ZiGreenCity

A Project Report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

MASTER OF COMPUTER APPLICATION

by

Priyam Srivastava

University Roll No. 1900290149073

Under the Supervision of

Mr. Ankit Verma

Assistant Professor

KIET Group of Institutions, Ghaziabad

Submitted to

DEPARTMENT OF COMPUTER APPLICATION

Affiliated to

Dr. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

July, 2021

DECLARATION

I hereby declare that the work presented in this report entitled **"ZiGreenCity** ", was carried out by me. I have not submitted the matter embodied in thisreport for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences andgiven credit to the original authors/sources.

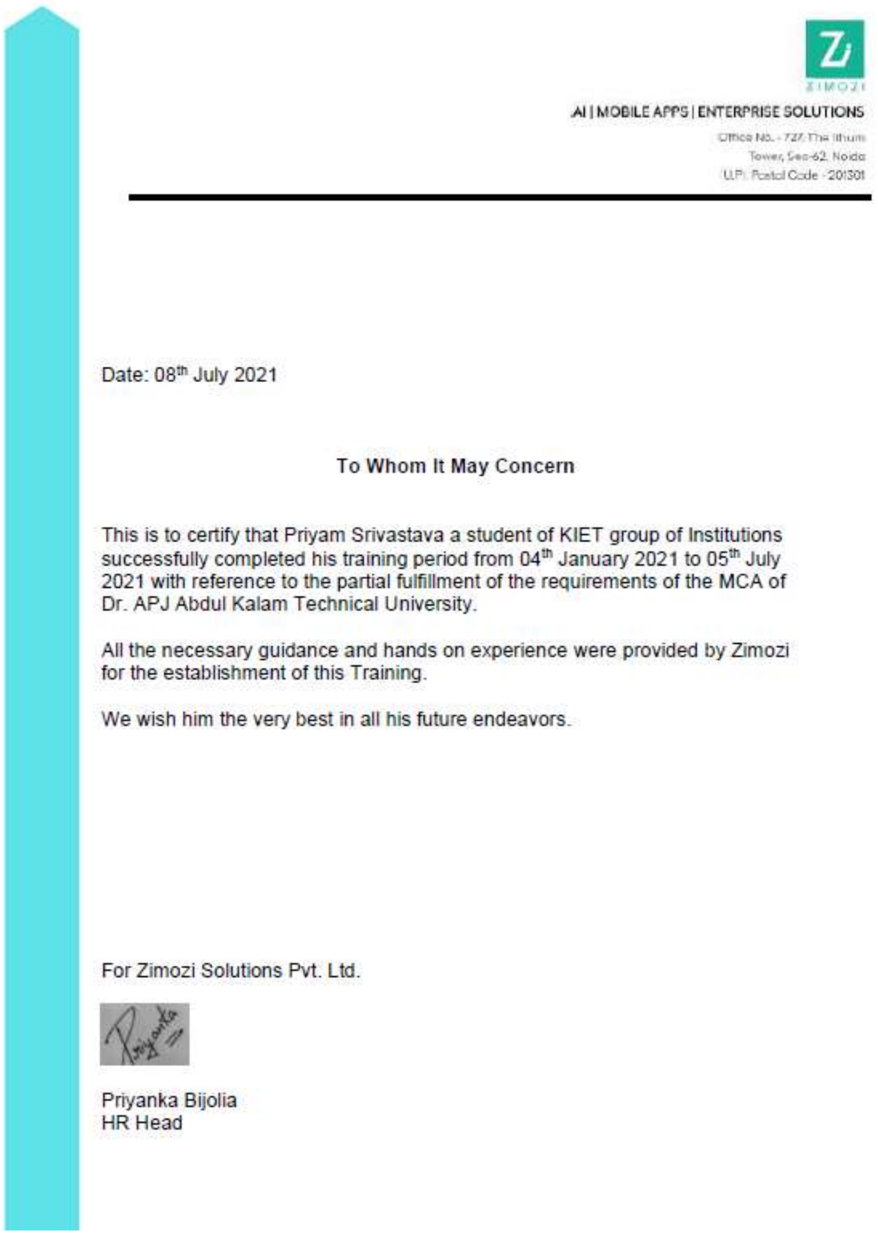
I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarismand the manipulation of the experiments and results, I shall be fully responsible and answerable.

[illegible]

Name: Priyam Srivastava

University Roll No. 1900290149073

TRAINING CERTIFICATE



ZIMOZI

AI | MOBILE APPS | ENTERPRISE SOLUTIONS

Office No. - 727 The Ithum
Tower, Sec-62, Noida
U.P. Postal Code - 201301

Date: 08th July 2021

To Whom It May Concern

This is to certify that Priyam Srivastava a student of KIET group of Institutions successfully completed his training period from 04th January 2021 to 05th July 2021 with reference to the partial fulfillment of the requirements of the MCA of Dr. APJ Abdul Kalam Technical University.

