A

Project Report

on

# Society Management System

By

Sharad Sharma (1813310183)
Sahil Siddiqui (1813310166)
Soumya Jaiswal (1813310214)
Kulyash Dahiya (1813310088)

Under the Supervision of
Mr. Eshank Jain

Submitted to the department of Computer Science and Engineering

For the partial fulfilment of the requirements
for award of Bachelor of Technology
in
Computer Science and Engineering


**Noida Institute of Engineering & Technology Gr. Noida**
**Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh,**
**India May, 2021-2022**

# Certificate

This is to certify that the Project report entitled "**Society Management System**" is a record of the work done by the following students:

| **Student name** | **Roll No.** |
|---|---|
| Sharad Sharma | 1813310183 |
| Sahil Siddiqui | 1813310166 |
| Soumya Jaiswal | 1813310214 |
| Kulyash Dahiya | 1813310088 |

This work is done under my/our supervision and guidance during the academic year of 2021-22. This report is submitted to the **Noida Institute of Engineering & Technology, Greater Noida** for partial fulfilment for the degree of **B.TECH. (Computer Science and Engineering)** of **Dr A.P.J Abdul Kalam Technical University, Lucknow, Uttar Pradesh, India.**

I/We wish him/her all the best for all the endeavors.

Signature of Guide:

Mr. Eshank Jain

# ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to **Mr. Eshank Jain, Department of Computer Science and Engineering, Noida Institute of Engineering & Technology**, Greater Noida, Gautam Buddha Nagar, Uttar Pradesh, India for her generous guidance, help and useful suggestions.

I express my sincere gratitude to **Prof. Chandra Shekhar Yadav, HOD CSE**, Noida Institute of Engineering & Technology, Greater Noida for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

**Date:**

**Student Name:**
Sharad Sharma
Sahil Siddiqui
Soumya Jaiswal
Kulyash Dahiya

# ABSTRACT

This study was done to show the benefits of using Flutter over other application development platforms. It deals with working on different platforms and their role in application development. In today's

world, ease of development is the thing every developer is looking for, and Flutter has built the foundation needed to support app development for both Android and iOS. Developers are forced to build the same app multiple times for different operating systems (operating systems) or settle for a similar low-quality solution at the trade-off of native speed and accuracy for native accuracy. mobile body. Flutter is an open source SDK for developing more reliable and high-performance mobile applications for operating systems like iOS and Android

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

To make it available to most users, a mobile app needs to be familiar withtwo independent platforms, Android and iOS. These two platforms have major differences that often require different skills to develop**.** For example, Java or Kotlin for Android and Objective C or Swift for iOS.

As a result, 4 developers and businesses in general struggled to cope withthe complexities involved in developing cross-platform applications. Flutter-like cross-platform framework has been discussed and implemented by different companies many times before.

However, that is not enough to meet the requirements of industrial development. Although the predecessor to was ineffective, Flutter, supported by Google, is gaining attention and developers are also findingit easier to use**.** Flutter apps can also work the same on both platforms, thereby reducing the cost and complexity of building apps on iOS and Android. Flutter was built completely from scratch and around August2017, only Google used it for commercial projects.

Flutter is a great tool from Google for creating cross-platform applicationswhich can be deployed to the web, desktop and mobile. Flutter is basically an open-source UI development kit to develop cross platforms

apps from a single code base. The earlier version of Flutter was known as codename “Sky” and first ran on the Android operating system.

It utilizes Dart, a quick OOP with a few valuable highlights. Flutter has its own user interface parts, alongside a motor to deliver them on the iOS and android stages. Most of those user interface sections, directly out of the container, fit in with the rules of Material Structure. Flutter doesn’t use any bridge to convert its widgets into native components which saves time and increases speed.

## 1.1 OBJECTIVE

The main aim of this system is to provide an Android application for society members that manages all the problems faced by a society that can be resolved by digitization. It should be generic for any society with minor customization.

It is an Ideal for use by society managing committee groups, the society management software keeps a track of all the society's billings and keeps a check on the society members, such as tracking members who have not paid their maintenance dues, etc.

Main Objective is to implement a management system for auditing the yearly collection of maintenance bills & expenditure. To maintain transparency. To visualize the availability of parking slots. To notify important issues & complaints about the existing problems (water, cleanliness) Emergency alertsin situations of catastrophe.

## 1.2 PROBLEM STATEMENT

• **Visitors Entry Management** –In this module, we will provide our user the ability to accept or deny the permit of their visitors.

• **Complaint Centre** -In this module, we allow our tenants to lodge a complaint any organizational/structural issue they have.

• **Payment services**: - Users can pay their Rent, management or utilities bills from payment gateway integrated application.

• **Digital Noticeboard**: - Users can also receive society notices and announcements from our app, a notification will be initiated to every resident whenever a new document is published.

• **Meeting organizer**: -In this section of our app, Users can arrange an association meeting or discuss the details for any events etc.

• **Emergency nos**.: -We also display all the nearby emergency contact information (such as Police, Hospitals, Security head etc.) in our app.

## 1.3 NEED AND SIGNIFICANCE

Today's Modern Societies without any Society ManagementSystems are just like Societies in olden days, with many Problems and less Solutions. For example: Just to call a Society Meeting a person has to go door to door to inform all the members of the meeting schedule, which is very time consuming. It was just because in olden days Societies were managed by people themselves, as they were having much Time and less Tasks. But today in the Era of Modern Technology Peoples have become busywith their works and also the Responsibilities of any Society has increased. Thus, to Fill this gap Between Peoples and their Society,"AXIOS " has come up with State-of-The-Art Society Management Software to solve all needs of Modern Society, which gives Flexibility to the user to manage all the tasks of their society with just a click. The Society Management Software helps in Managing the Documentation records, Sales/ Purchase management, Maintenance schedules, Amenities updates, etc. About Axios: In the age of Modern Technology, we are totally interlinked with Technology for our daily works. Thus, people have become smart and our societies have become smarter. With the help of software systems, it has become really easy to manage Our Housing Society or Residential Complexes. Discovering this need of managing the services, accounting, facilities, documentation of any society, Axios Software India Pvt. Ltd. has come up with a very Reliable and user-friendly "Society Management Software " in the field of Society management. This software enables us to manage any society or

residential apartments with all its services, accounting, maintaining records, managing the amenities, work management, Sales & purchases management with just a single click that too featuring a world class user interface. The software is an online base system, which gives us freedom to be up-to-date anywhere. Society Management Software: AXIOS is a website for maintaining the day-to-day activities of the operating societies. This is newly launched and has easy to use features with facility for online maintenance payments option for members. It provides online services like housing society accounting software, society management software and business management software. Axios giving successful and comprehensive services. Axios is a smart society management software.

## 1.4 EXISTING SYSTEMS

In the existing housing society management system, a traditional way of communication is used which includes a common notice board system operated by responsible society members. The data is stored in the files and the processing of the data is done manually and the report generation is slow. In the notice board system, one has to take the whole responsibility to operate and maintain the notice board. It creates dependability with the specific person.

Sometimes the person has to compromise with his own time schedule for these common activities. It is observed that complaints by society members are neglected by higher management because one has to take constant follow up until the issue gets solved.

### 1.2.1 Drawback of existing system

Residents don’t have that much control over their security details. Whenever a visitor arrives at the residence gate, and asks the security tolet him go in the premises using the house number of any resident, the resident won’t know about this until it’s too late.

Our application plans on giving more control to residents over this security and making management less of a liability.

# CHAPTER 2

# LITERATURE SURVEY

Robart A. Sowah and Seth Y. fiawoo., has discussed The components of designed and developed system include (1) a web application through which workers would input data at their various workplaces (2) a database hosted on a central server that would store information entered by workers (3) an application programming interface (API) that would take requests from the Android application, query the database and serve the results back to the Android application and (4) an android application that processes and displays results to users. The android application is developed using Android Studio in conjunction with android SDK tools. The application retrieves data from a database per user request and displays the retrieved information on an android device. Users of this application would be able to analyse data quicker hence make quick decisions as they would not be drowned in a flood of detailed information. There is also an added benefit of having access to company data on the go. Jarle Hansen, Tor Gronli., have focused on the principle of cloud computing and distributed computing. There are many handful technologies, which push data or content on mobile devices/tablets. The technologies studied are Google Cloud Messaging (GCM), C2DM (Cloud to Device Messaging) and Xtify is for authenticating a user, as well as handling all aspects of messages and delivery to the target application on the target device. The basic notification application is implemented using Google Cloud messaging YavuzSelimYilmaz.,provides a facility using GCM (Google Cloud Messaging) to send data from server to Android mobile device of user, and also receives messages from other devices within a network according to Android developer website, “a service that helps developers sends data from servers to their Android applications on Android devices.” Hence, the overall purpose and

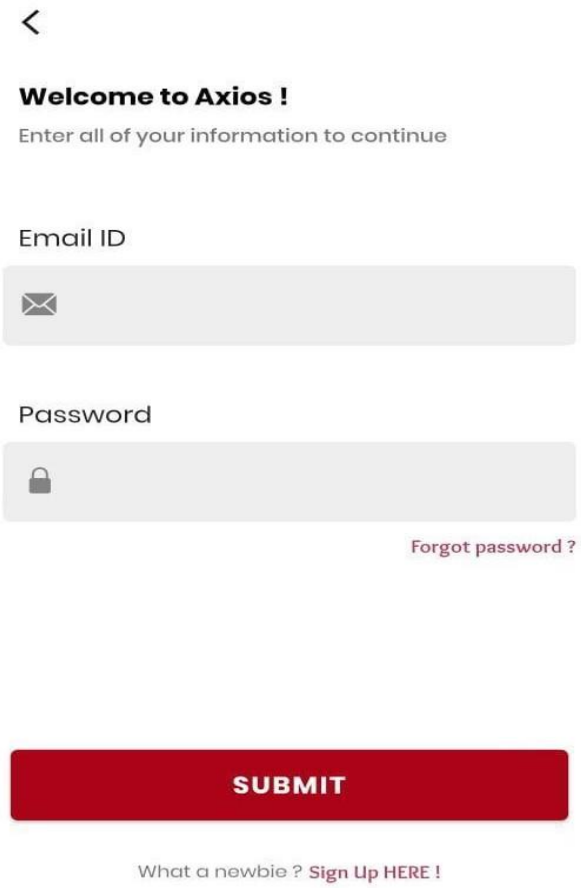
scope of this project is reachable by Push notification technology (GCM)

