# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

**An Internship Project Report**
**on**

## *Apartment Maintenance*

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

**by**


**Aditya C**
**1RN18IS007**


**Under the Guidance of**


**Mr. Pramoda R**
**Assistant Professor**
**Department of ISE**


ESTD:2001
*An Institute with a Difference*


## Department of Information Science and Engineering

## RNS Institute of Technology

**Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,**
**Channasandra, Bengaluru-560098**


**2021-2022**

# RNS INSTITUTE OF TECHNOLOGY

**Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,**
**Channasandra, Bengaluru - 560098**

## DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

## CERTIFICATE

Certified that the Internship work entitled ***Apartment Maintenance*** has been successfully completed by **Aditya C (1RN18IS007)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

| **Mr. Pramoda R** | **Dr. Suresh L** | **Dr. M K Venkatesha** |
|---|---|---|
| Internship Guide<br>Assistant Professor<br>Department of ISE | Professor and HoD<br>Department of ISE<br>RNSIT | Principal<br>RNSIT |

**External Viva**

| **Name of the Examiners** | **Signature with Date** |
|---|---|
| **1.** _____ | **1.** _____ |
| **2.** _____ | **2.** _____ |

# DECLARATION

I, **Aditya C [USN: 1RN18IS007]** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Apartment Maintenance*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :

**Aditya C**

**(1RN18IS007)**

# ABSTRACT

In recent years, with the rapid urbanization and increase in population, there has been a tremendous demand for housing across all metropolitan cities. Apartments or multi storied residential complexes are cropping up in every nook and corner of cities. On one hand these apartments cater to the sky rocketing demand for housing and on the other hand these apartments have their own drawbacks of carrying out regular maintenance.Usually maintenance activities is carried out by a trusted group of residents who helm on to the responsibilities like President, Vice president, Secretary, Treasurer, Chief etc. Now, the residents are not regularly aware of maintenance works happening in their society. To bridge this gap, we have implemented a Apartment Maintenance app . It is an application implemented using flutter which can be run on Browser, Android smartphones and Iphones. The services that this app on a smartphone provides are plenty.

This App has got features like Resident Login, Event Reminder, Memberships, events, Complaints, Visitors and Help Desk. Each feature has been implemented with separate screens. The App works for these given features and more features can be added/modified later. In addition, you can raise complaints through this App. For every selected feature, a UI pops up and guides the user with the rest of the process.The helpdesk has the numbers of all the required service professionals.

# ACKNOWLEDGMENT

At the very onset I would like to express our gratitude to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First of all, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to our beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

We are extremely grateful to our own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

We place our heartfelt thanks to **Mr Pramoda R** Assistant Professor , Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

I thank **Mr. Akshay D R, Co-Founder and CEO, Enmaz Engineering Services**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

**ADITYA C**

# TABLE OF CONTENTS

# List of Figures

# List of Abbreviations

| | |
|---|---|
| SQL | Structured Query Language |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| BFM | Behavioral Feature Matrix |
| GFM | Grammatical Feature Matrix |
| SDLC | Software Development Life Cycle |
| HTTP | Hypertext Transfer Protocol |
| DSDEPHR | Distributed Storage design for Encrypted Personal Health Record |
| HD | Hadoop Database |
| SRS | Software Requirement Specification |

# Chapter 1

# INTRODUCTION

## 1.1 Introduction to Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia,Web platform, and the web from a single codebase. First described in 2015, Flutter was released in May 2017. Flutter provides a clean interface for custom embedders that can power Flutter apps on new hardware and operating systems.Since Dart is portable, Flutter can use the same rendering stack no matter which embedder spins it up, maximizing code reuse.Flutter's platform channels can put a single Dart interface on native code for mobile, web, desktop, or your embedded platform.

## 1.2 History

The first version of Flutter was known by the codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit[6] with the stated intent of being able to render consistently at 120 frames per second.During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.

On May 6, 2020, the Dart software development kit (SDK) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with new CanvasKit renderer and web specific widgets, early-access desktop application support for Windows,

macOS, and Linux and improved Add-to-App APIs.[9] This release included sound null-safety, which caused many breaking changes and issues with many external packages, but the Flutter team included instructions to mitigate these changes as well.

On September 8th, 2021, the Dart SDK in version 2.14 and Flutter version 2.5 were released by Google. The update brought improvements to the Android Full-Screen mode and the latest version of Google's Material Design called Material You. Dart received two new updates, the newest lint conditions have been standardized and preset as the default conditions as well Dart for Apple Silicon is now stable.

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It was unveiled in November 2007.

## 1.3 Dart Platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.[10]

On Windows, macOS, and Linux[11] Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.[12]

For better performance, release versions of Flutter apps targeting Android and iOS are compiled with ahead-of-time (AOT) compilation

## 1.4 About Apartment Maintenance

Property maintenance relates to the upkeep of a home, apartment, rental property or building and may be a commercial venture through a property maintenance company, an

employee of the company which owns a home, apartment or a self-storage pastime for example day-to-day housekeeping or cleaning

In a residential environment, a building manager will typically supervise a team of porters or concierge, cleaners, electrical and mechanical contractors, and depending on the size of the development, a team of administrative staff. If the development comprises several blocks, it is common that the Building Manager will report to an estate manager although both titles have become interchangeable. To a lesser extent, the term "development manager" is also used. Traditionally, this role's title was "house manager". The disparity in the job titles can reflect some differences in the job description but in essence, the title that perhaps best defines the role is that of building services manager as the main aspect of the job relates to the day-to-day running of the development with particular focus on the maintenance, site staff management, health and safety and presentation of the building or residential complex. The biggest challenge in the role is to manage residents' expectations and match these up to the budget constraints and prevalent legal requirements

# Chapter 2

## LITERATURE REVIEW

This project was done by referring to about 5 reference paper from various sources like springer, ieee, elsevier etc. The project has been developed based on the future enhancements in these papers. Careful Analysis has been done on each and every paper and implementation has been done. We studied the implementation of flutter from a few papers and learnt about the technologies and extensions it requires.

Following is the list of papers referred along with the associated links:

1. Modern App Development with Dart and Flutter : This research paper details about how flutter apps should be run and deployed. It has a comprehensive lit of all flutter tools and frameworks which can be used to do this project. Detailed explanation of Dart is available. The paper introduces the programming language Dart, the language used for Flutter programming. It then explains the basics of app programming with Flutter in version 2. Using practical examples such as a games app, a chat app and a drawing app, important aspects such as the handling of media files or the connection of cloud services are explained. The programming of mobile as well as desktop applications is discussed.

2. Application development using Flutter : This research paper is all about flutter application development specially for the Android Interface. As we are primarily focusing on Android interface, this research paper is useful in our project. It has the list of all android compatible libraries and package os flutter which cater to Android smartphones. Cross-platform mobile application development is the pressing priority in today's world and generation. Developers are enforced to either construct the same application numerous times for various OS (operating systems) or accept a low-quality similar solution that trades native speed and accuracy for portability.

Flutter is an open-source SDK for developing high-performance and more reliable mobile applications for operating systems like iOS and Android. Significant features of the Flutter are Just-in-time compilation which executes the computer code that encompasses compiling during program execution at run time rather than preceding execution. More frequently, this comprises of bytecode translation lesser-known as source code to machine code, which is unswervingly executed

3. A prototype of high rise residential home management system : Residential high-rises are unique properties that differ from landed properties such as bungalows or terrace houses. They are unique insofar as, after the properties have been occupied, facilities must be jointly managed by residents. The continuous growth of high-rise residential buildings specifies that there is a need for effective ownership and property management system to instill a valuable living experience among high-rise residents in this country. Looking at this scenario, this research takes an initiative to develop a mobile application that supports streamline information with assistive features to improve communication and information sharing for high-rise residential management in Malaysia. This aim is achieved by answering three research objectives, including identifying user requirements, designing the application prototype and developing the artefact and evaluating the artefact.

4. Android Based Responsive Approach for Residential Society Maintenance : This research paper is all about a study of residential complexes and maintenance apps in usage. Housing societies are an immense responsibility to manage. With their sizes increasing every day to take form similar to integrated townships, managing them has become task of its own. Today"s modern society needs a much modern solution for its proper management, which can replace the human efforts. Any Co-operative Society or Housing Society need a hassle-free and efficient society management system for managing its day-to-day activities. Society management system has not only to deal with its complaints but has to keep an eye on all other facility and activities under the society premises. Thus taking help of society management software makes us feel free.