All the necessary guidance and hands on experience were provided by Zimozi for the establishment of this Training.

We wish him the very best in all his future endeavors.

For Zimozi Solutions Pvt. Ltd.

Priyanka

Priyanka Bijolia
HR Head

CERTIFICATE

Certified that **Priyam Srivastava (1900290149073)** have carried out the project work having "**ZiGreenCity**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody elsefrom this or any other University/Institution.

Date:

Priyam Srivastava (Univ. Roll No – 1900290149073)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Mr. Ankit Verma
Assistant Professor
Department of Computer Application
KIET Group of Institutions, Ghaziabad

Signature of External Examiner **Signature of Internal Examiner**

Dr. Ajay Kumar Shrivastava
Head, Department of Computer Application
KIET Group of Institutions, Ghaziabad

ABSTRACT

In today’s smart world every one of us using smartphone no matter of which platform (Android, iOS, Windows). Each platform has got many applications which made it popular in present generation. This project is developed to make earth cleaner and greener.

Anyone can use this app and contribute in cleaning earth, ZiGreenCity app is an non profitable app that hopes to creates a groundswell of people, all across the world, engaged with the issue of waste management.

Simply people need to start their journey and try to pick waste material from their environment and marks each item in an app which is collected, and they will get reward batches for that, it’s great initiative to make our environment clean and green.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Ankit Verma,** for his guidance, help and encouragementthroughout my research work.

Words are not enough to express my gratitude to **Dr. Ajay Kumar Shrivastava**, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my lifefilled with enjoyment and happiness.

Priyam Srivastava

Univ. Roll No. 1900290149073

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT TITLE

- ZiGreenCity
- An application used to make our earth greener and cleaner.
- Provide proper functionality in very efficient manner so that user can easily use this application.
- This project is part of my training at my organisation.

1.2 PROJECT DESCRIPTION

A cleaner way to move. Come onboard a global community engaged with making our world a cleaner place! “Unlitter” your environments and track the impact you’ve made with the ZiGreenCity app. Not only are you able to log the difference you’ve made as an individual, the app connects you to Striders all around the globe, so you can see how we’re progressing together as a community. Striders enjoy a cleaner way to move – it’s fun, simple, and good for the environment. So, what are you waiting for? Join the movement today!

ZiGreenCity is a non-profit app that hopes to create a groundswell of people, all across the world, engaged with the issue of waste management.

1.3 PROJECT SCOPE

Unlittering made fun with an intuitive log system, Start and end your unlittering journeys with a simple tap. Logging can be done in seconds with our seamless logging system, beginning with easy tagging from a range of waste categories that fit your finds, to snapping photographs of wacky litter that you’ve come across.

Achievements up for grabs, there are a bunch of titles and achievements for our Striders to work towards. Elevate your profile with cool new titles like “Strider Supreme” or “Can Crusher” by completing achievement prerequisites. Work towards being the most decked out Strider in the community!

Track (y)our impact, every logged Stride goes into a global database that keeps track of the community’s unlittering efforts. Track how much you’ve done for the environment with details such as number and type of pick-ups logged, distance covered and time spent Striding, Stride routes, and even photos taken! These details are also available for local and global communities, so you can stay motivated from seeing what the rest of the community is doing – you’ll never feel like you’re doing this alone with ZiGreenCity.

1.4 HARDWARE/SOFTWARE USED IN PROJECT

Hardware and software are two necessities for development on this tool. The hardware and the software used in the application are as follows:

- Microsoft Windows 10
- Android Studio
- Minimum 30 GB hard disk Storage Space
- Laptop Machine
- Google Firebase

1.5 PROJECT SCHEDULE (GANTT CHART)

The estimated period for this project was 6 months. The work was divided into several phases as shown in the Gantt chart. Firstly, we research and got familiarized with the tool which we used during later phases. The design and planning of the software and hardware part followed. After that we start coding for this project and then testing and debugging were performed along the way. Hence, complete the project in time as expected.