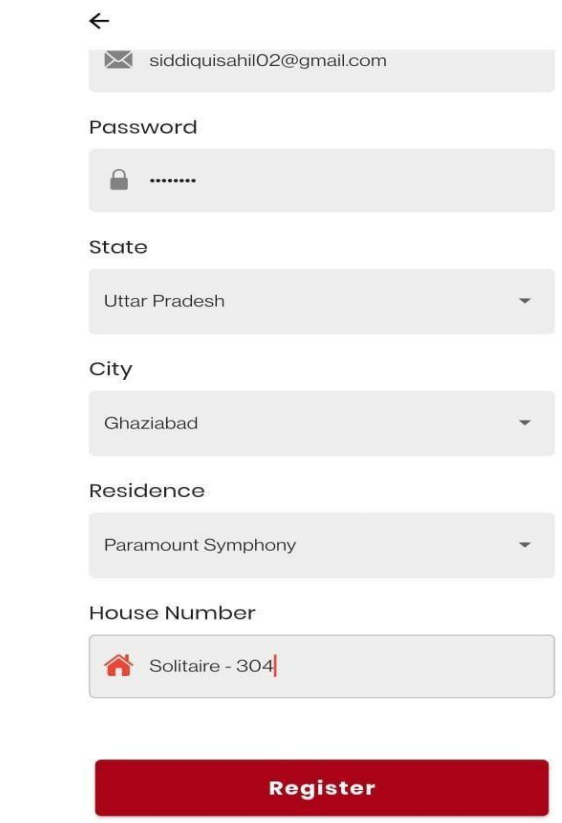
# CHAPTER 3

# DESIGNING

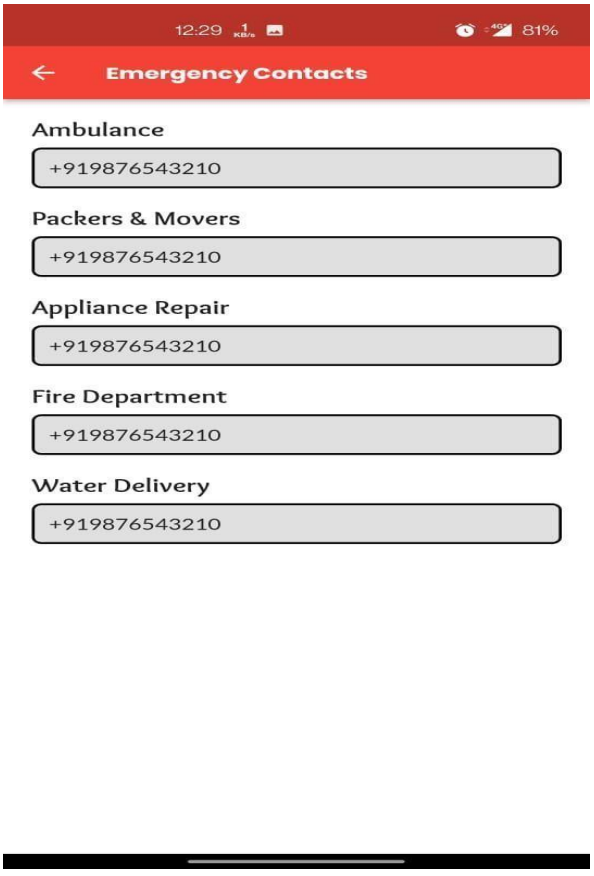
## 3.1 MODULES

### 3.1.1 Login Module

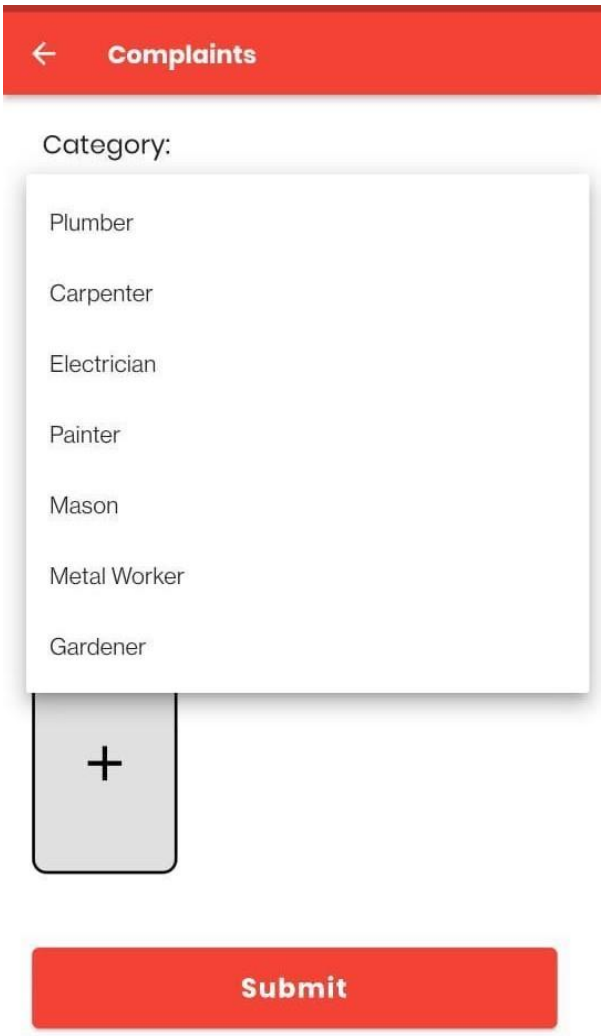





**3.2** Emergency Contacts



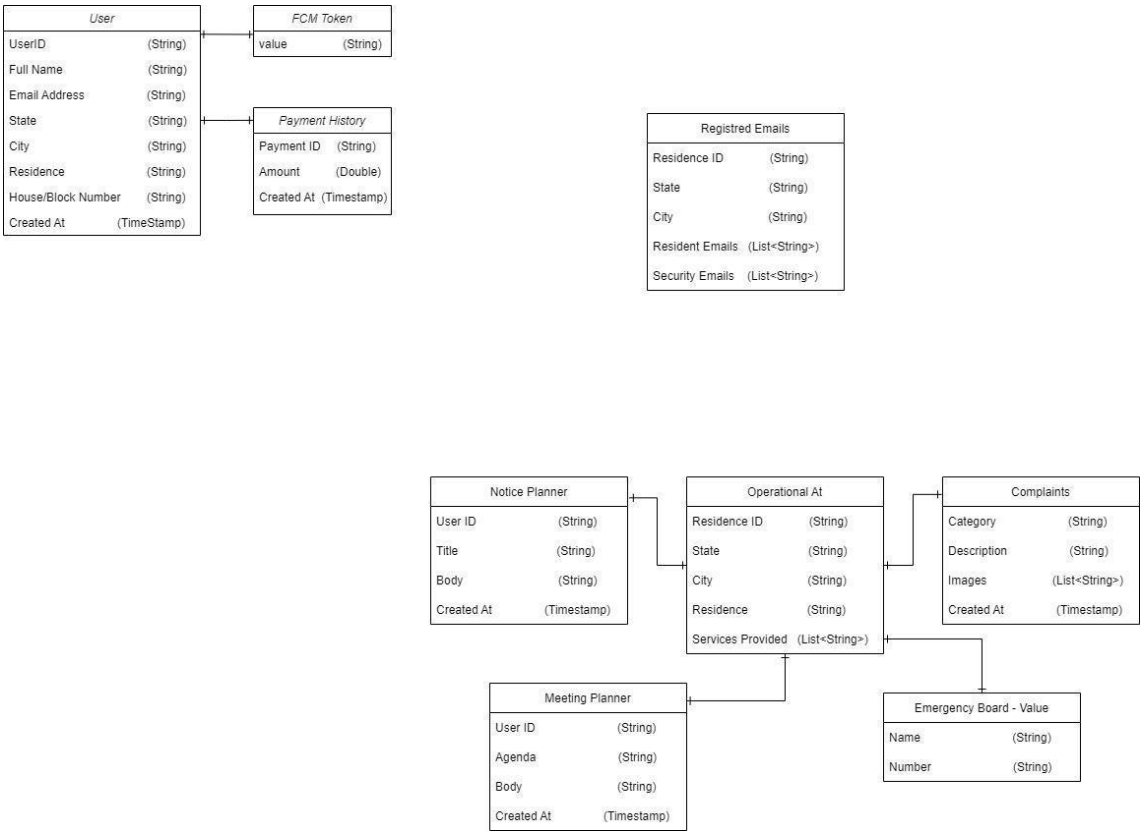

Notice Board
NIET Placement
29 June 2012
Contents. The target group for whom the notice is. Date, ...
Posted on: May 18, 2022 1:47 PM
Hello
This is to inform you that you
are a piece of shit
Posted on: May 18, 2022 1:01 PM


Complaints
Category:
Plumber
Carpenter
Electrician
Painter
Mason
Metal Worker
Gardener
+
Submit


Axios
Hello, App Scapper
appscapper@gmail.com (Resident)
Solitaire - 306, Paramount Symphony, Ghaziabad,
Uttar Pradesh
Your Dues
Rs. 345.43
Pay Now
Notice Board
Complaint Center
Meeting Board
Emergency Contacts


Meeting Planner
Regarding New Bill
Sunday
12.05.2022
Posted on: May 18, 2022 11:53 PM
Regarding New Bill
Sunday
12.05.2022
Flutter
Okay

## 3.2 DATA FLOW DIAGRAMS




### ER DIAGRAM

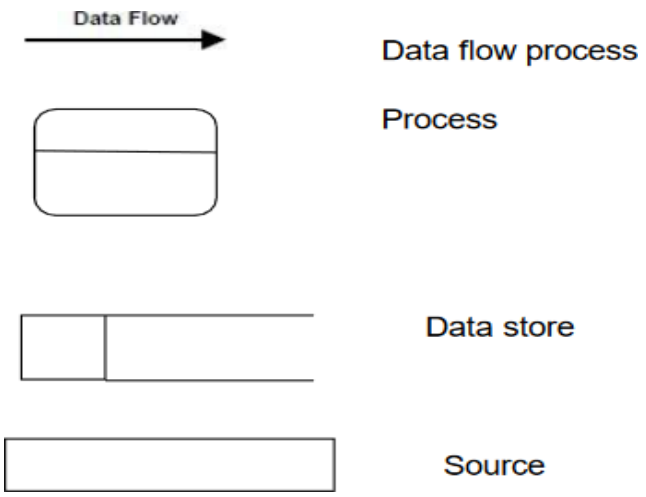
Data flow diagrams are the basic building blocks that define the flow of data in a system to the particular destination and difference in the flow when any transformation happens. It makes the whole procedure like a good document and makes it simpler and easier to understand for both programmers and non-programmers by dividing it into the sub-process.