5. ADMINISTRATION SYSTEM FOR END TO END LUXURY APARTMENT MANAGEMENT SOFTWARE :

   A research paper on another Project on Apartment Maintenance. This project goes in detail about the implementation. In this era of modern technology renting an apartment is within our hands. But the challenges after that is time consuming and misleading. With the current change in outlook in innovative field, there is an earnest need to embrace and appreciate the force of innovation. Property management remains watchful to confront the difficulties of progress by utilizing another methodology that encourages simple administration of properties. Subsequently there is need to develop a management system for the property and this board framework that can improve on work for the property owners and tenants with the goal that all their work can be productive and successful.

# Chapter 3

# ANALYSIS

## 3.1 Introduction

Collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. 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.

## 3.2 Existing System

The existing system has many loopholes and limitations. It has too much manual work from filling a form to preparing a document, this increases burden on the workers which doesn't yield the result it should. If any modification has to be made it is completely manual and is error prone. The present systems does not use advanced technologies with minimum user interaction which causes more hustle and less satisfaction for the users. It contains less functionalities for the tenants. A proper verification is not maintained in these systems.

**Limitations of existing system are:**

- Unprotected guest data.
- Takes time in retrieving single data from the database.
- It only provides text based interface which is not as user friendly as Graphical user interface.

## 3.3 Proposed System

The proposed software is a web application that gives all the features provided by the existing systems, with enhanced specifications. This software provides equal importance to the needs of the owner and the tenant. It will provide smooth functioning for all the modules and the data is stored in a sorted manner.

With the use of scripting and programming language an interesting software has been made for the user. Various pages are created about the apartment, owner, tenant, employee, maintenances, rent payment details, complaints and also the visitors to the apartments.

## 3.4 Use Case Diagram

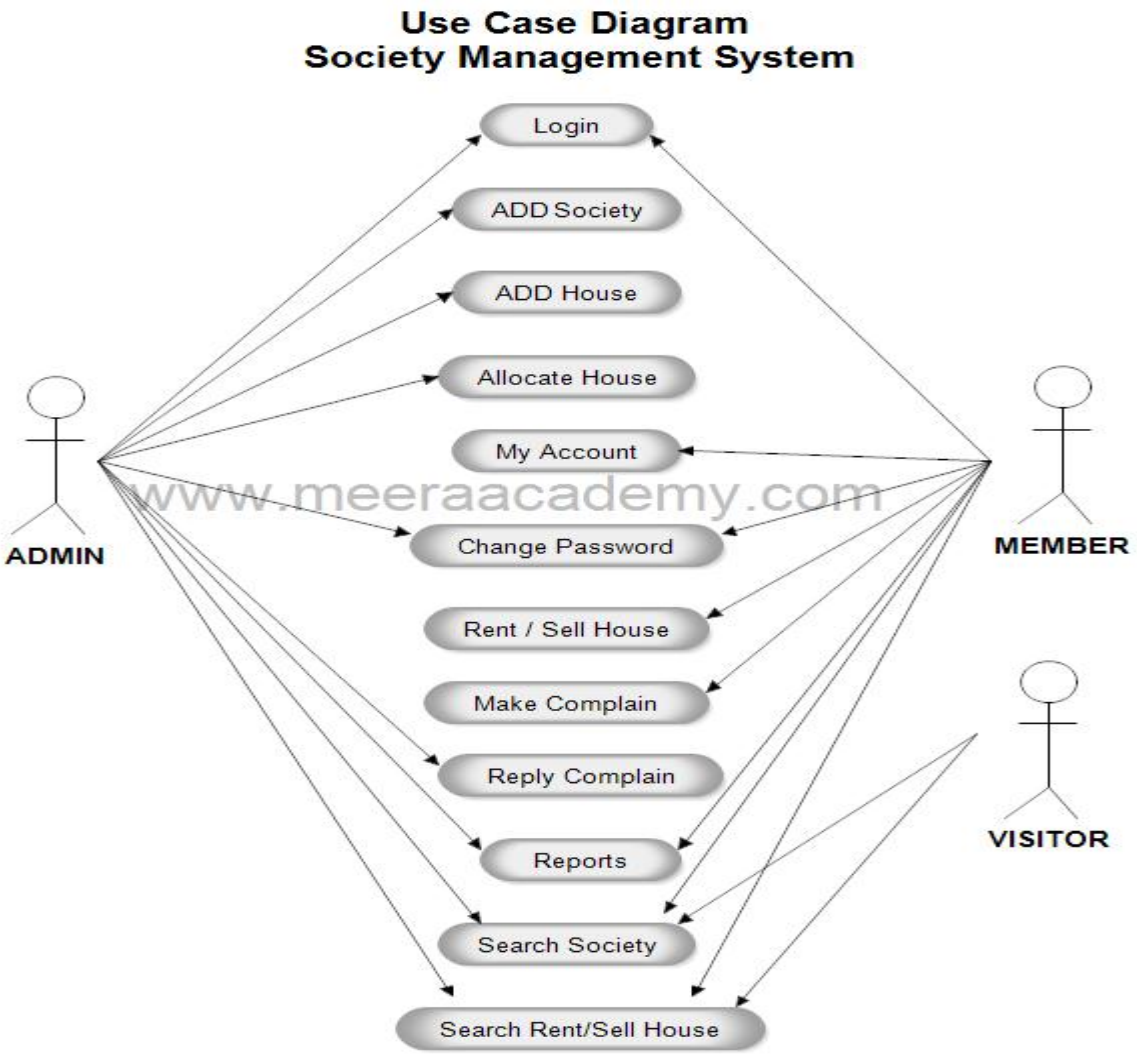


Fig 3.1: Use case Diagram

# Chapter 4

# SYSTEM DESIGN

## 4.1 Introduction

The database was updated each time the administrator; add, deletes or deletes data on the system. It's only the administrator who has access to the system to view or make changes when necessary. The system was designed to allow the administrator to view, edit, delete and add data to the database.