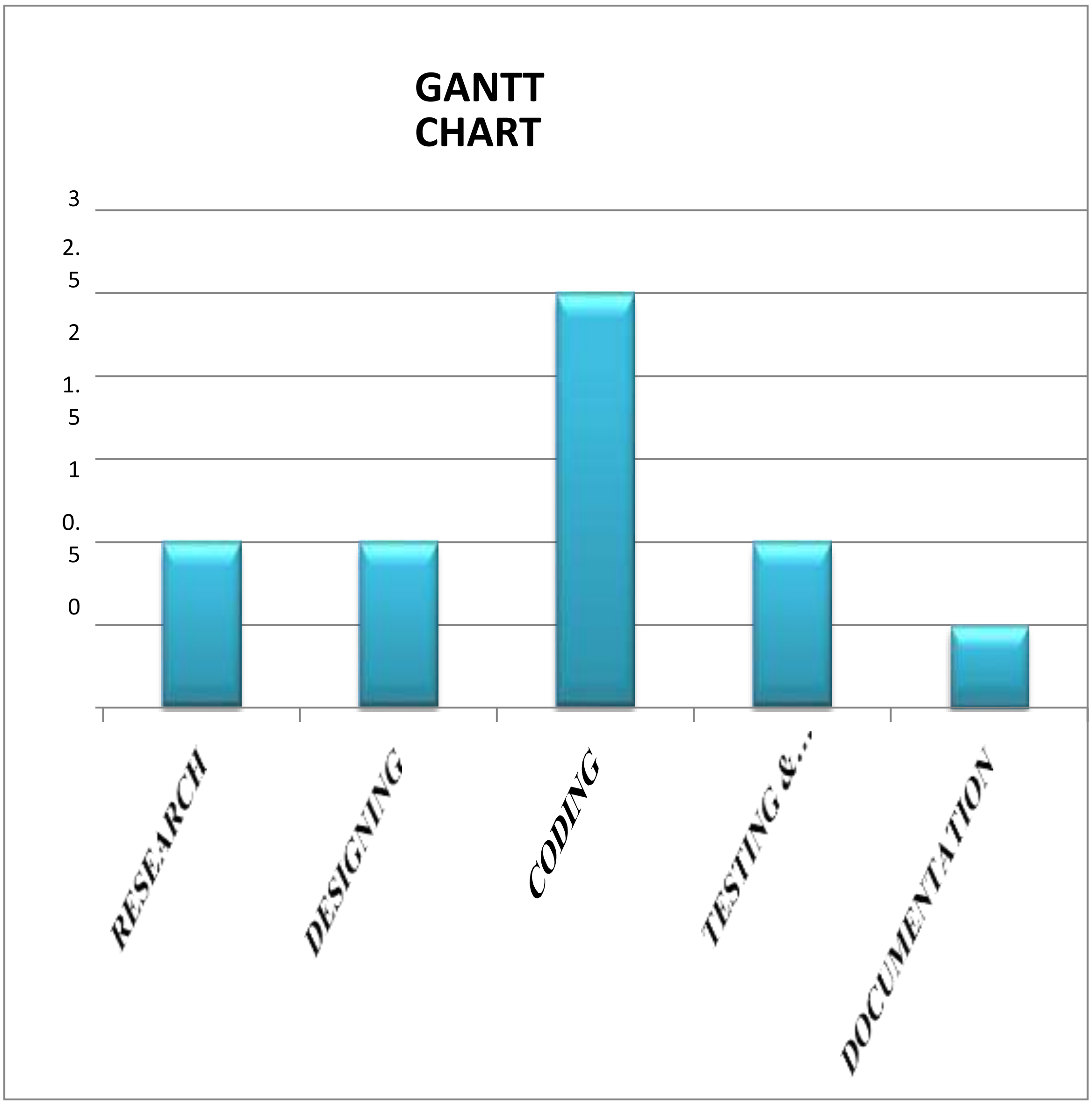


Figure 1.1 Gantt Chart

CHAPTER 2
LITERATURE REVIEW

2.1 WHAT IS ANDROID

Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by a startup of the same name, Android, Inc. In 2005, as part of its strategy to enter the mobile space, Google purchased Android, Inc. and took over its development work (as well as its development team).

Google wanted the Android OS to be open and free, so most of the Android code was released under the open-source Apache License. That means anyone who wants to use Android can do so by downloading the full Android source code. Moreover, vendors (typically hardware manufacturers) can add their own proprietary extensions to Android and customize Android to differentiate their products from others. This development model makes Android very attractive to vendors, especially those companies affected by phenomenon of Apple's iPhone, which was a hugely successful product that revolutionized the smartphone industry. When the iPhone was launched, many smartphone manufacturers had to scramble to find new ways of revitalizing their products. These manufacturers saw Android as a solution, meaning they will continue to design their own hardware and Android as the operating system that powers it. Some companies that have taken advantage of Android's open-source policy include Motorola and Sony Ericsson, which have been developing their own mobile operating systems for many years.