The data flow diagrams are the simple blocks that reveal the relationship between various components of the system and provide high level overview, boundaries of particular systems as well as provide detailed overview of system elements.

The data flow diagrams start from source and end at the destination level i.e., it decomposes from high level to lower levels. The important things to remember about data flow diagrams are that it indicates the data flow for one way but not for loop structures and it doesn’t indicate the time factors.

This section reveals about the data flow analysis which states about data that have been used, classification of data flow diagrams based on their functions and the other different levels used in the project.

The general notations for constructing a block diagram in this project are:



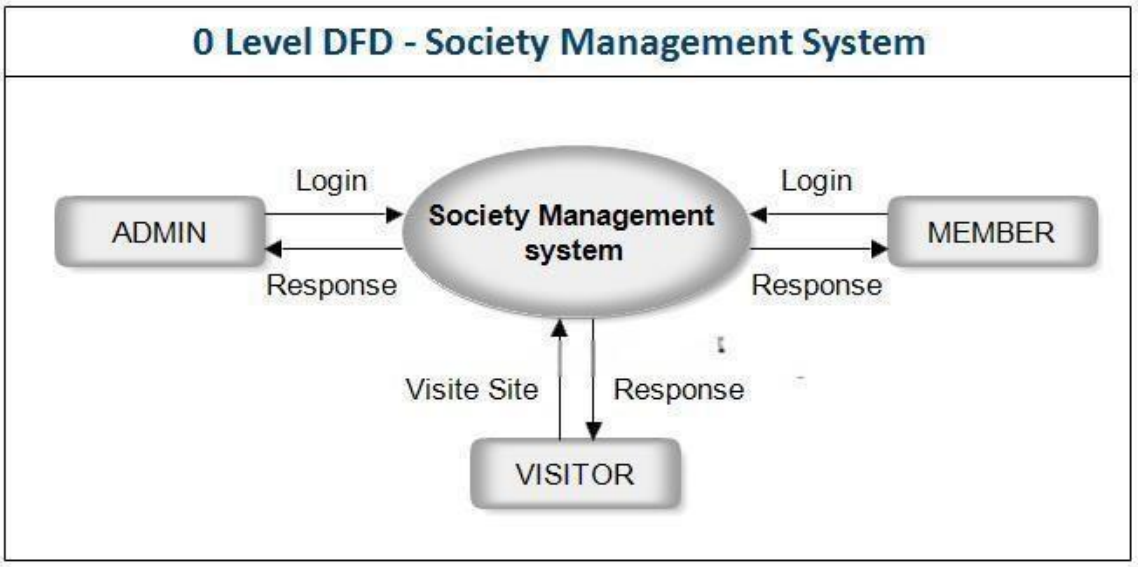

- Data flow processes: It will define the direction i.e., the data flow from one entity to another entity.

- Process: Process defines the source from where the output is generated for the specified input. It states the actions performed on data such that they are transformed, stored or distributed.

- Data store: It is the place or physical location where the data is stored after extraction from the data source.

- Source: It is the starting point or destination point of the data, starting point from where the external entity acts as a cause to flow the data towards

destination.



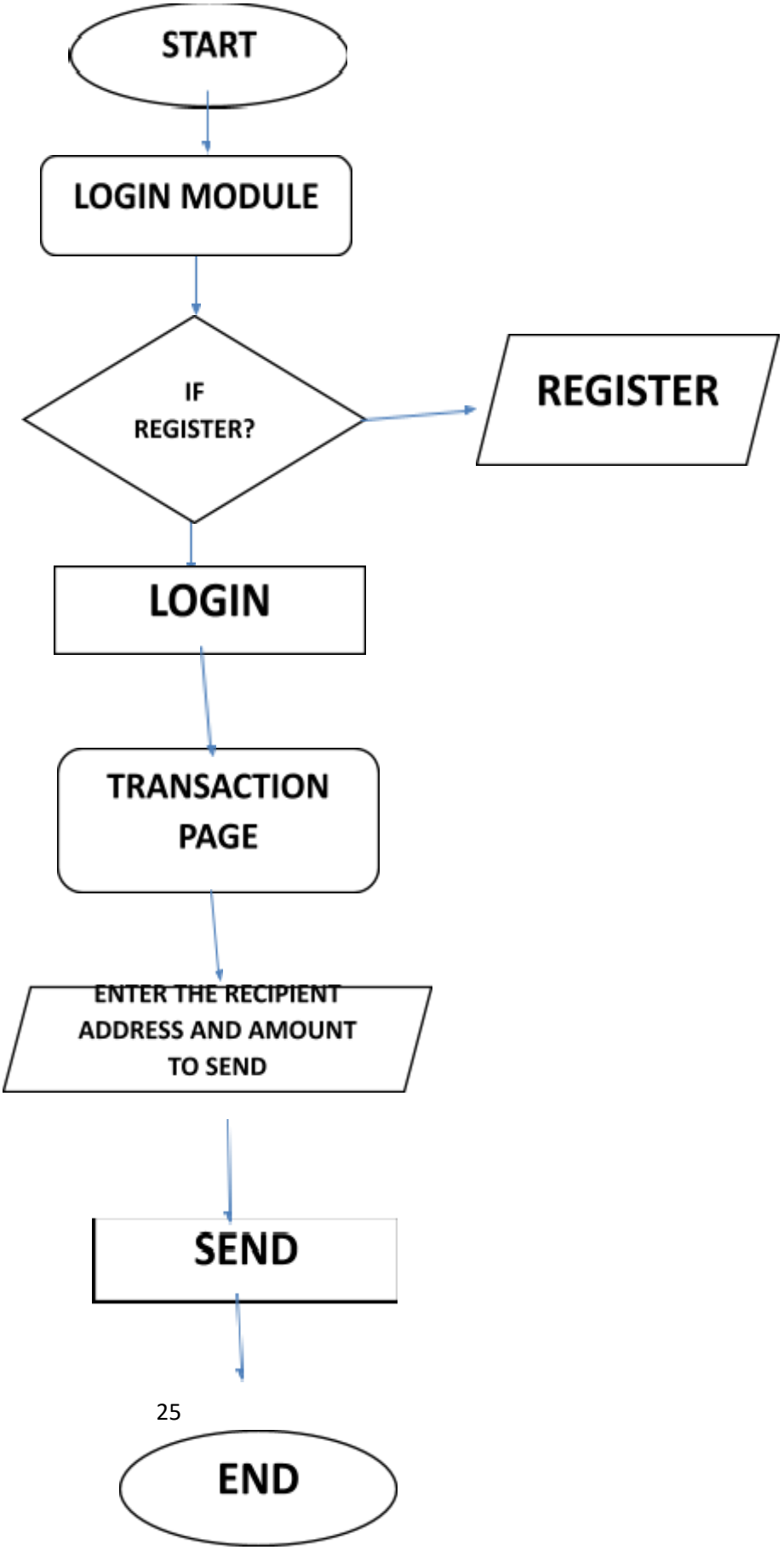

### 3.2.1 Constructing Data Flow Diagrams

The data flow diagrams can be constructed by dividing the process into different levels like DFD 0, DFD 1, DFD 2, etc., for constructing the data flow diagram. For this process, these simple steps are to be followed.

- The data flow diagram can be constructed only when the process have one data flow in and one data flow out.
- The process should modify the incoming data and outgoing data.
- The data store should not be alone, should be connected with one process at least.
- The external entities of the process should be involved with one data flow.
- In the data process the data flow should be from top to bottom and from left to right.
- In the data flow diagram, the data stores and their destinations are named with

capital letters and the data flow and process should be small capitalizing the Starting letter.

These rules should be followed for constructing the data flow diagrams.



# CHAPTER 4

# PROPOSED METHODOLOGY

The home page of the app will display the details for current visitors and services logged into the society using your permit, your current outstanding bill, complaint tracking and any unread notices. It will also allow user users to navigate to:

## 1. Visitor management section

This module will allow users to accept or deny the permission for entry of any visitor/vehicles in the premises. Whenever a visitor shows up at the society gates, the management will ask for the information from the guest (like where, purpose, how long etc.) and then ping a notification to the registered user of that estate to allow or deny their passage into the society.

## 2. Daily services management:

This module will allow users to see if their daily services personnel (maid, cook, driver, laundry etc.) have checked into the society or not. Once a personal check into the
society, he/she will be asked to show their ID badge/number to be marked “checked into the premise” for
that day and vice versa for check-outs. Respective users can also see this data.

## 3. Hire a Personnel:

This module will allow users to hire a service offered by a registered vendor (like carpenter, appliance repair, salon) on a one-time or monthly basis. Residents can also rate and leave comments on completion of the job too, which will be
visible to the other residents also.

## 4. Complaint Centre:

This module will allow users to send their complaints to the responsible department of the management and will keep track of it and users can also upload a picture/video of the issue with the complaint.

