  the | Pass |
|  |  |  |  |  |  | product |  |
|  |  |  |  | Upload Image Size  < 1 MB | Image Upload | Successful Upload | Successful Upload | Fill other details | Pass |
|  |  |  | Upload Photo |  |  |  |  |  |
| Upload Image Size  > 1 MB | Image Upload | Error | Error Alert Message  Displayed | Re- Upload photo size  < 1MB | Pass |

Project Name: Society Management System

Module Name: Search Items

Date of Creation: 22/10/2020

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case**  **No** | **Test Scenario** | **Pre- Condition** | **Test Steps** | **Test Case** | **Test Data** | **Expected Result** | **Actual Result** | **Post Condition** | **Status** |
| 1 | Search Items | All data should be stored in database | Search for items through the search bar | Enter input to search for an item | <Search Input> | Data is shown if available | Data is shown if available | - | Pass |
| Enter input to search for an item | <Search Input> | Alert Message in case no data fetched | No data fetched from database, alert  message | - | Pass |
|  |  |  |  |  |  | displayed |  |  |

Project Name: Society Management System

Module Name: Get Owner Contact Details

Date of Creation: 22/10/2020

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test**  **Case No** | **Test Scenario** | **Pre- Condition** | **Test Steps** | **Test Case** | **Test Data** | **Expected Result** | **Actual Result** | **Post Condition** | **Status** |
| 1 | Click on product details and get owner details | Data must be stored in database | Owner Details must be displayed on click of product details | Click on the product to see details | <Click> | Data is shown | Data is shown | Owner contact details are shared. | Pass |
| Error | <Invalid > | Invalid Error | Invalid Error | Re-Fresh webpage and try | Pass |
|  |  |  |  |  |  |  |  | again |  |

**CHAPTER 7 CONCLUSION AND DISCUSSION**

**7.1 Self Analysis of Project Viabilities**

* SMART & DIGITAL SOCIETY is a mobile application for residential Societies.
* It comes with a host of secure features to help you manage your Society matters virtually and efficiently.
* This app completely automates all your routine Societal tasks such as visitors’ registry, event, service center, complaint filing, car's Parking, maintenance bill towards Society & Utilities & many more.
* **Some of our best features -**

1. Visitor Management - Allow/Reject visitor entry right from your app. Get notified whenever a visitor is at the gate and get control over the access, they can have to you.
2. Visitor Digital Logs - Say goodbye to the manual visitor books! Keep visitor logs on our app, which are securely stored on cloud. Stay safe and organized with Smart Society App.
3. Billing - Utility bills, maintenance bills will be given online through our Smart & Digital society Application.
4. Digital Notice Board - All the important events or activities happening in the housing society.

**7.2 Summary of Project work:-**

* A society management and billing project that effectively manages and handles all the functioning of a co-operative 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 parking, cultural funds, emergency funds and other charges and adds them 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. The system consists of automatic bill generation facility. It calculates various associated costs, adds them up and provides a bill accordingly. This system is developed in android so it is available for all android based smart form and it is saving your lots of time and work.

**CHAPTER 8 LIMITATION AND FUTURE ENHANCEMENT**

* Only few amounts of data can be visible at a time
* Data like notice generate, maintenance, Event details, complaint box details are limited to see we increase size of visible details.
* Only limited to Android users
* We develop this application in android studio so it’s only limited to android user in future we add this app to iOS user also.
* Can’t make online call and meeting.
* We are not any online service to this app in future we add a different section for online call and meetings.
* Can’t make society Accounting.
* We also add a section for calculating Accounting for societies.

* Can’t display society information.
* When multiple society are connected then we also add section for display the information about societies.