The main advantage to adopting Android is that offers as unified approach to application development. Developers need only develop for Android in general, and their applications should be able to run on numerous different devices, as long as the device are powered using Android. In the word of smartphones, applications are the most important part of the success chain.[5]

2.2 FEATURES OF ANDROID

Because Android is open source and freely available to manufacturers for customization, there are no fixed hardware or software configurations. However, the base Android OS supports many features, including

- **Storage** - SQLite, a lightweight relational database, for data storage. Chapter discusses data storage in more detail
- **Connectivity** - GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth (includes A2DP and AVRCP), Wi-Fi, LTE and WIMAX
- **Media support** - H.263, H.264 (in 3GP or MP4 container), MPEG-4 SP, AMR, AMR-WB (in 3GP container), AAC, HE-AAC (in or 3GP container), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- **Hardware support** - Accelerometer sensor, camera, digital compass, proximity sensor, and GPS.
- **Multi-touch** - multi-touch screens.
- **Multi-tasking** - multi-tasking applications.
- **Tethering** - Sharing of Internet connections as a wired/wireless hotspot

Android web browser is based on the open source WebKit and Chrome’s V8 JavaScript engine.[5]

2.3 ARCHITECTURE OF ANDROID

The Android OS is roughly divided into five sections in four main layers

- **Linux kernel -** This is the kernel on which Android is used. This layer contains all the low-level device drivers for the various hardware components of an Android device.
- **Libraries -** These contain the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The WebKit library provides functionalities for web browsing

- **Android runtime -** The Android runtime is located in the same layer with the libraries and provides a set of core libraries that enable developers to write Android apps using the Java programming language. The Android runtime also includes the Dalvik virtual machine, which enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. (Android applications are compiled into Dalvik executables). Dalvik ka specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU power.
- **Application framework** -The application framework exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications.
- **Applications** - At this top layer are the applications that ship with the Android device (such as Phone, Contacts, Browser, and so on), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer [5]

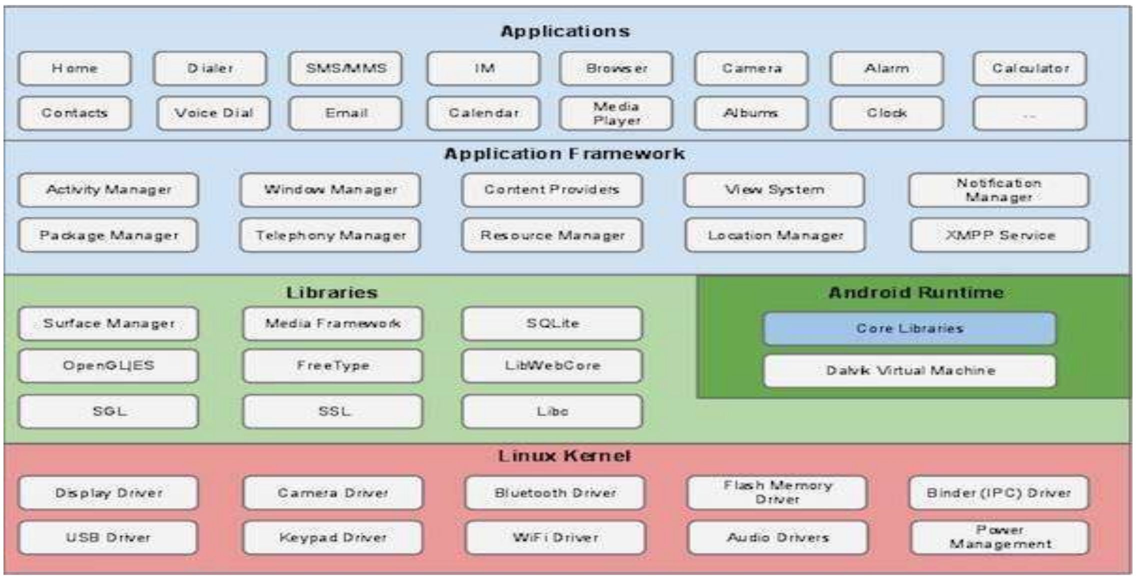


Figure 2.1 Android Architecture

2.4 THE ANDROID MARKET

As mentioned earlier, one of the main factors determining the success of a smartphone platform is the applications that support it. It is clear from the success of the Phone that application play a very vital role in determining whether a new platform swims or sink. Also, making these applications accessible to the general user is extremely important.

Users can simply use the Google Play application devices to directly download third-party applications to their devices. Both paid and free applications are available in the Google Play Store, although paid applications are available only to users in certain countries because of legal issues.[5]

2.5 OBTAINING THE REQUIRED TOOLS

Now that you know what Android is and what its feature set contains, you are probably anxious to get your hands dirty and start writing some applications! Before you write your first app, however, you need to download the required tools.

For Android development, you can use a Mac, a Windows PC, or a Linux machine. You can freely download all the necessary tools. Most of the examples provided in this book are written to work on Android Studio. For this book, I am using a Windows 10 computer to demonstrate all the code samples. If you are using a Mac or Linux computer, the screenshots should look similar. Some minor differences might be present, but you should be able to follow along without problems.[5]

2.5.1 ANDROID STUDIO

The first and most important piece of software you need to download is Android Studio 2. After you have downloaded and installed Android Studio 2, you can use the SDK Manager to download and install multiple versions of the Android SDK. Having multiple versions of the SDK available enables you to write programs that target different devices. For example, you can write one version of an application that specifically targets Android Nougat, but because that flavor of Android is on less than 1% of devices, with multiple versions of the SDK you can also write a version of your app that uses older features and targets Marshmallow or Lollipop users. You can use the Android Device Manager to set up device emulators.[5]