## 5. Digital Noticeboard:


This module will manage the task to deliver all the issued notices or announcements to the user via push notification and notice board section in the application in real time**.**

## 6. Payment Centre:

This module will allow users to pay their management, utilities and outstanding bills from the app using a payment gateway integrated in the application itself. Credit and debit cards will be accepted

## 7. Emergency Board:


This section will contain all the necessary nearest emergency phone numbers like Ambulance, Blood Bank, Fire Station, Police etc**.**

## 8. Associations Planner:

This section will help society members' associations to log, pre-plan

and keep track of their meetings. Valid Users can generate an event for monthly meetings of association and can add agendas and minutes of the last meetings.

## V. ADVANTAGES OF THE PROPOSED SYSTEM

This application offers an organized, managed and responsive platform for users to interact with their residential management. Using Flutter, we can easily deploy applications on Android and iOS platforms (which hold 72.84% and 26.34% of Mobile operating systems market share respectively) without writing separate code, any updates can be pushed using only one codebase and Flutter will compile for both platforms. Using Firebase cloud services as our backend will allow us to make our application scalable, flexible and reactive. Firebase offers Realtime cloud services such as Cloud Firestore, Authentication and Cloud messaging (Push Notifications). Our application will keep track of every entry and transaction for every resident and digitize the way management works in our society making it easier, robust and faster for our users and management, and will reduce the amount of paperwork done in our current system, ultimately reducing the use of paper. Our Visitor management system guarantees that no unauthorized
a person/vehicle will enter the residential premises making it a better and secure residential estate.

# CHAPTER 5

# FEASIBILITY STUDY

## 5.1 TECHNICAL FEASIBILITY

Axios – a society management mobile application is equipped with the following Hardware and Software requirements:

### 5.1.1 Hardware Requirements

- Phone

  Processor: Snapdragon 444 or Apple A6

  Disk space: Minimum 50 MB

  RAM: 2GB

- Internet

### 5.1.2 Software Requirements

- Operating System: Android, iOS
- Text Editor: Android Studio, Visual Studio Code
- Programming Language

  Login Module

  - Dart

  Transaction Module

- Dart
- JavaScript

- Other external development tools

  Firebase

## 5.2 FINANCIAL FEASIBILITY

Being a mobile application, Axios will have an associated deployment cost. Since the system doesn't consist of any multimedia data transfer, bandwidth required for the operation of this application is very low. The system will follow the freeware software standards. No cost will be charged from the potential customers. Bug fixes and maintaining tasks will have an associated cost. At the initial stage the potential market space will be the local universities and higher educational institutes. Beside the associated cost, there will be many benefits for the customers. Especially the extra effort that is associated with paper making and marking will be significantly reduced while the effort to create descriptive statistical reports will be eliminated, since reports generation is fully automated.

From these it's clear that Axios is financially feasible.

## 5.3 RESOURCE FEASIBILITY

Resources that are required for the Axios – society management mobile application project includes:

- Programming device (Laptop)
- Deployment space (Google Play store, Apple AppStore)
- Programming tools (freely available)
- Programming individuals

So, it's clear that the project Axios – society management mobile application has the required resource feasibility.

# CHAPTER 5

# RESULTS

Looking at the results as a whole, Flutter wins the majority of most categories in the development area. There are however some differences that are interesting to take note of when comparing Flutter to native builds.

Figure 1: Images of Native Android and Flutter Applications

| Type | lines of code | Code files that were needed for the application |
|---|---|---|
| Android native | 217 | 9 |
| Android Flutter | 125 | 3 |
| iOS native | 363 | 6 |
| iOS Flutter | 125 | 3 |

Figure 3: Application lines of code and file count

| Type | Total | Navigation Base | First View | Second View |
|---|---|---|---|---|
| Android native | 12h | 3h | 2h | 7h |
| iOS native | 8h | 2h | 0.5h | 5.5h |
| Flutter | 6h | 4h | 1h | 2h |

Figure 4: Development time of each code base

### 5.1 Code Size

As shown in Figure 4, Flutter requires the fewest lines of code and files to build your application. The native iOS project file size and app size were small, but there were significantly more lines of code than the other builds. Native Android created the most files and required less lines of code than native iOS.

## 5.2 Development Time

Android natives had the longest development time of 12 hours, followed by iOS natives with 8 hours, and finally Flutter with the shortest of 6 hours. If you are developing both native applications, you can take advantage of the use of drag and drop to accelerate your development. This is useful until you need to connect the deleted element to code This is more difficult than using the code directly to generate the layout. Layouts are always visible, even if you don't create them, which makes development easier. An equivalent development feature of Flutter is the hot reload feature. This allows you to build your application and reload it based on the newly added features.

## 5.3 Flutter run time CPU performance comparison

| Type | Language | Highest | Lowest | Mean | Standard Deviation |
|---|---|---|---|---|---|
| Native iOS | Swift | 92.7% | 14.3% | 32.9% | 13.75360872 |
| Flutter iOS | Dart | 101.7% | 18.8% | 35.3% | 18.00680891 |
| Native Android | Kotlin | 34.6% | 1.0% | 11.7% | 6.88638675 |
| Flutter Android | Dart | 32.3% | 1.0% | 13.2% | 9.29106696 |

Flutter iOS has higher top CPU performance than native iOS, and native Android has top CPU performance better than Flutter Android. But overall, Flutter has better average CPU performance on both OSs, outperforming native CPU performance on both iOS and Android.

### 5.4 Application Looks

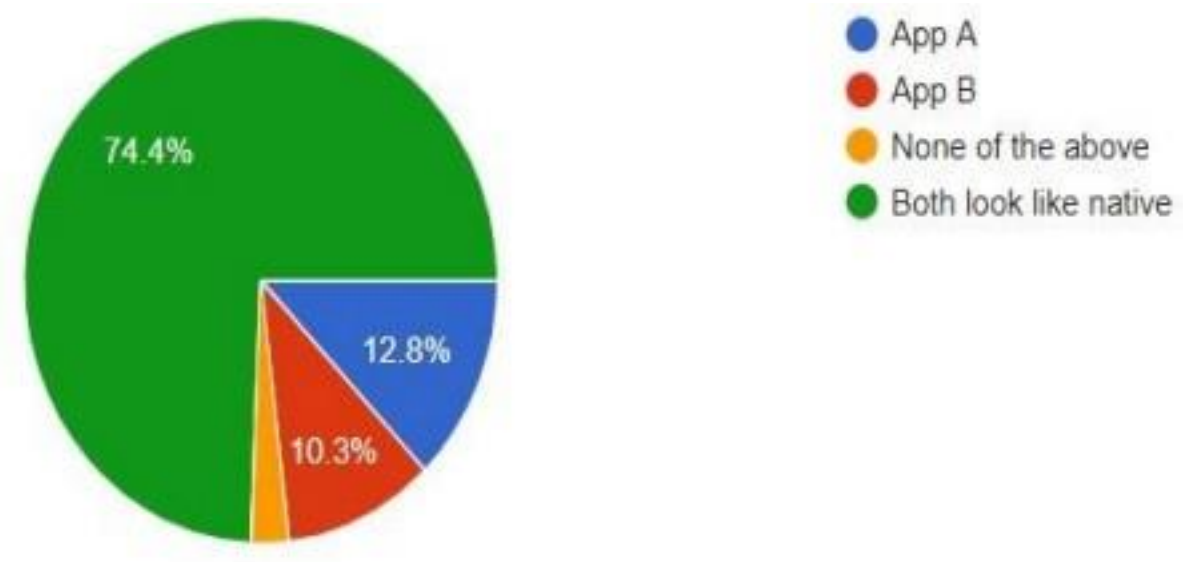
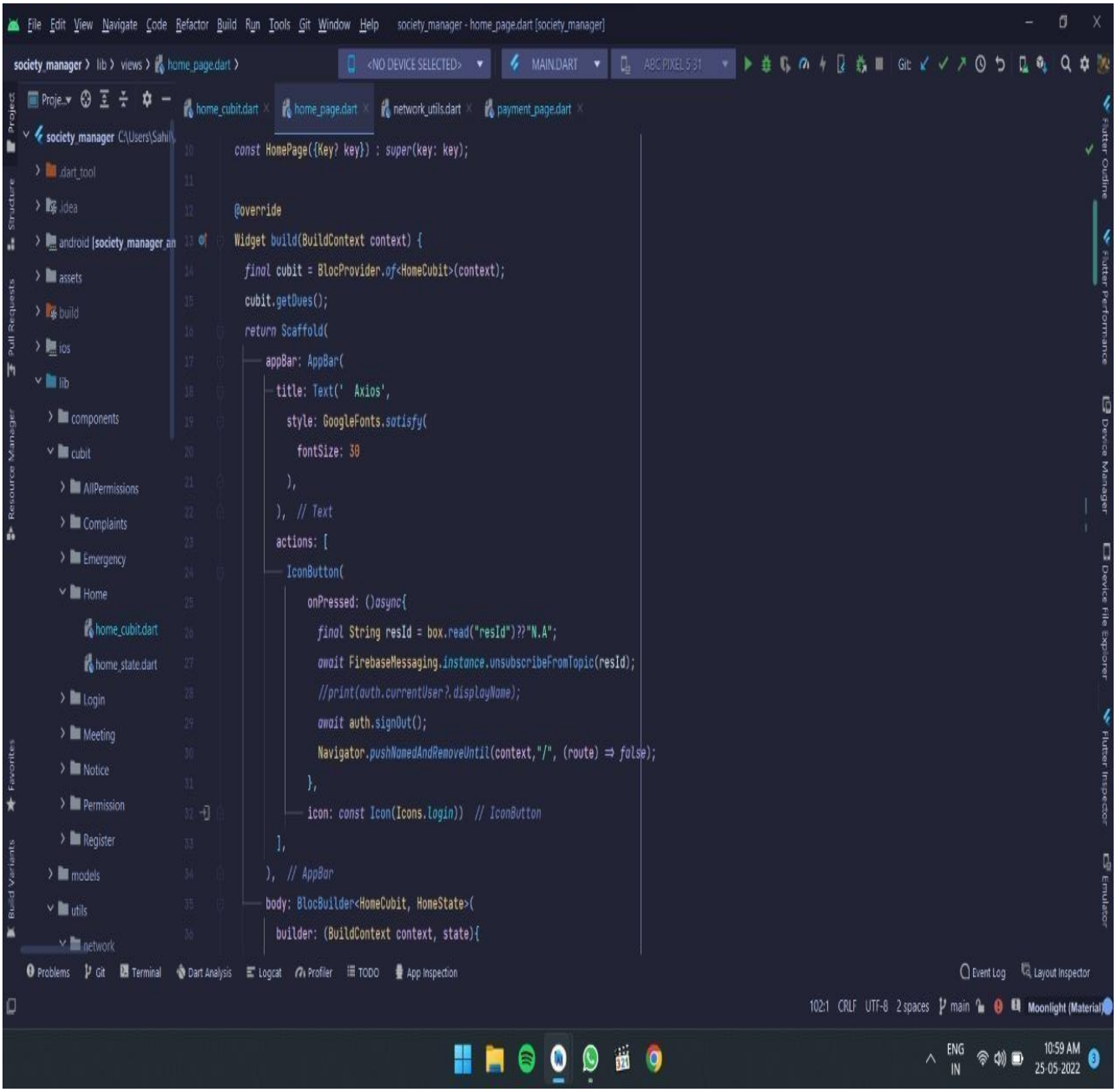
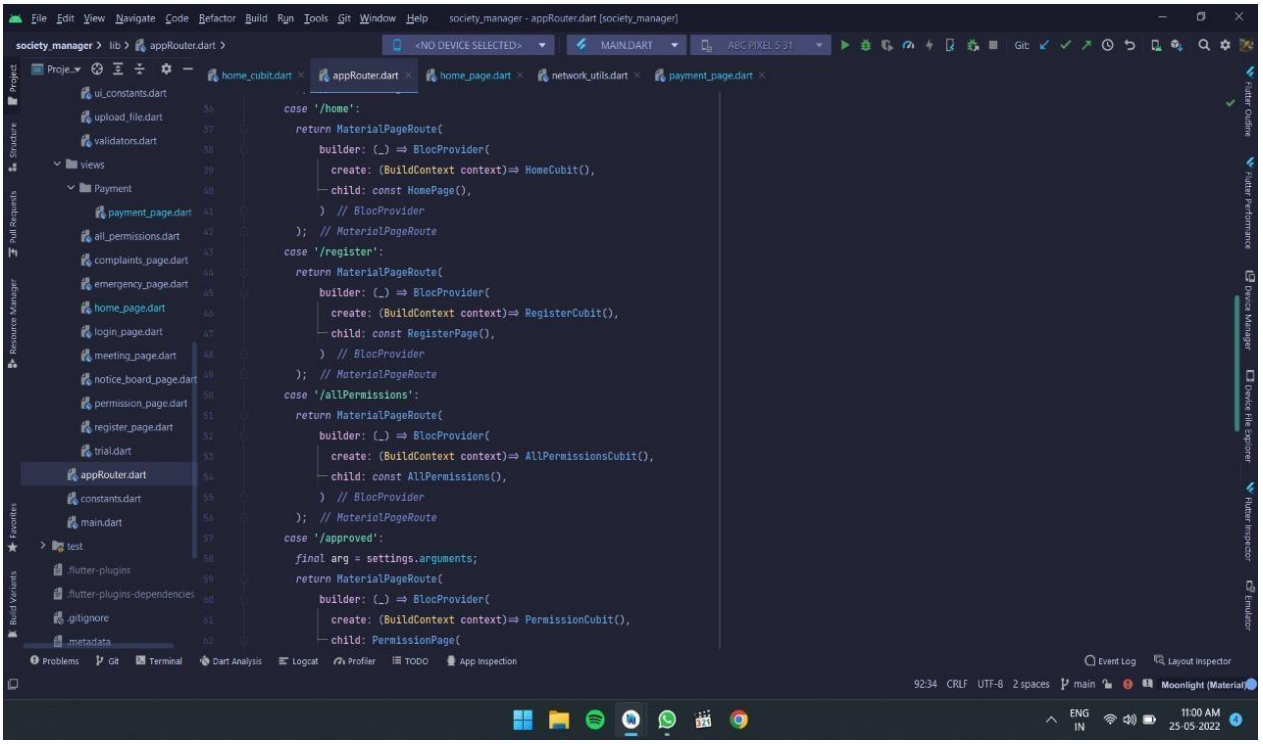