## 4.2 System Architecture and Diagram

### 4.2.1 General Flutter Widget Tree

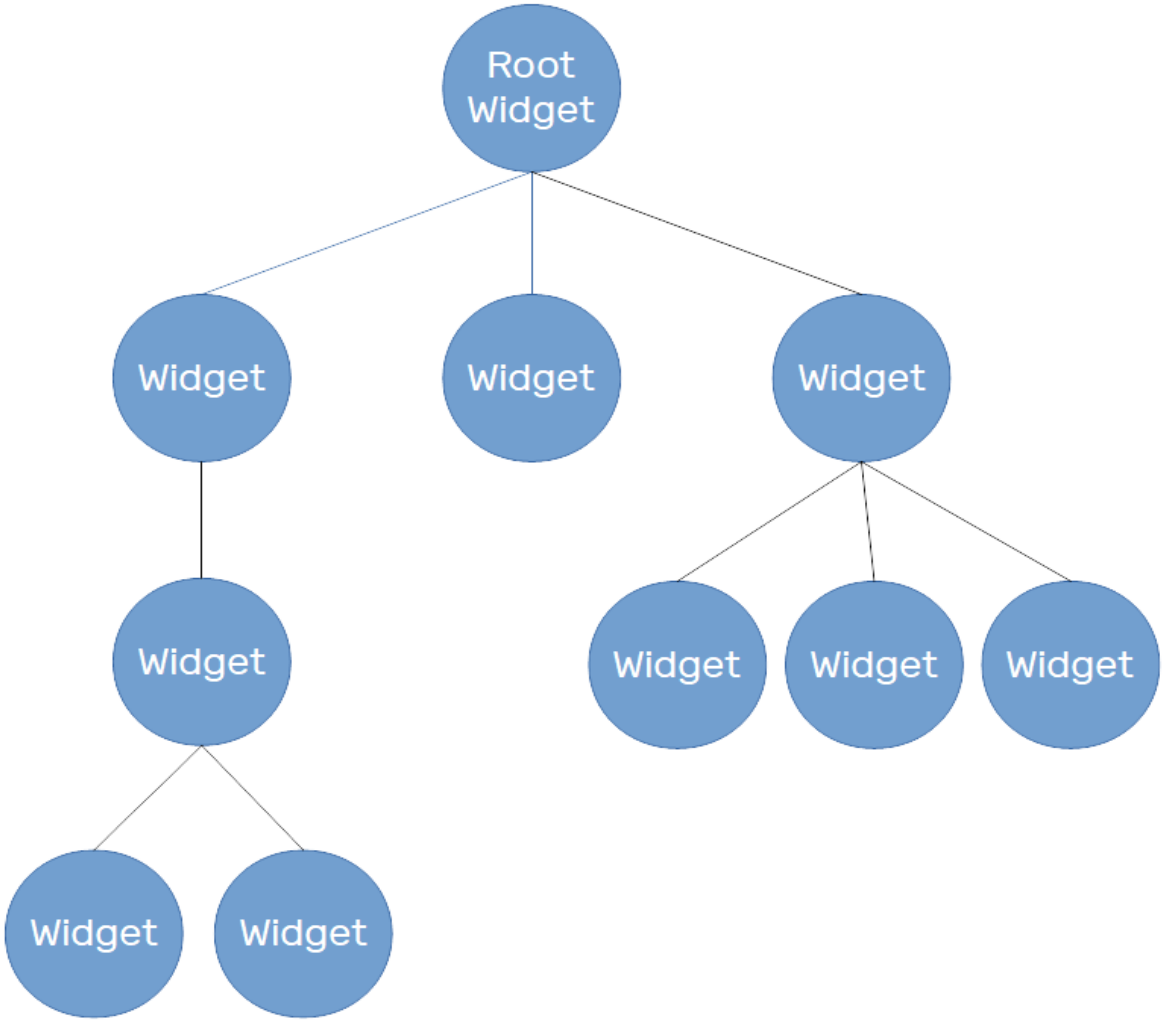

Fig 4.1: Flutter Widget Tree

### 4.2.2 Login Screen

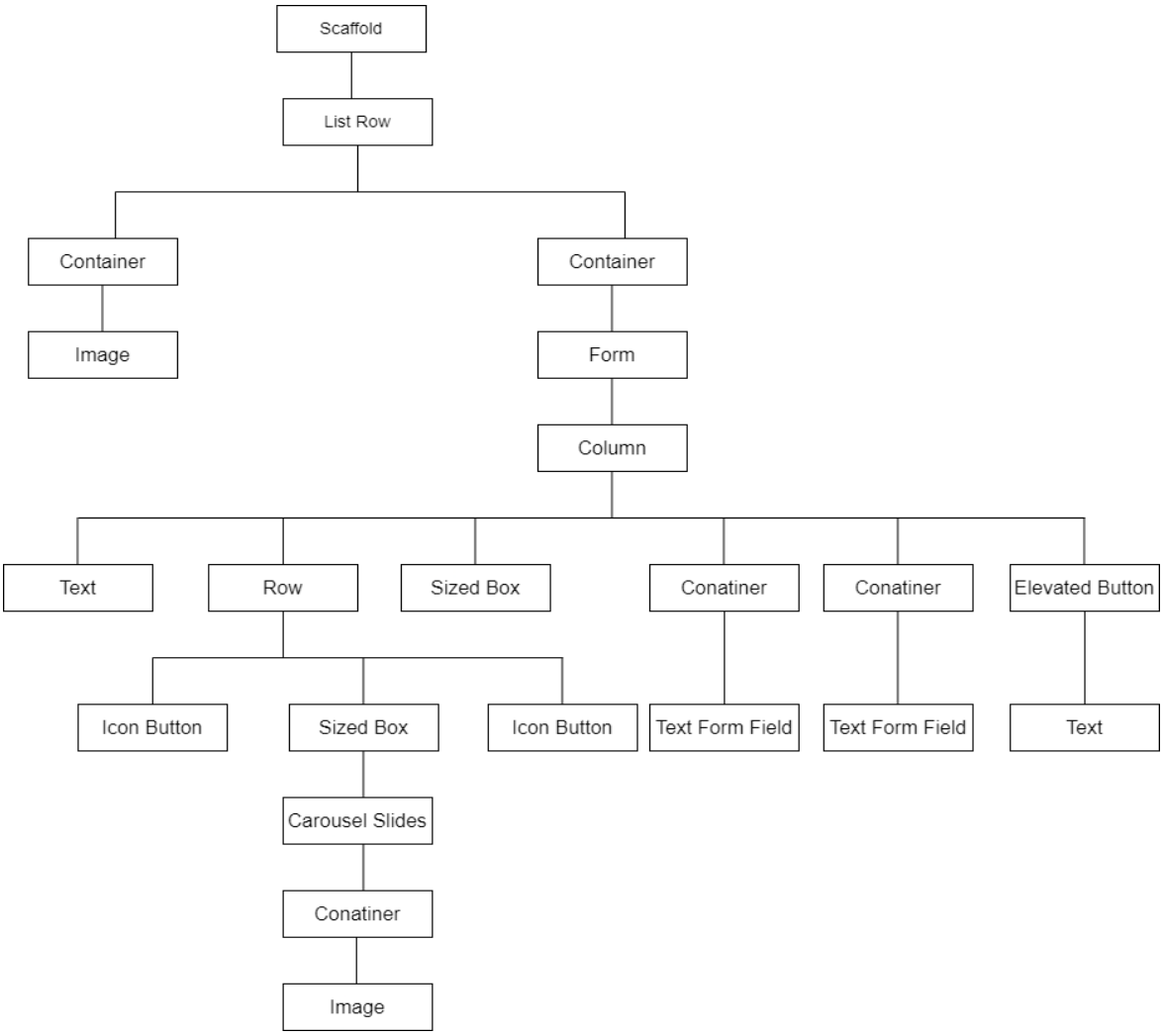


Fig 4.2: Container Flow Diagram

### 4.2.3 Main Screen

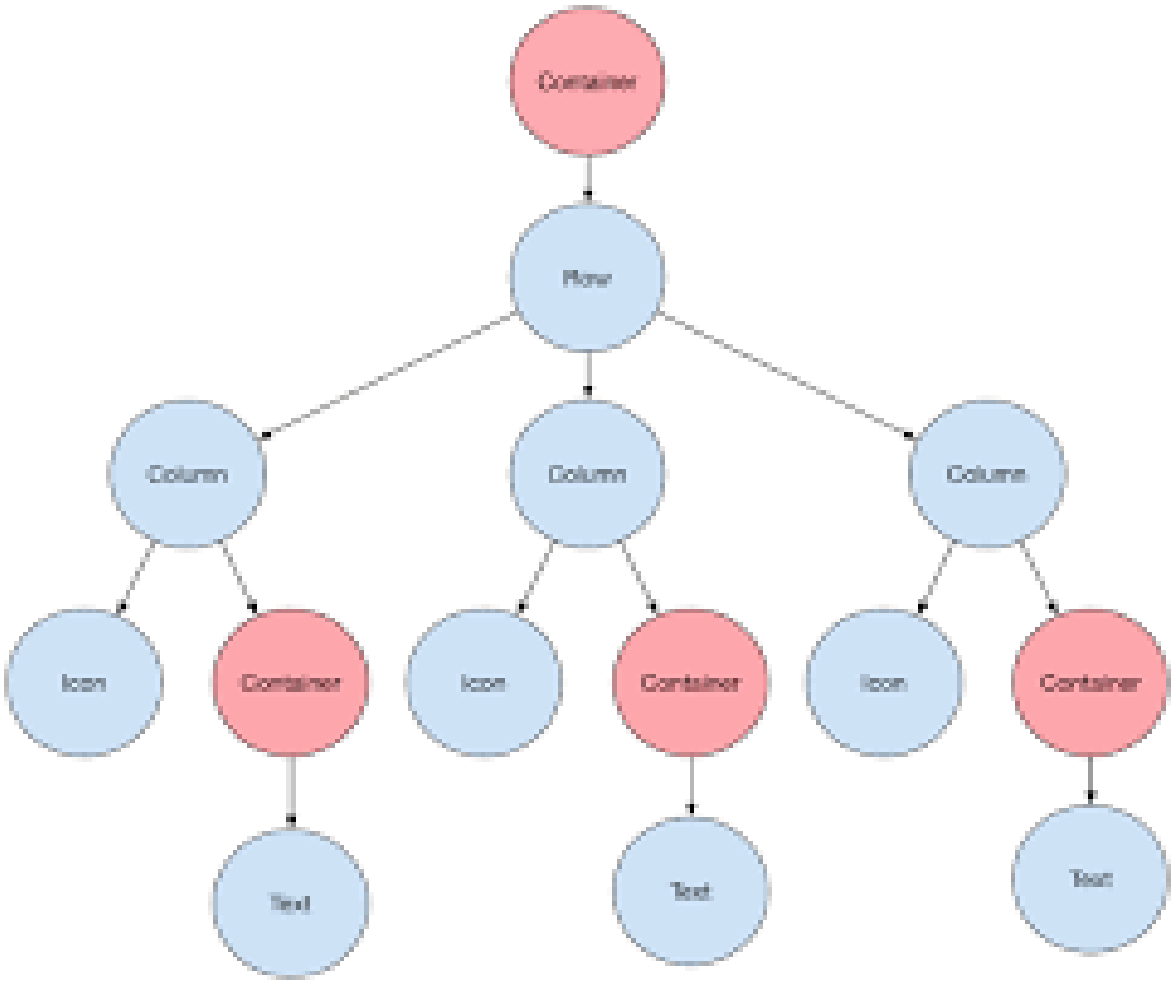


Fig 4.3: Main Screen Diagram

# Chapter 5

# DETAILED DESIGN

## 5.1 Architecture

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap. An architectural diagram must serve several different functions. To allow relevant users to understand a system architecture and follow it in their decision-making, we need to communicate information about the architecture. Architectural diagrams provide a great way to do this.