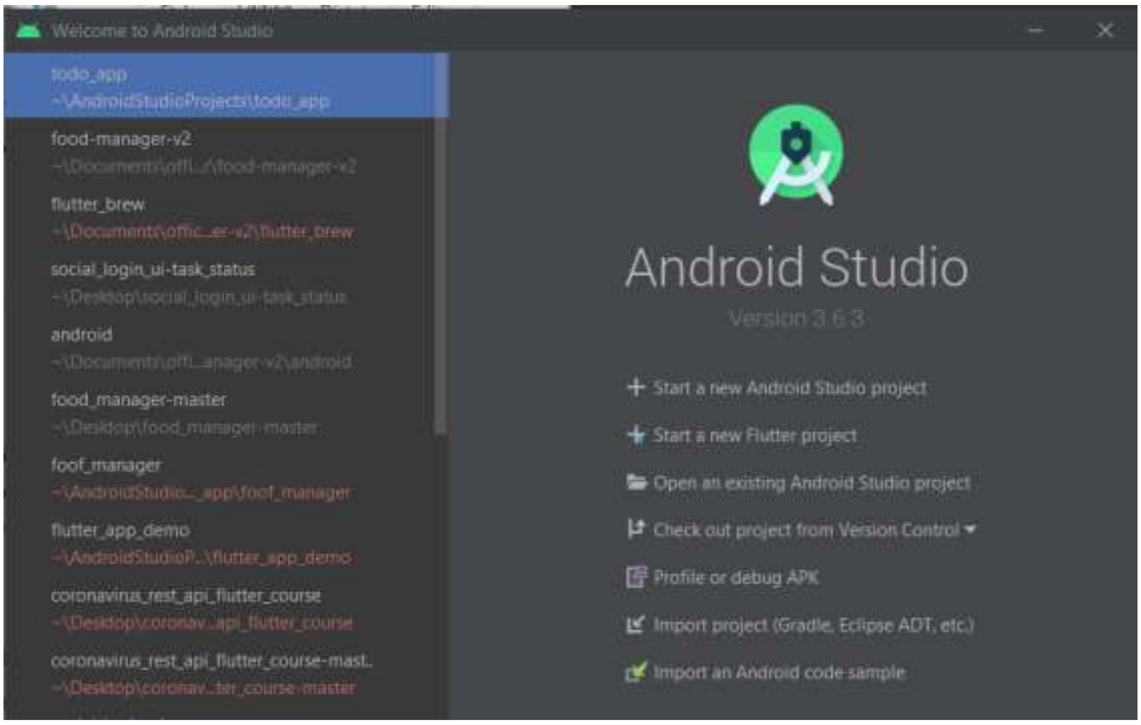

Welcome to Android Studio
todo_app
~\AndroidStudioProjects\todo_app
food-manager-v2
~\Documents\offi...r\food-manager-v2
flutter_brew
~\Documents\offic...er-v2\flutter_brew
social_login_ui-task_status
~\Desktop\social_login_ui-task_status
android
~\Documents\offi...anager-v2\android
food_manager-master
~\Desktop\food_manager-master
foof_manager
~\AndroidStudio..._app\foof_manager
flutter_app_demo
~\AndroidStudioP...\flutter_app_demo
coronavirus_rest_api_flutter_course
~\Desktop\coronav...api_flutter_course
coronavirus_rest_api_flutter_course-mast...
~\Desktop\coronav...ter_course-master
Android Studio
Version 3.6.3
Start a new Android Studio project
Start a new Flutter project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample

Figure 2.2 Android Studio

2.5.2 CREATING ANDROID VIRTUAL DEVICES (AVDs)

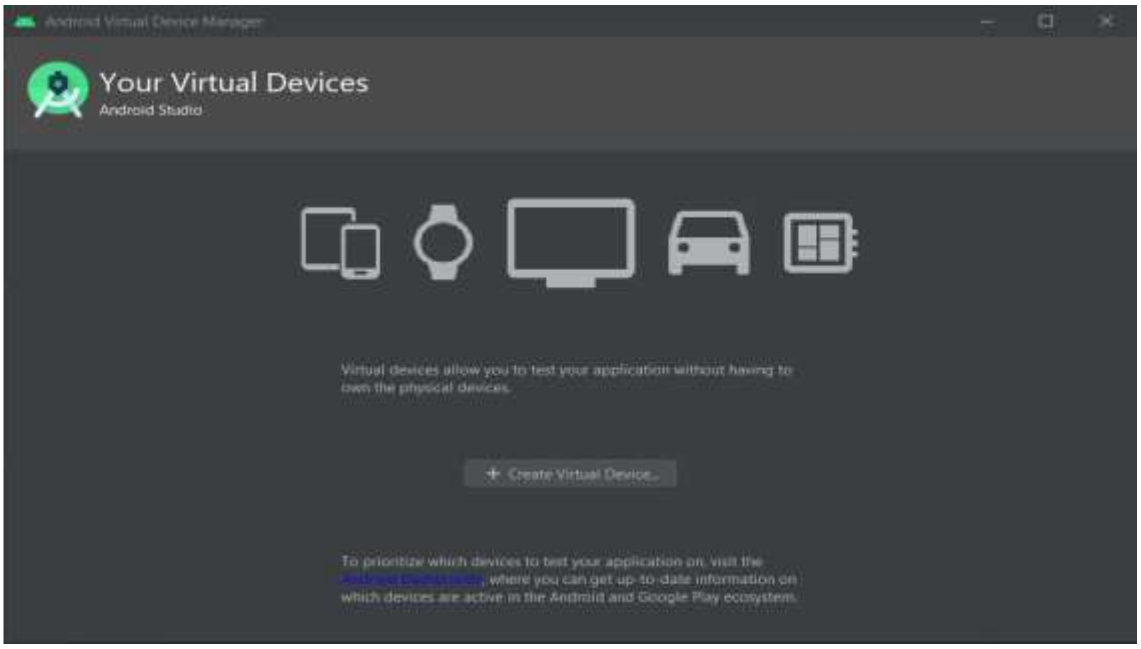
The next step is to create an Android Virtual Device (AVD) you can use for testing your Android applications. An AVD is an emulator instance that enables you to model an actual device. Each AVD consists of a hardware profile; a mapping to a system image; and emulated storage, such as a secure digital (SD) card. One important thing to remember about emulators is that they are not perfect. There are some applications, such as games (which are GPU heayy) or applications that use sensors such as the GPS or accelerometer. These types of applications cannot be simulated with the same speed or consistency within an emulator as they can when running on an actual device. However, the emulator is good for doing some generalized testing of your applications.

You can create as many AVDs as you want to test your applications with different configurations. This testing is important to confirm the behavior of your application when it is run on different devices with varying capabilities.

Use the following steps to create an AVD. This example demonstrates creating an AVD (put simply, an Android emulator) that emulates an Android device running Android N on the Nexus 5x hardware specs.

Step 1

Start AVD vizard to create a virtual device.



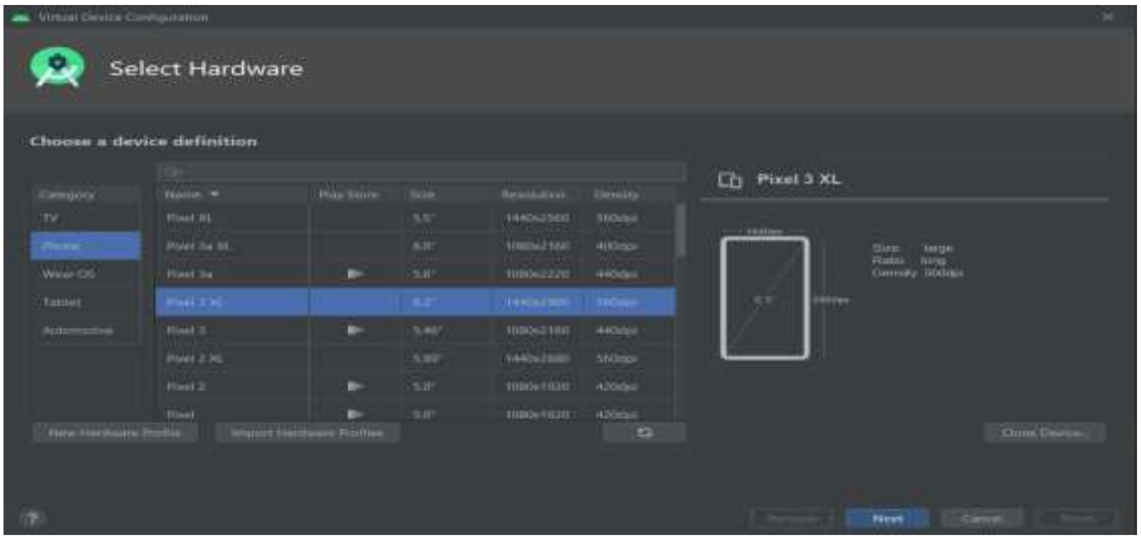
Virtual devices allow you to test your application without having to own the physical devices.

To prioritize which devices to test your application on, visit the Android Dashboards, where you can get up-to-date information on which devices are active in the Android and Google Play ecosystem.