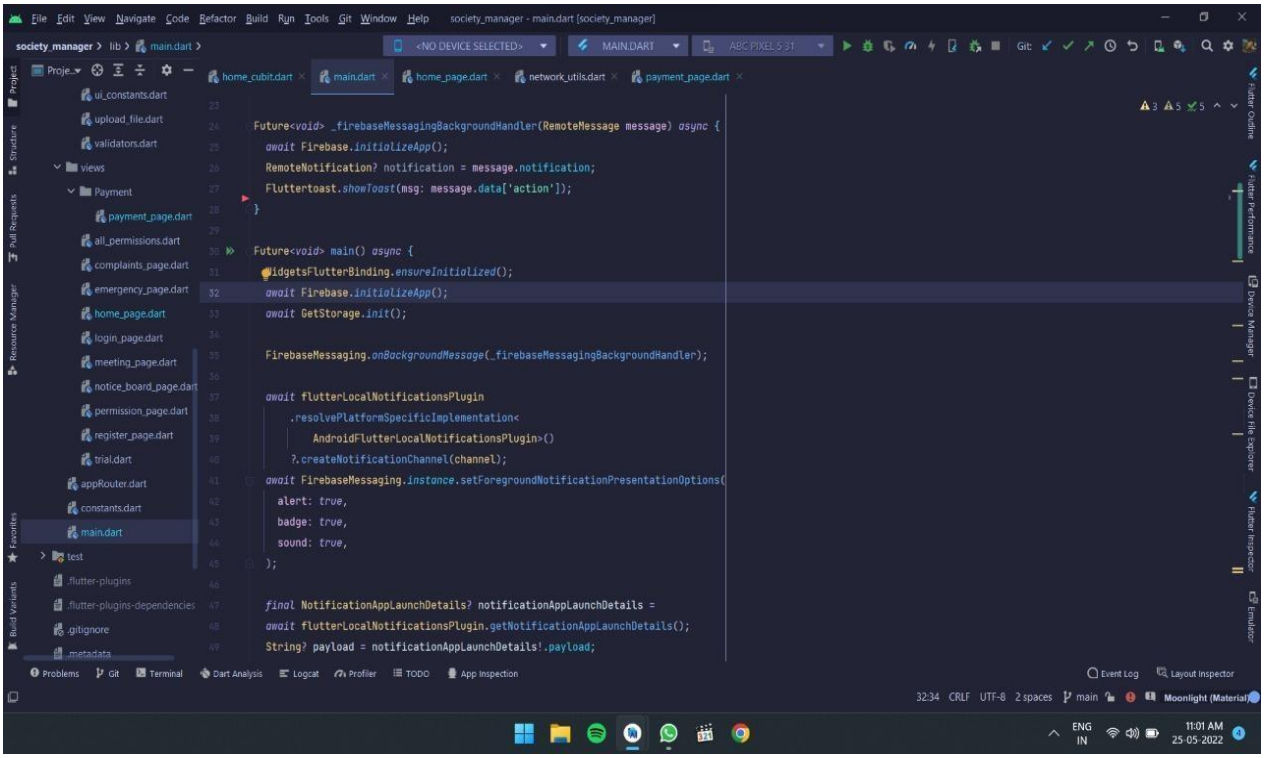

Diagram showing answers for looks of application Native (A) and Flutter (B)

This diagram is a survey conducted which clearly shows that most people believe that both the applications look the same.