To put down some major functions, an architectural diagram needs to:

- Break down communication barriers
- Reach a consensus
- Decrease ambiguity

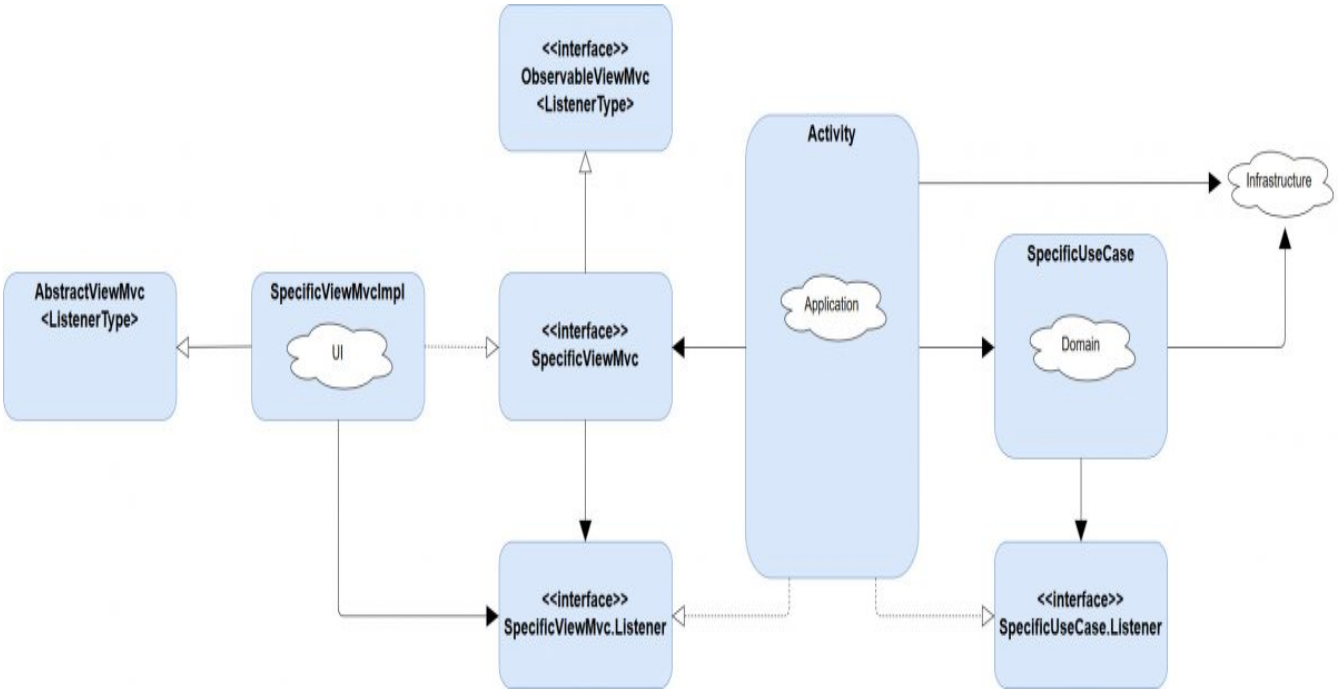


Fig 5.1: Detailed Architecture Diagram

User interface logic is the logic that defines how application looks and captures user's interaction with apps' screens.

Decoupling of user interface logic from your Activities is hands down the best investment into long-term quality of the code. The reasons for that are numerous:

1. UI logic usually has the most detailed requirements (UI mockups)
2. Its rate of change is usually much higher than the rest of the application
3. Code that describes user interface is hacky, messy, unclear, etc. (especially on Android)
4. Manual testing of user interface logic is easy
5. Automated testing of user interface logic is very hard and you can't test everything

## 5.2 Data Flow Diagram

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. Its focus is on the flow of information, where data comes from, where it goes and how it gets stored. The system supports multiple access of users at the same time and hence the data flow is shown from both sides.

Level 0:

Level 0 indicates the multiple users accessing the software from different areas at different time. The user can either be an admin, Owner or a tenant.

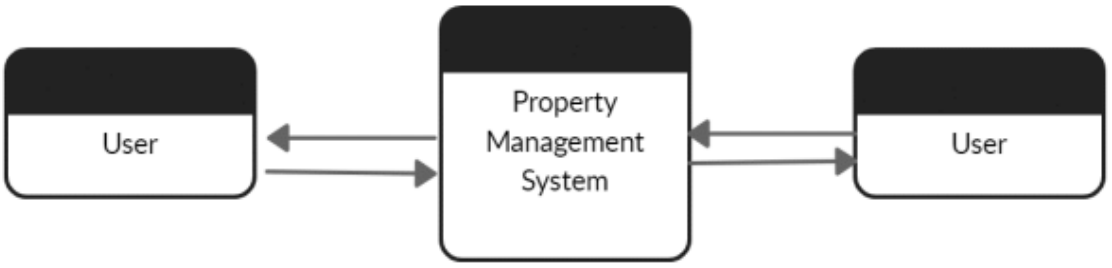


Fig 5.2: Level 0 Diagram

Level 1:

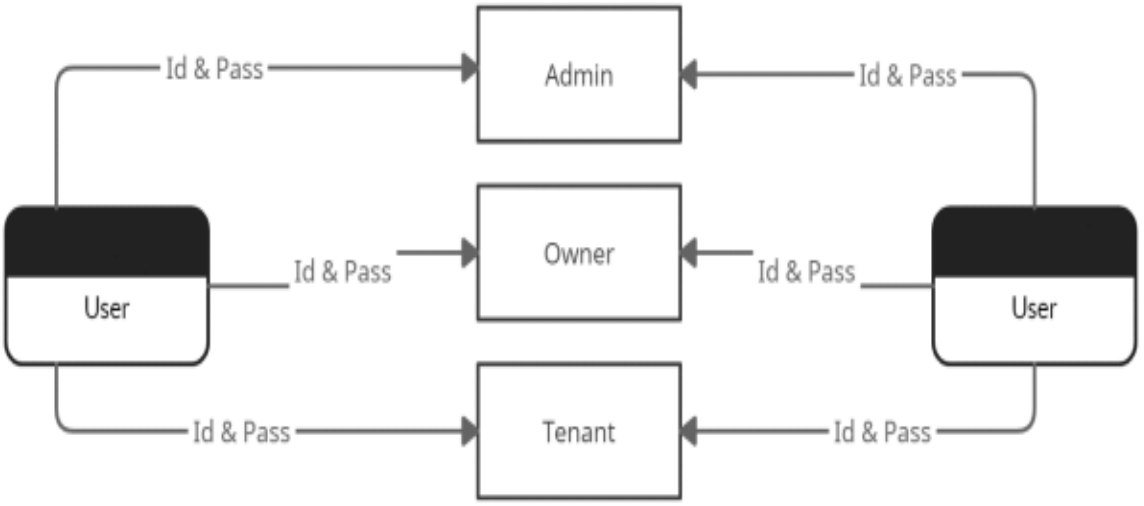


Fig 5.3: Level 1 Diagram

Level 2:

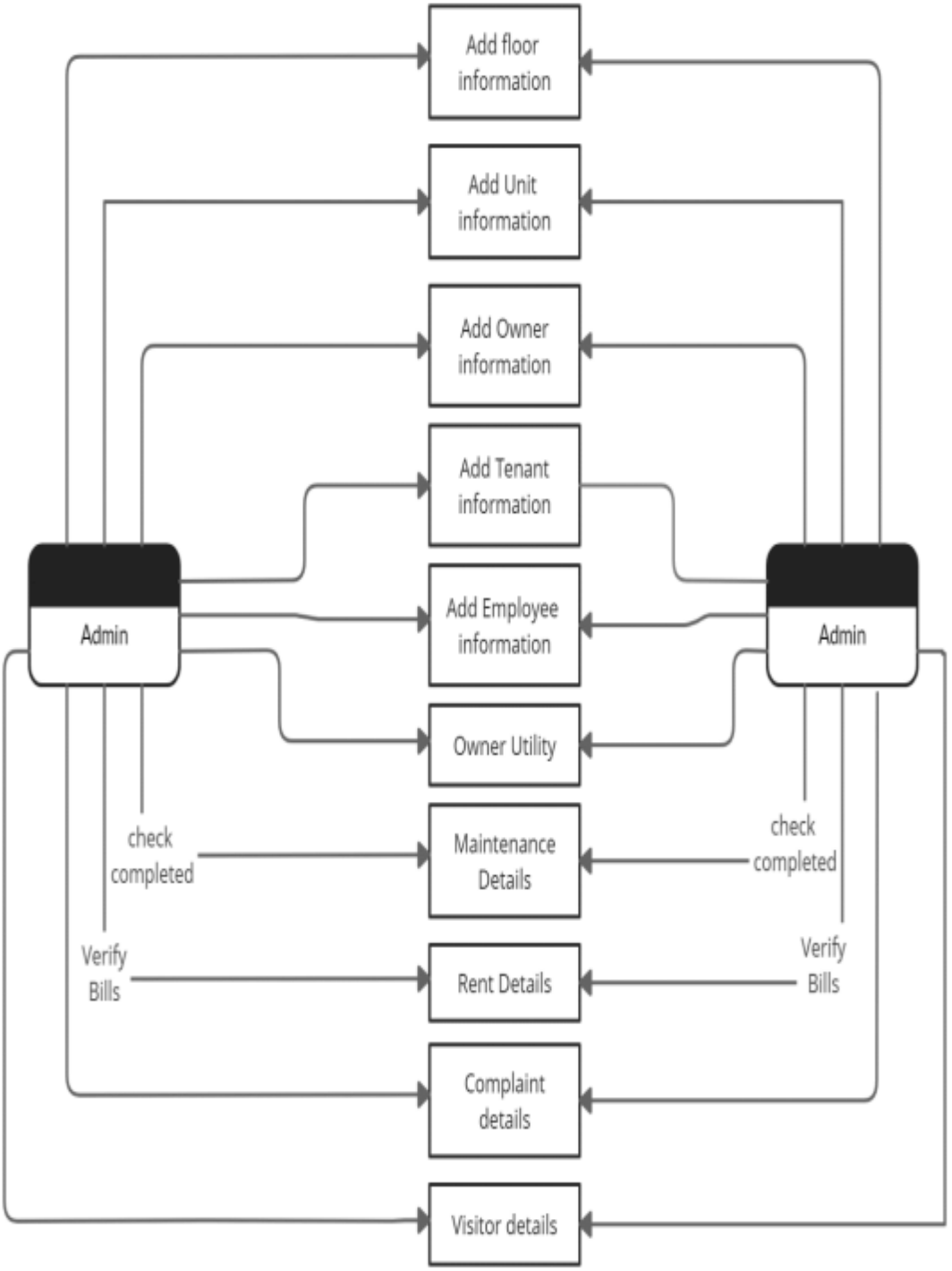


Fig 5.4: Level 2 Diagram

Level 3:

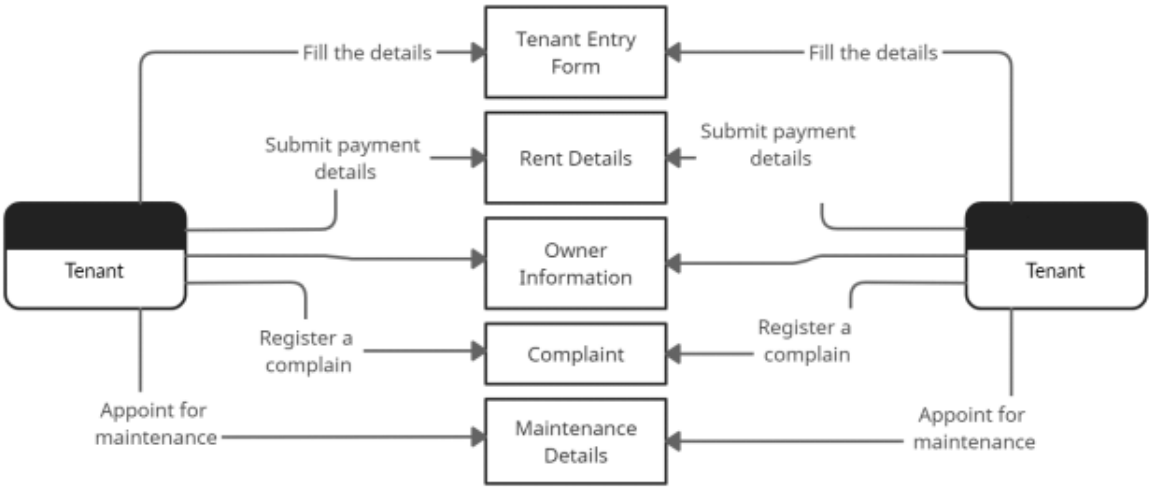


Fig 5.5: Level 3 Diagram

# Chapter 6

# IMPLEMENTATION

## 6.1 Requirement Specifications

### 6.1.1 Hardware Requirements

- CPU: Pentium processor and above
- RAM: 2 GB
- HDD: 40 GB

### 6.1.2 Software Requirements

- **Operating System:** Windows 8 and above
- **Front-end Design:** Visual Studio Code
- **Front-end Language**: Dart

### 6.1.3 Flutter

**Flutter** is an open-source UI, software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, Web Platform and the web from a single codebase.

The major components of Flutter include:

- Dart Platform
- Flutter Engine
- Foundation Library
- Design-specific widgets
- Flutter Development Tools (DevTools)

### 6.1.3.1 Dart Platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On Windows, macOS, and Linux Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful, hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state

### 6.1.3.2 Flutter Engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform Specific SDKs such as those provided by Android and iOS. The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain.

### 6.1.3.3 Foundation Library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

#### 6.1.3.4 Design Specific Widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and *Cupertino* widgets implement Apple's iOS Human Interface Guidelines.

## 6.2 Discussion of Code Segments

### 6.2.1 main.dart

```dart
import 'package:flutter/material.dart';
import 'package:flutter_screenutil/flutter_screenutil.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:metaville2/screens/constants.dart';
import 'package:metaville2/screens/loginscreen.dart';

Run | Debug | Profile
void main() {
  runApp(ScreenUtilInit(
    designSize: Size(375, 812),
    builder: () => MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Navigation Basics',
      theme: ThemeData(
        scaffoldBackgroundColor: Constants.scaffoldBackgroundColor,
        visualDensity: VisualDensity.adaptivePlatformDensity,
        textTheme: GoogleFonts.poppinsTextTheme(),
      ), // ThemeData
      home: Login(),
    ), // MaterialApp
  )); // ScreenUtilInit
}
```