Figure 2.3 Creating Virtual device

Step 2

Select device of your choice to make device. There will be many options, you can select according to your requirements



Category
TV
Phone
Wear OS
Tablet
Automotive

Name ▾	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2560	560dpi
Pixel 3a XL		6.0"	1080x2160	400dpi
Pixel 3a	▶	5.6"	1080x2220	440dpi
Pixel 3 XL		6.3"	1440x2960	560dpi
Pixel 3	▶	5.46"	1080x2160	440dpi
Pixel 2 XL		5.99"	1440x2880	560dpi
Pixel 2	▶	5.0"	1080x1920	420dpi
Pixel	▶	5.0"	1080x1920	420dpi

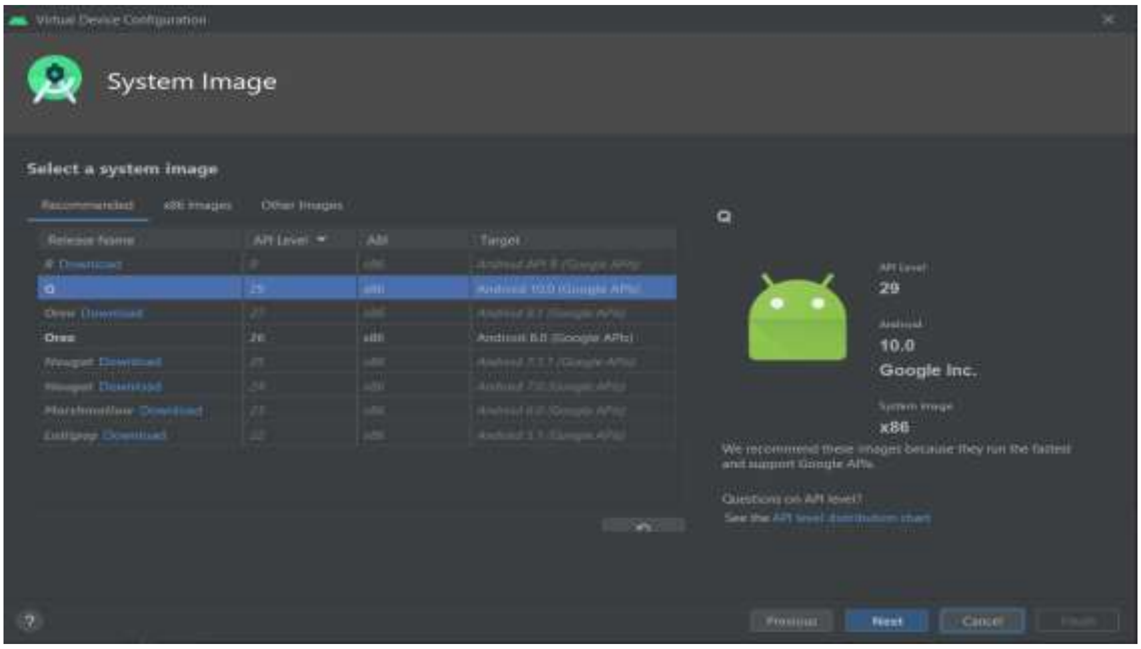
Pixel 3 XL

Size: large
Ratio: long
Density: 560dpi

Figure 2.4 Selecting device

Step 3

Select version of android you want to run in your virtual device.



Release Name	API Level	ABI	Target
R Download	R	x86	Android API R (Google APIs)
Q	29	x86	Android 10.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo	26	x86	Android 8.0 (Google APIs)
Nougat Download	25	x86	Android 7.1.1 (Google APIs)
Nougat Download	24	x86	Android 7.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0 (Google APIs)
Lollipop Download	22	x86	Android 5.1 (Google APIs)

Q

API Level
29

Android
10.0
Google Inc.

System Image
x86

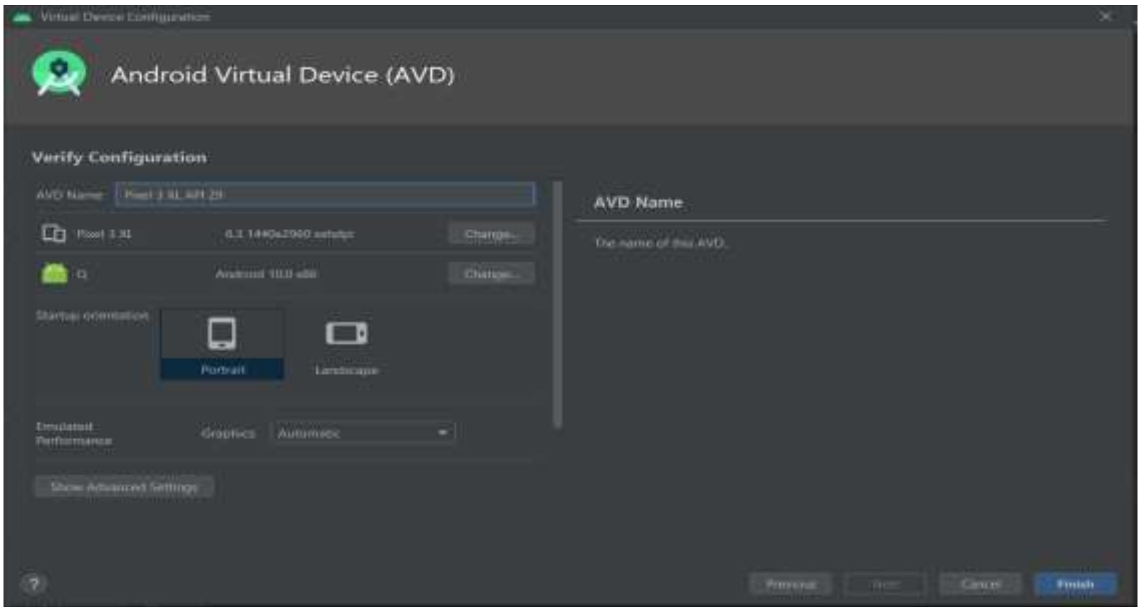
We recommend these images because they run the fastest and support Google APIs.