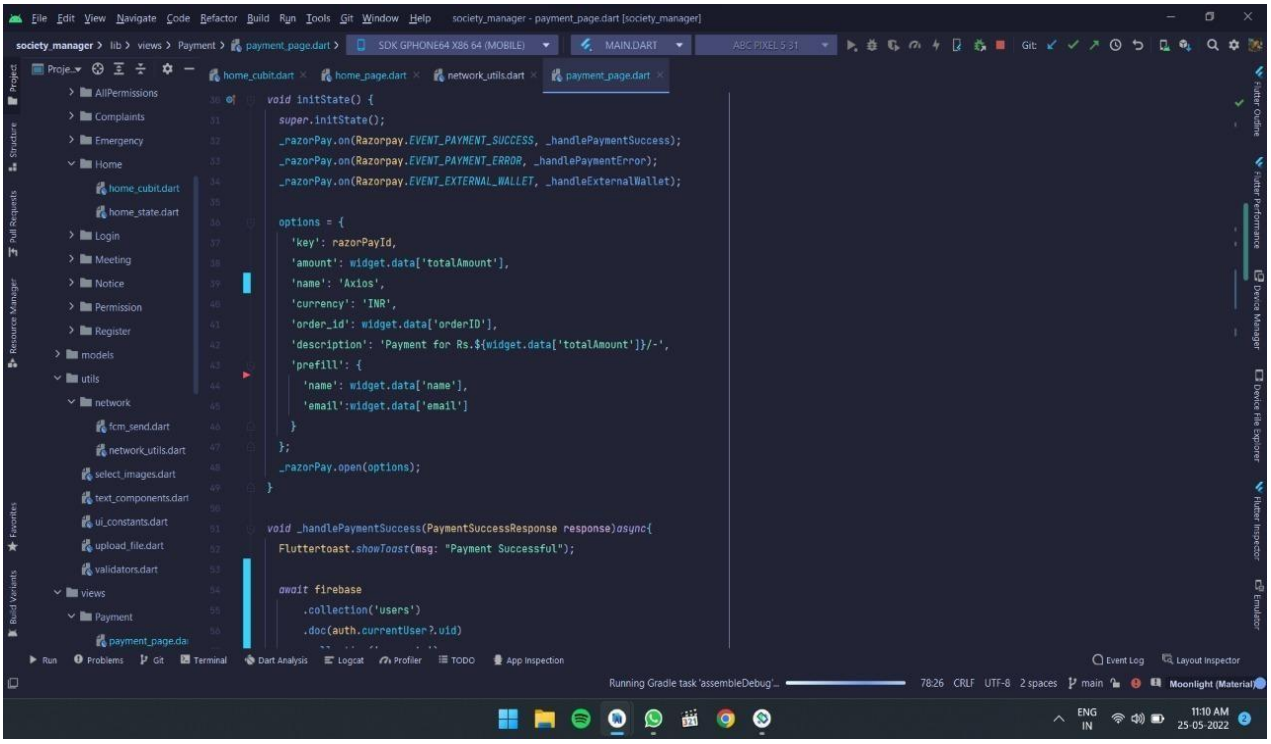
# Code Snippets



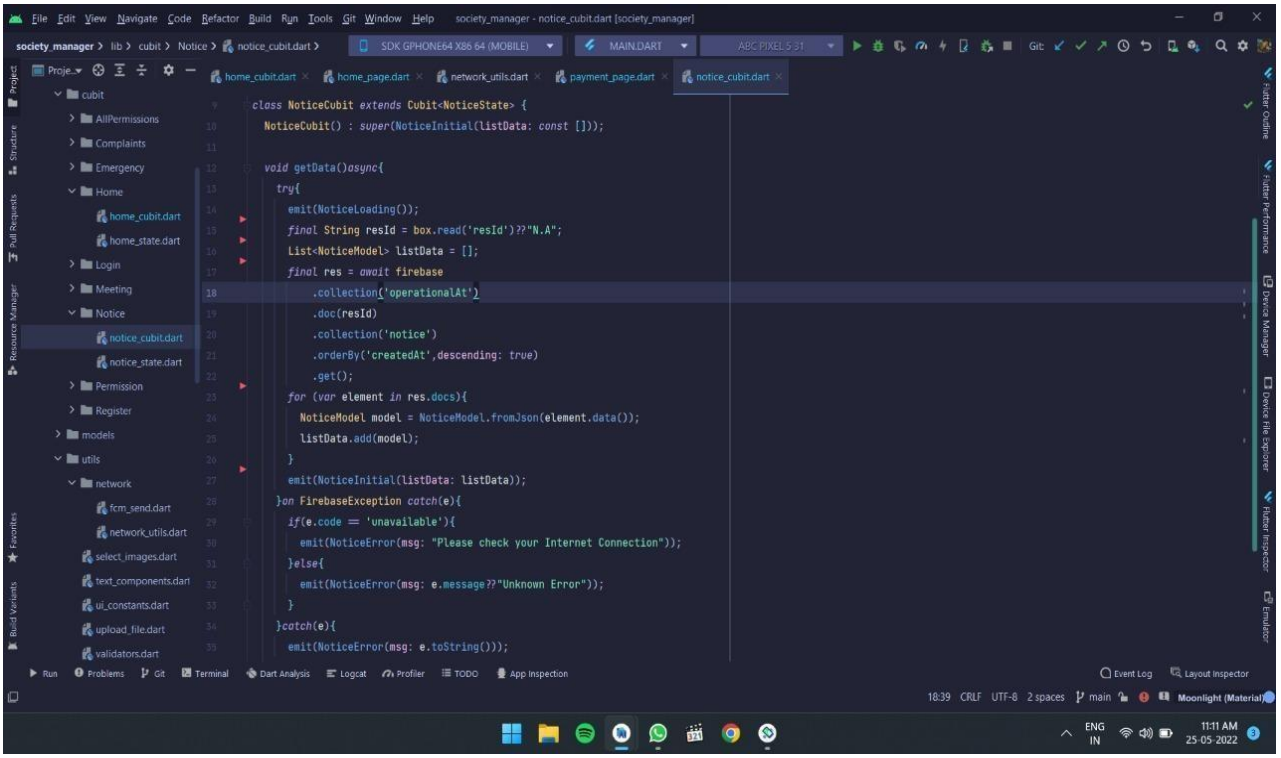

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
society_manager - appRouter.dart [society_manager]
society_manager > lib > appRouter.dart >
<NO DEVICE SELECTED>
MAIN.DART
ABC PIXEL 5 31
Git:
Proje...
ui_constants.dart
upload_file.dart
validators.dart
views
Payment
payment_page.dart
all_permissions.dart
complaints_page.dart
emergency_page.dart
home_page.dart
login_page.dart
meeting_page.dart
notice_board_page.dart
permission_page.dart
register_page.dart
trial.dart
appRouter.dart
constants.dart
main.dart
test
.flutter-plugins
.flutter-plugins-dependencies
.gitignore
.metadata
home_cubit.dart
appRouter.dart
home_page.dart
network_utils.dart
payment_page.dart
case '/home':
return MaterialPageRoute(
builder: (_) => BlocProvider(
create: (BuildContext context)=> HomeCubit(),
child: const HomePage(),
) // BlocProvider
); // MaterialPageRoute
case '/register':
return MaterialPageRoute(
builder: (_) => BlocProvider(
create: (BuildContext context)=> RegisterCubit(),
child: const RegisterPage(),
) // BlocProvider
); // MaterialPageRoute
case '/allPermissions':
return MaterialPageRoute(
builder: (_) => BlocProvider(
create: (BuildContext context)=> AllPermissionsCubit(),
child: const AllPermissions(),
) // BlocProvider
); // MaterialPageRoute
case '/approved':
final arg = settings.arguments;
return MaterialPageRoute(
builder: (_) => BlocProvider(
create: (BuildContext context)=> PermissionCubit(),
child: PermissionPage(
Problems
Git
Terminal
Dart Analysis
Logcat
Profiler
TODO
App Inspection
Event Log
Layout Inspector
92:34 CRLF UTF-8 2 spaces main
Moonlight (Material)
ENG
IN
11:00 AM
25-05-2022