### 6.2.2 login.dart

```dart
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:flutter/widgets.dart';
import 'dart:async';

import 'package:metaville2/screens/dashboard.dart';

class Login extends StatefulWidget {
  var photos = ['images/building.png','images/family.png'];

  @override
  _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {
  FocusNode focusEmail = new FocusNode();
  FocusNode focusPass = new FocusNode();

  var _timer;
  int _pos=0;

  @override
  void initState() {
    _timer = Timer.periodic(Duration(seconds: 7), (Timer t) {
      setState(() {
        _pos = (_pos + 1) % widget.photos.length;
      });
    }); // Timer.periodic
    super.initState();
  }
  @override
  Widget build(BuildContext context) {
    SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
      statusBarColor: Colors.transparent,
      statusBarIconBrightness: Brightness.light,
    ));
    return GestureDetector(
      onTap: (){FocusManager.instance.primaryFocus?.unfocus(); setState(() {
      });},
      child: Scaffold(
          resizeToAvoidBottomInset: false,
          backgroundColor: Colors.white,
          body: SafeArea(
            child: Container(
              margin: EdgeInsets.all(45),
              child: Column(
                crossAxisAlignment: CrossAxisAlignment.center,
                children: [
```

```
FractionallySizedBox(
  alignment: Alignment.topCenter,
  widthFactor: 0.85,
  child: Container(
    child: Stack(children: [AnimatedOpacity(opacity: _pos==1?1:0,
        duration: Duration(milliseconds: 1000),
        child: Image.asset(widget.photos[1])), // AnimatedOpacity
      AnimatedOpacity(opacity: _pos==0?1:0,
        duration: Duration(milliseconds: 1000),
        child: Image.asset(widget.photos[0])) // AnimatedOpacity
    ]), // Stack
  ), // Container
), // FractionallySizedBox
Container(
  alignment: Alignment.centerLeft,
  child: Text(
    'Login',
    style: TextStyle(
      fontFamily: 'OpenSans',
      fontSize: 28,
      fontWeight: FontWeight.bold,
    ), // TextStyle
  ), // Text
), // Container
SizedBox(
  height: 20,
), // SizedBox
Container(
  child: TextField(
    onTap: (){setState(() {
    });},
    focusNode: focusEmail,
    keyboardType: TextInputType.emailAddress,
    textInputAction: TextInputAction.done,
    style: TextStyle(
      color: Colors.black,
      fontFamily: 'OpenSans',
    ), // TextStyle
    decoration: InputDecoration(
      enabledBorder: UnderlineInputBorder(
        borderSide: BorderSide(
          color: Colors.grey.shade400,
        ), // BorderSide
      ), // UnderlineInputBorder
```

### 6.2.3 Dashboard

```
@override
Widget build(BuildContext context) {
  SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
    statusBarIconBrightness: Brightness.dark,
  ));
  return Scaffold(
      body: Material(
        color: Colors.grey[200],
        child: SingleChildScrollView(
          child: Column(
            children: [
              Container(
                padding: const EdgeInsets.only(left: 16, right: 16, top: 48),
                child: Column(
                  mainAxisAlignment: MainAxisAlignment.start,
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Row(
                      mainAxisAlignment: MainAxisAlignment.end,
                      mainAxisSize: MainAxisSize.max,
                      children: <Widget>[
                        InkWell(
                            onTap: () {},
                            child: Container(
                              width: 50,
                              height: 50,
                              decoration: BoxDecoration(
                                image: DecorationImage(
                                  image: NetworkImage(
                                      "https://st3.depositphotos.com/15648834/17930/v/600/depositphotos_179308454-stock-illustration-unknown-perso
                                fit: BoxFit.cover,
                                ), // DecorationImage
                                borderRadius:
                                BorderRadius.all(Radius.circular(50.0)),
                              ), // BoxDecoration
                            )) // Container // InkWell
                      ], // <Widget>[]
                    ), // Row
                    Column(
                      children: [
                        Align(
                          alignment: Alignment.topLeft,
                          child: RichText(
```

```
          alignment: Alignment.topLeft,
          child: RichText(
            text: TextSpan(
                text: "Welcome back, Adithya!",
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 16,
                )), // TextStyle // TextSpan
          ), // RichText
        ), // Align
        SizedBox(
          height: 20,
        ), // SizedBox
        Align(
          alignment: Alignment.topLeft,
          child: RichText(
            text: TextSpan(
                text: "You have got 3 events\nthis week!",
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 24,
                  fontWeight: FontWeight.bold,
                )), // TextStyle // TextSpan
          ), // RichText
        ), // Align
        SizedBox(
          height: 30,
        ), // SizedBox
      ],
    ), // Column
    Container(
      height: 190,
      child: PageView(
        controller: PageController(viewportFraction: 0.9),
        scrollDirection: Axis.horizontal,
        pageSnapping: true,
        children: <Widget>[
          _buildEventsCard(
              title: "Residents Weekly\nMeeting",
              subject: "Upcoming festivals",
              time: "7 PM - 8 PM, Mon",
              imageList: images1,
              count: 22),
          _buildEventsCard(
              title: "Grounds Maintenance\nMeeting",
              subject: "Cleanliness Guidelines",
              time: "5 PM - 6 PM,  Wed",
```

```
                    SizedBox(
                      width: 10,
                    ), // SizedBox
                    IconButton(
                      icon: Image.asset('images/fb.png'),
                      iconSize: 50,
                      onPressed: () {},
                    ), // IconButton
                    SizedBox(
                      width: 10,
                    ), // SizedBox
                    IconButton(
                      icon: Image.asset('images/twitter.png'),
                      iconSize: 50,
                      onPressed: () {},
                    ), // IconButton
                  ],
                ), // Row
                SizedBox(
                  height: 20,
                ), // SizedBox
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [ Text('New to MetaVille? ',
                    style: TextStyle(
                      fontSize: 14,
                      color: Colors.grey.shade600,
                      fontWeight: FontWeight.normal,
                    ), // TextStyle
                  ), Text('Register', // Text
                    style: TextStyle(
                      fontSize: 14,
                      color: Color(0xff0165ff),
                      fontWeight: FontWeight.bold,
                    ), // TextStyle
                  ), // Text
                  ],
                ), // Row
              ],
            ), // Column
          ), // Container
        ) // SafeArea
      ), // Scaffold
    ); // GestureDetector
  }
}
```

# Chapter 7

# TESTING

## 7.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process

- A software configuration that includes a software requirement specification, a design specification and source code.
- A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

## 7.2 Levels of Testing

### 7.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output.

Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by-step instructional document. The unit testing is the process of testing the part of the program to verify whether the program is working correct or not. In this part the main intention is to check the each and every input which we are inserting to our file. Here the validation concepts are used to check whether the program is taking the inputs in the correct format or not.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. Unit test cases embody characteristics that are critical to the success of the unit.

### 7.2.2 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program is working fine or not. This testing can be done by choosing the options in the program and by giving suitable inputs.

### 7.2.3 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. System testing is done after integration testing is complete. System testing should test functional and non-functional requirements of the software.

### 7.2.4 Validation Testing

In this, requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end user)

### 7.2.5 Output Testing

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

### 7.2.6 User Acceptance Testing

User acceptance testing is a type of testing performed by the end user or the client to verify/accept the software application to the production environment.

UAT is done in the final phase of testing.