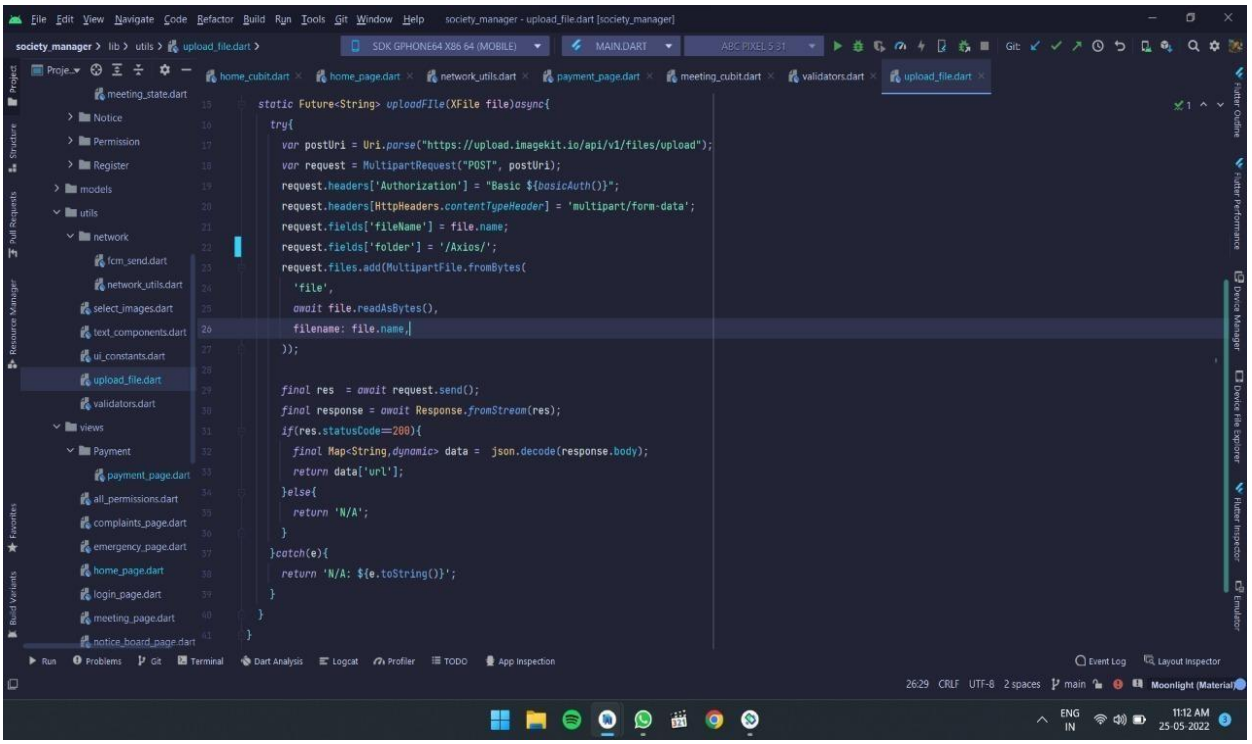


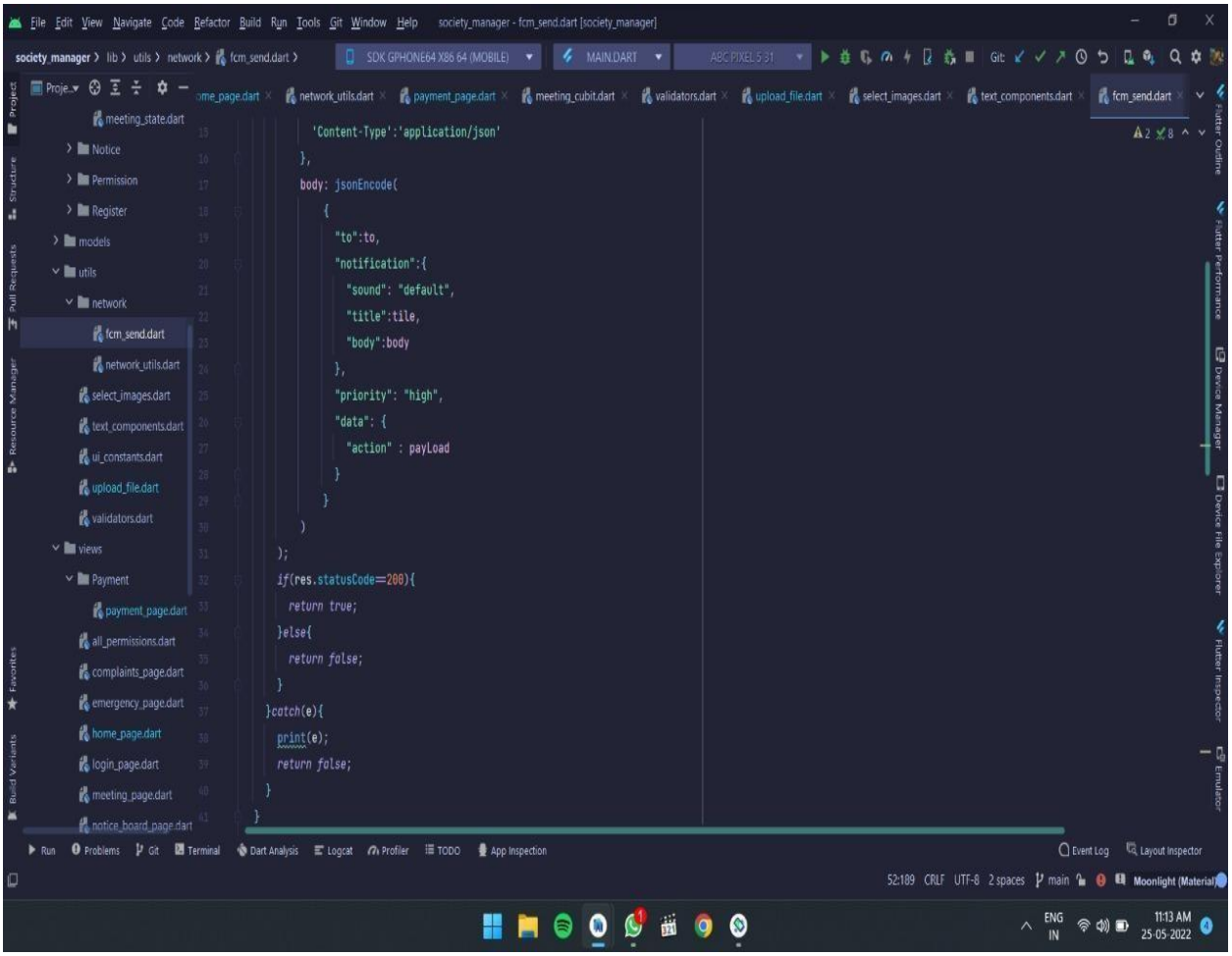
```
Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
  await Firebase.initializeApp();
  RemoteNotification? notification = message.notification;
  Fluttertoast.showToast(msg: message.data['action']);
}

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  await GetStorage.init();

  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);

  await flutterLocalNotificationsPlugin
      .resolvePlatformSpecificImplementation<
          AndroidFlutterLocalNotificationsPlugin>()
      ?.createNotificationChannel(channel);
  await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
    alert: true,
    badge: true,
    sound: true,
  );

  final NotificationAppLaunchDetails? notificationAppLaunchDetails =
  await flutterLocalNotificationsPlugin.getNotificationAppLaunchDetails();
  String? payload = notificationAppLaunchDetails!.payload;
```

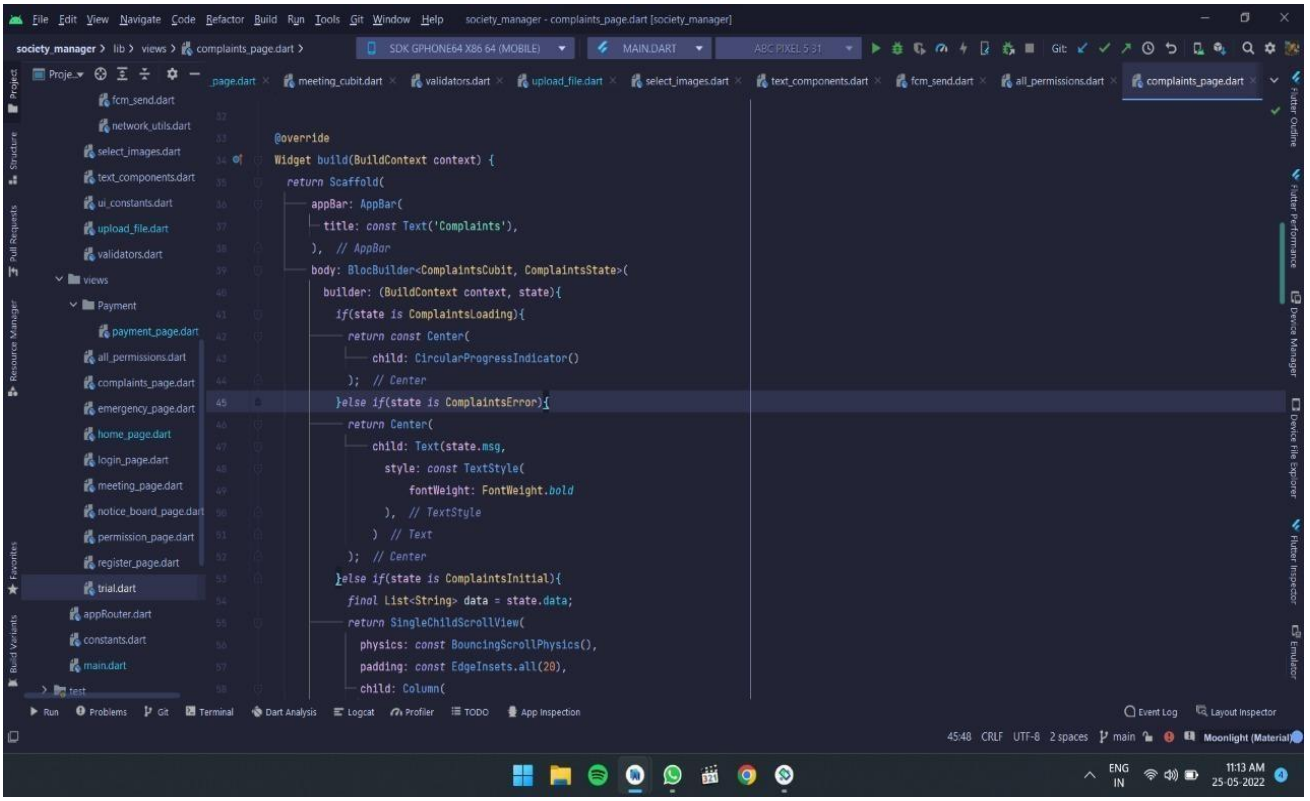


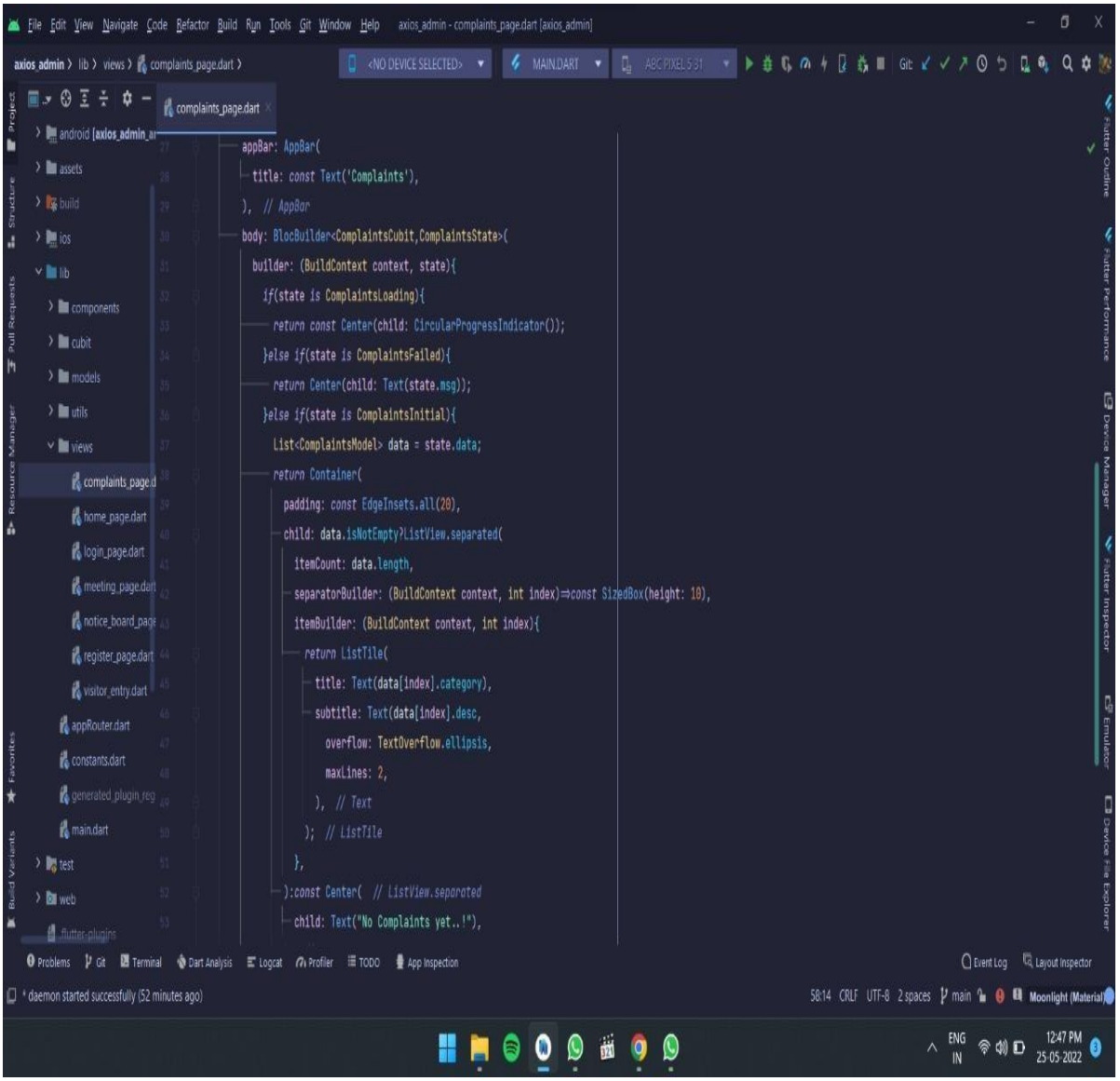
```
void initState() {
  super.initState();
  _razorPay.on(Razorpay.EVENT_PAYMENT_SUCCESS, _handlePaymentSuccess);
  _razorPay.on(Razorpay.EVENT_PAYMENT_ERROR, _handlePaymentError);
  _razorPay.on(Razorpay.EVENT_EXTERNAL_WALLET, _handleExternalWallet);

  options = {
    'key': razorPayId,
    'amount': widget.data['totalAmount'],
    'name': 'Axios',
    'currency': 'INR',
    'order_id': widget.data['orderID'],
    'description': 'Payment for Rs.${widget.data['totalAmount']}/-',
    'prefill': {
      'name': widget.data['name'],
      'email':widget.data['email']
    }
  };
  _razorPay.open(options);
}

void _handlePaymentSuccess(PaymentSuccessResponse response)async{
  Fluttertoast.showToast(msg: "Payment Successful");

  await firebase
      .collection('users')
      .doc(auth.currentUser?.uid)
```