# Chapter 8

# RESULTS

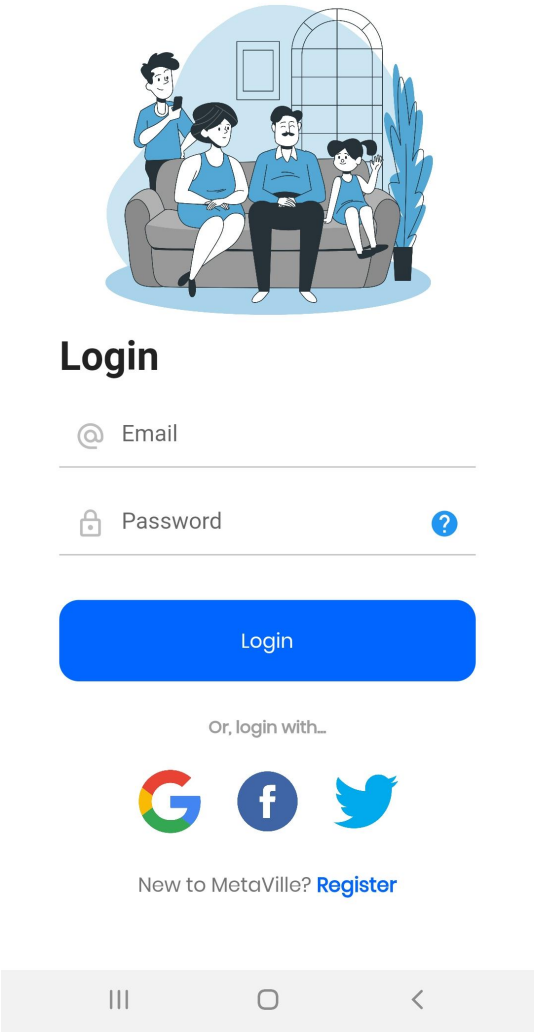


Fig 8.1: Login Screen

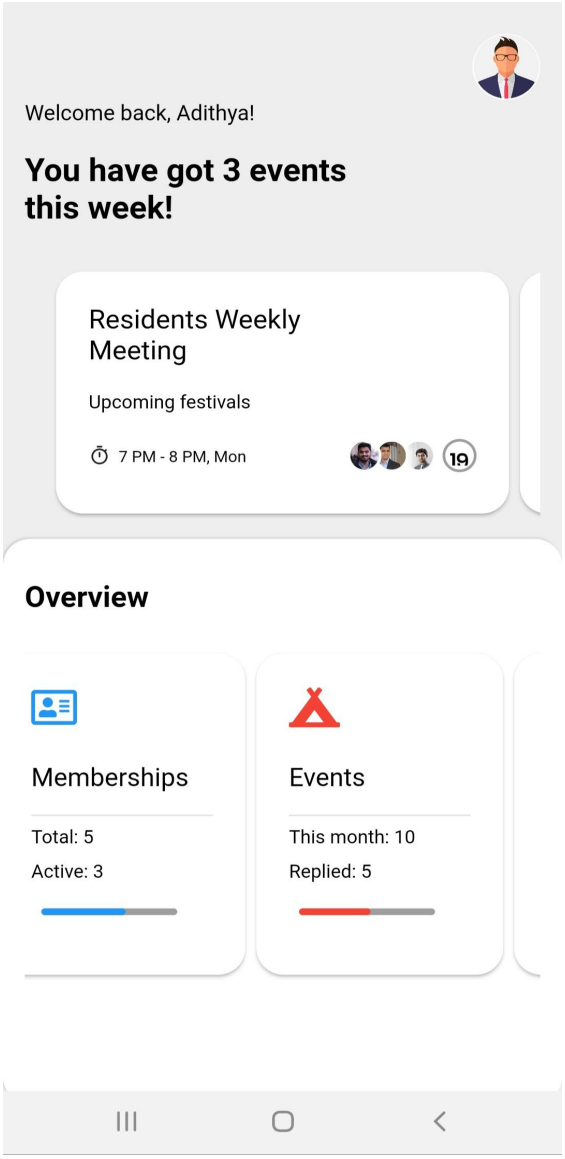


Fig 8.2: Main Screen

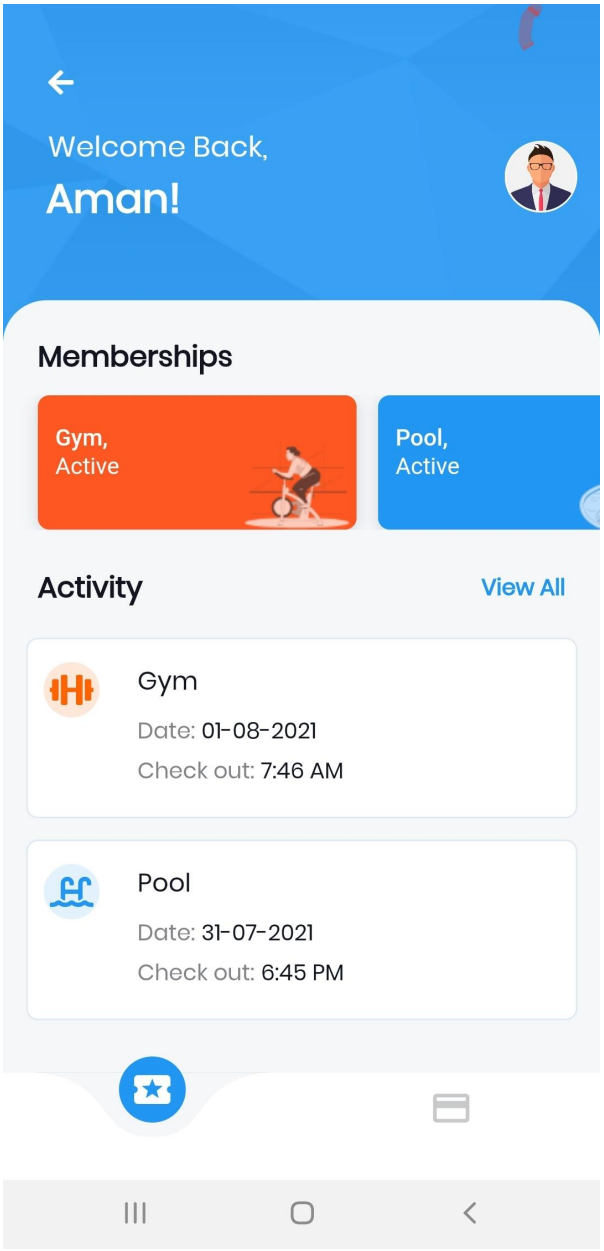


Fig 8.3: Memberships screen

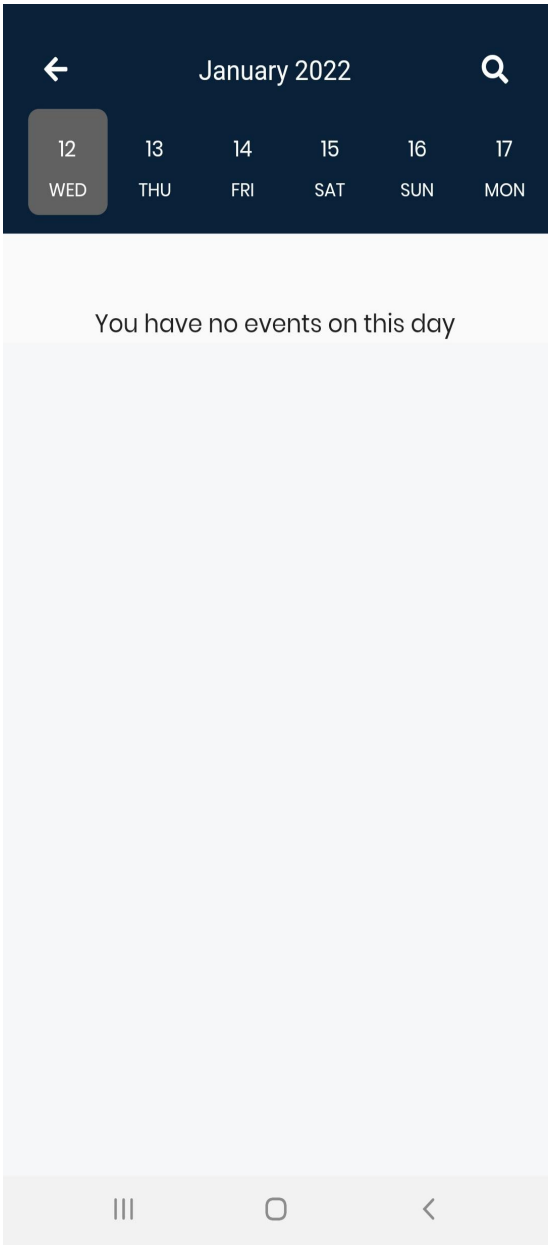


Fig 8.4: Events Screen

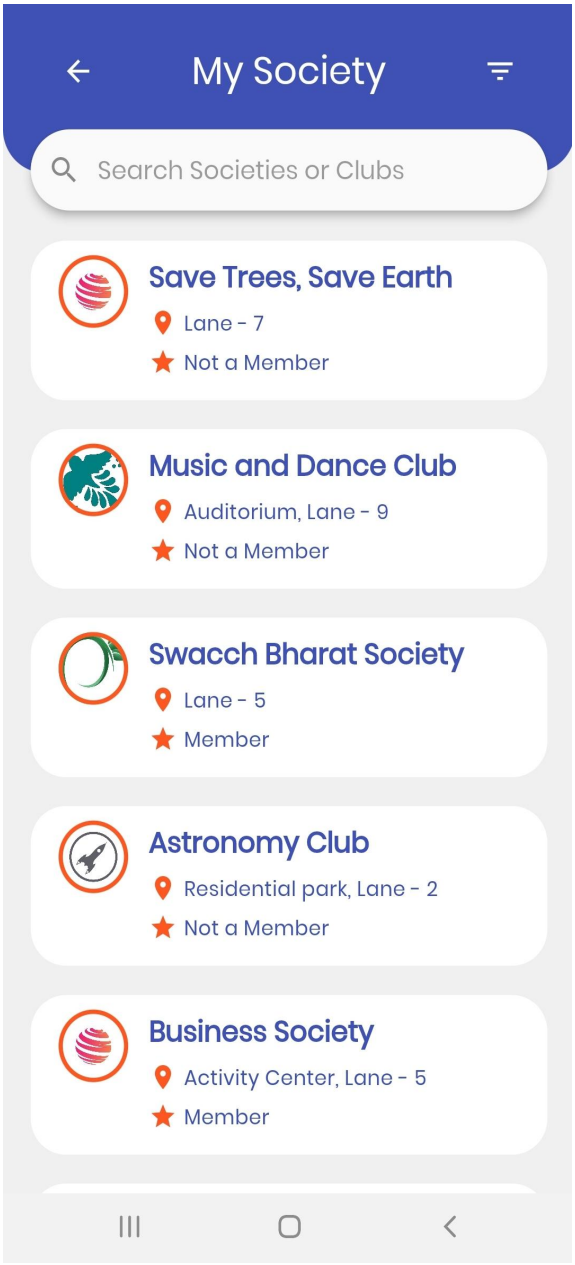


Fig 8.5: Clubs Screen

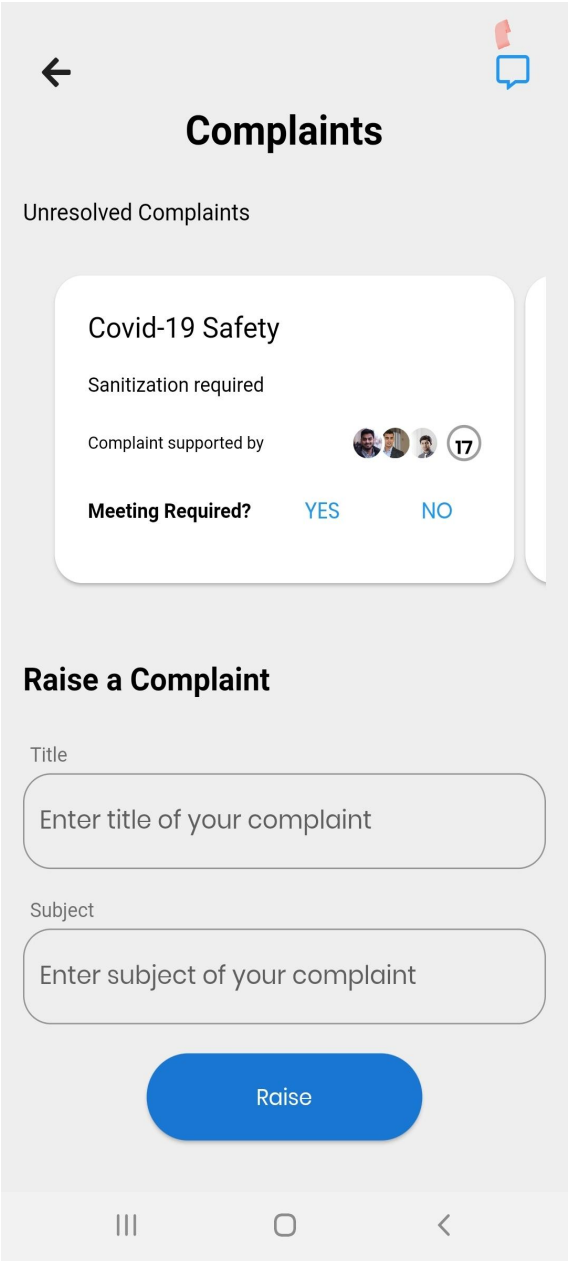


Fig 8.6:Complaints Screen

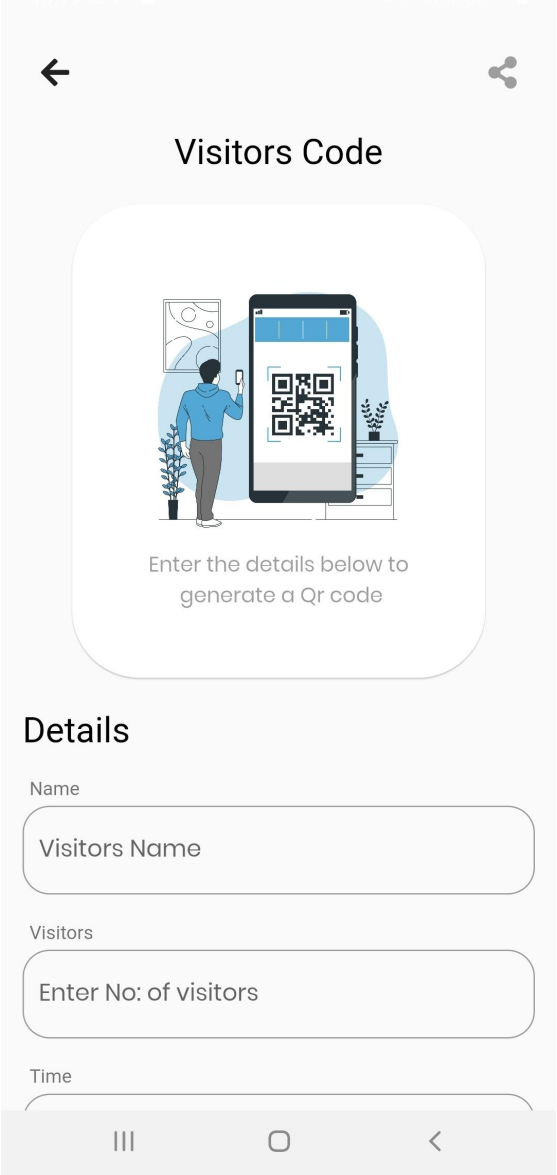


Fig 8.7: Visitors Screen

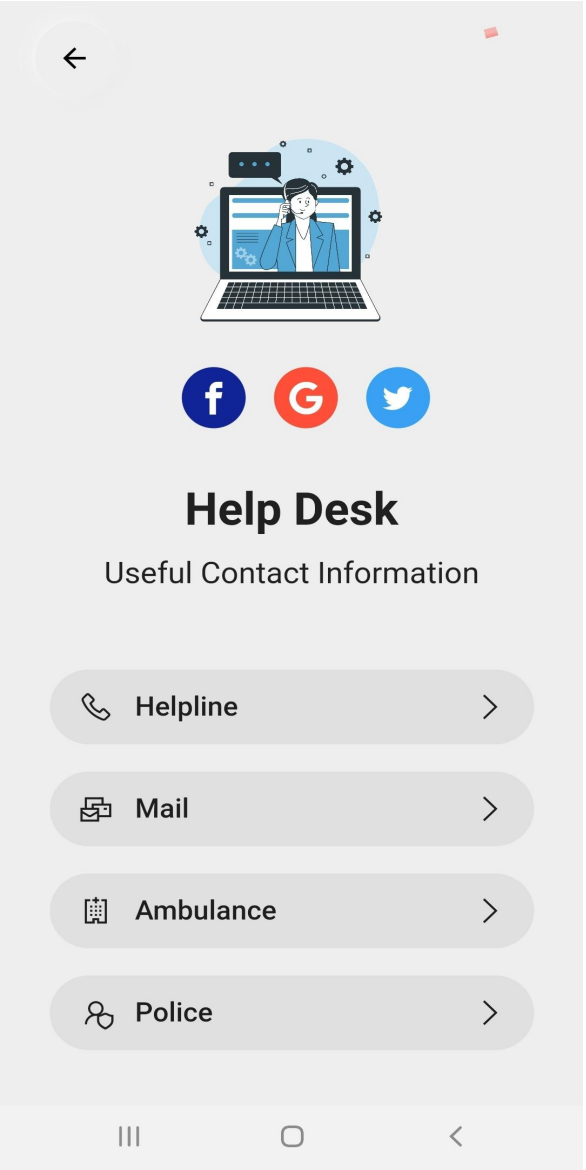


Fig 8.8: Helpdesk Screen

# Chapter 9

# CONCLUSION AND FUTURE WORK

## 9.1 Conclusion

- Easy to use and adaptable project for all users
- The UI is simplistic and easy to understand
- Residents can use the app to be informed about meetings, activities and infrastructure available in the society premises
- Residents can raise complaints and service requests in their flats using this App

## 9.2 Future Work

- A in-app payments can be included to take all maintenance charges
- UPIs or Payment Gateways can be integrated with this app
- Whatsapp chat links can be included for quicker messaging
- Utility service professionals can be contacted via this app
- Inter- apartment communications can be implemented

# REFERENCES


- Flutter : https://docs.flutter.dev/
- https://stackoverflow.com/questions/tagged/flutter
- The Dart Programming Language book by Gilad Bracha
- https://flutter.dev/community