society_manager - notice_cubit.dart [society_manager]
class NoticeCubit extends Cubit<NoticeState> {
NoticeCubit() : super(NoticeInitial(listData: const []));
void getData()async{
try{
emit(NoticeLoading());
final String resId = box.read('resId')??"N.A";
List<NoticeModel> listData = [];
final res = await firebase
.collection('operationalAt')
.doc(resId)
.collection('notice')
.orderBy('createdAt',descending: true)
.get();
for (var element in res.docs){
NoticeModel model = NoticeModel.fromJson(element.data());
listData.add(model);
}
emit(NoticeInitial(listData: listData));
}on FirebaseException catch(e){
if(e.code == 'unavailable'){
emit(NoticeError(msg: "Please check your Internet Connection"));
}else{
emit(NoticeError(msg: e.message??"Unknown Error"));
}
}catch(e){
emit(NoticeError(msg: e.toString()));


society_manager - upload_file.dart [society_manager]
static Future<String> uploadFile(XFile file)async{
try{
var postUri = Uri.parse("https://upload.imagekit.io/api/v1/files/upload");
var request = MultipartRequest("POST", postUri);
request.headers['Authorization'] = "Basic ${basicAuth()}";
request.headers[HttpHeaders.contentTypeHeader] = 'multipart/form-data';
request.fields['fileName'] = file.name;
request.fields['folder'] = '/Axios/';
request.files.add(MultipartFile.fromBytes(
'file',
await file.readAsBytes(),
filename: file.name,
));
final res = await request.send();
final response = await Response.fromStream(res);
if(res.statusCode==200){
final Map<String,dynamic> data = json.decode(response.body);
return data['url'];
}else{
return 'N/A';
}
}catch(e){
return 'N/A: ${e.toString()}';
}
}
}


File Edit View Navigate Code Refactor Build Run Tools Git Window Help society_manager - fcm_send.dart [society_manager]
society_manager > lib > utils > network > fcm_send.dart >
SDK GPHONE64 X86 64 (MOBILE)
MAIN.DART
Git:
meeting_state.dart
Notice
Permission
Register
models
utils
network
fcm_send.dart
network_utils.dart
select_images.dart
text_components.dart
ui_constants.dart
upload_file.dart
validators.dart
views
Payment
payment_page.dart
all_permissions.dart
complaints_page.dart
emergency_page.dart
home_page.dart
login_page.dart
meeting_page.dart
notice_board_page.dart
network_utils.dart
payment_page.dart
meeting_cubit.dart
validators.dart
upload_file.dart
select_images.dart
text_components.dart
fcm_send.dart
'Content-Type':'application/json'
},
body: jsonEncode(
{
"to":to,
"notification":{
"sound": "default",
"title":tile,
"body":body
},
"priority": "high",
"data": {
"action" : payLoad
}
}
)
);
if(res.statusCode=200){
return true;
}else{
return false;
}
}catch(e){
print(e);
return false;
}
}
Run
Problems
Git
Terminal
Dart Analysis
Logcat
Profiler
TODO
App Inspection
Event Log
Layout Inspector
52:189 CRLF UTF-8 2 spaces main
Moonlight (Material)
ENG
IN
11:13 AM
25-05-2022


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
society_manager - complaints_page.dart [society_manager]
society_manager > lib > views > complaints_page.dart >
SDK GPHONE64 X86 64 (MOBILE)
MAIN.DART
Git:
fcm_send.dart
network_utils.dart
select_images.dart
text_components.dart
ui_constants.dart
upload_file.dart
validators.dart
views
Payment
payment_page.dart
all_permissions.dart
complaints_page.dart
emergency_page.dart
home_page.dart
login_page.dart
meeting_page.dart
notice_board_page.dart
permission_page.dart
register_page.dart
trial.dart
appRouter.dart
constants.dart
main.dart
test
meeting_cubit.dart
validators.dart
upload_file.dart
select_images.dart
text_components.dart
fcm_send.dart
all_permissions.dart
complaints_page.dart
@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(
title: const Text('Complaints'),
), // AppBar
body: BlocBuilder<ComplaintsCubit, ComplaintsState>(
builder: (BuildContext context, state){
if(state is ComplaintsLoading){
return const Center(
child: CircularProgressIndicator()
); // Center
}else if(state is ComplaintsError){
return Center(
child: Text(state.msg,
style: const TextStyle(
fontWeight: FontWeight.bold
), // TextStyle
) // Text
); // Center
}else if(state is ComplaintsInitial){
final List<String> data = state.data;
return SingleChildScrollView(
physics: const BouncingScrollPhysics(),
padding: const EdgeInsets.all(20),
child: Column(
Run
Problems
Git
Terminal
Dart Analysis
Logcat
Profiler
TODO
App Inspection
Event Log
Layout Inspector
45:48
CRLF
UTF-8
2 spaces
main
Moonlight (Material)
ENG
IN
11:13 AM
25-05-2022


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
axios_admin - complaints_page.dart [axios_admin]
axios_admin > lib > views > complaints_page.dart
<NO DEVICE SELECTED>
MAIN.DART
ABC PIXEL 5 31
Git:
complaints_page.dart
android [axios_admin_a
assets
build
ios
lib
components
cubit
models
utils
views
complaints_page.d
home_page.dart
login_page.dart
meeting_page.dart
notice_board_page
register_page.dart
visitor_entry.dart
appRouter.dart
constants.dart
generated_plugin_reg
main.dart
test
web
flutter_plugins
appBar: AppBar(
title: const Text('Complaints'),
), // AppBar
body: BlocBuilder<ComplaintsCubit,ComplaintsState>(
builder: (BuildContext context, state){
if(state is ComplaintsLoading){
return const Center(child: CircularProgressIndicator());
}else if(state is ComplaintsFailed){
return Center(child: Text(state.msg));
}else if(state is ComplaintsInitial){
List<ComplaintsModel> data = state.data;
return Container(
padding: const EdgeInsets.all(20),
child: data.isNotEmpty?ListView.separated(
itemCount: data.length,
separatorBuilder: (BuildContext context, int index)=>const SizedBox(height: 10),
itemBuilder: (BuildContext context, int index){
return ListTile(
title: Text(data[index].category),
subtitle: Text(data[index].desc,
overflow: TextOverflow.ellipsis,
maxLines: 2,
), // Text
); // ListTile
},
):const Center( // ListView.separated
child: Text("No Complaints yet..!"),
Problems
Git
Terminal
Dart Analysis
Logcat
Profiler
TODO
App Inspection
Event Log
Layout Inspector
daemon started successfully (52 minutes ago)
58:14 CRLF UTF-8 2 spaces main
Moonlight (Material)
ENG
IN
12:47 PM
25-05-2022

# CHAPTER 6

# CONCLUSION

We proposed an Android application for housing society management to maintain transparency between society members and management and to minimize the paperwork load. It also helps to reduce the efforts for manually communicating with each society member for providing notifications and important information.

Flutter is a useful tool kit that enables easy ways of creating new applications. It has gotten more and more popular

recently. The basic results in this report indicates flutter has a slight edge as compared to native application development platforms but further more conclusive tests still need to be carried out to come to a final conclusion. Appearance wise, Flutter and native seem to differentiate little to a majority of users. It is able to mimic the native looks to a certain point.

To conclude the answers and ideas of Flutter, it is a tool with a promising feature if the community continues to grow in the direction that it is right now. The line to be drawn when choosing Flutter over two separate native builds, can be chosen at the development of smaller to medium applications which are more flexible. Considering that Flutter's strong side is being a cross platform solution, Flutter still performs well on

a single application base if compared to native applications. Flutter may not beat native for developing applications at this point but the results show good potential for the future although further studies need to be done in these areas to conclude safer answers.

# CHAPTER 7

# FUTURE SCOPE

This project can be further improved by:

- Making the application available in multiple regional languages.
- Allowing in-app internet calling facility.
- Public chatroom for discussing/interacting with other residents and event planning.
- Society members would be able to interact like on WhatsApp groups for better user experience as a future scope to work upon.
- Releasing a separate web application for management

# CHAPTER 8

# REFERENCES

Following reference has been used to develop this project:

**Web Source: -**


1. Madhuram, Ashu Kumar and Pandyamanian “Cross Platform Development using Flutter”, IJESC Research article, volume 20, Issue No. 4, 2019.
2. (2021) Flutter website. [Online]. Available: https://docs.flutter.dev/
3. (2021) Firebase website. [Online]. Available: https://firebase.google.com/
4. Linda Rosencrance. (2019) TechTarget webpage on Google Firebase. [Online]. Available:
5. https://searchmobilecomputing.techtarget.com/definition/ Google-Firebase