Questions on API level?
See the API level distribution chart

Figure 2.5 Selecting Android Version

Step 4

Click next when prompted and finish virtual device creation.



AVD Name: Pixel 3 XL API 29

Pixel 3 XL 6.3 1440x2960 xxhdpi

Q Android 10.0 x86

Startup orientation: Portrait Landscape

Emulated Performance Graphics: Automatic

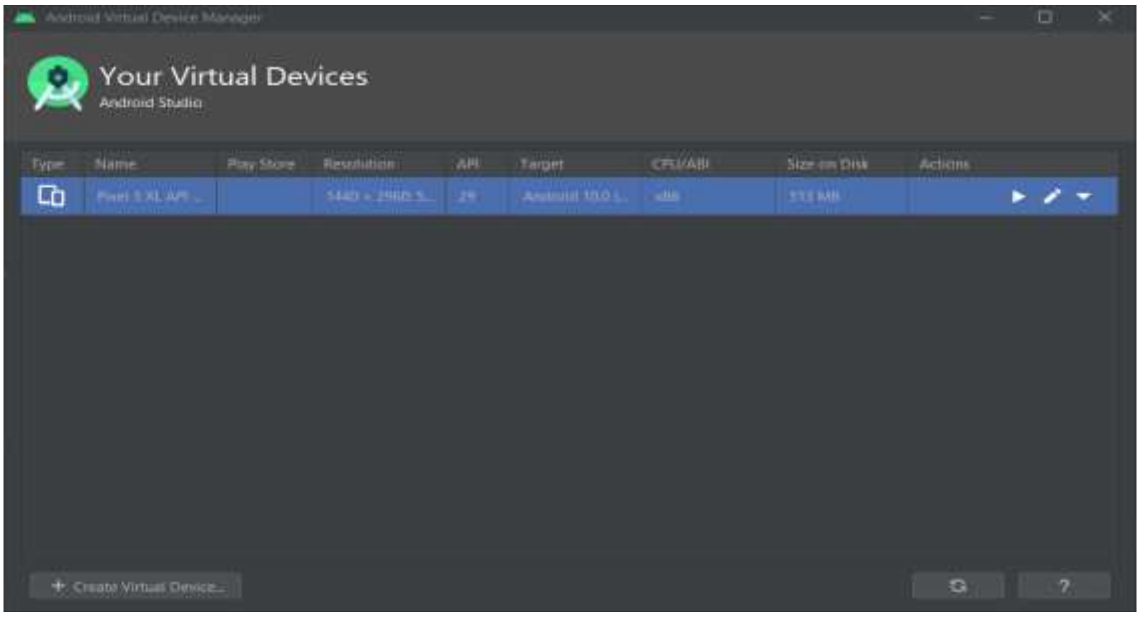
AVD Name

The name of this AVD.

Figure 2.6 Selecting Device mode

Step 5

When device is ready, click on play button to start the device.



Android Virtual Device Manager

Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Pixel 3 XL API ...		1440 × 2960: 5...	29	Android 10.0 (...	x86	513 MB	

Figure 2.7 Starting Virtual device

Step 6

Your device is ready to use.

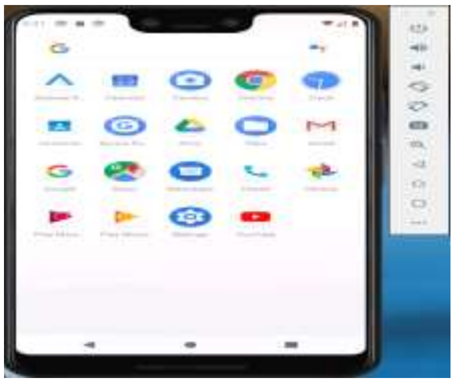


Figure 2.8 Virtual Android Device

2.5.3 CREATING NEW PROJECT

Step 1

Start android Studio, and click on ‘start new flutter project’

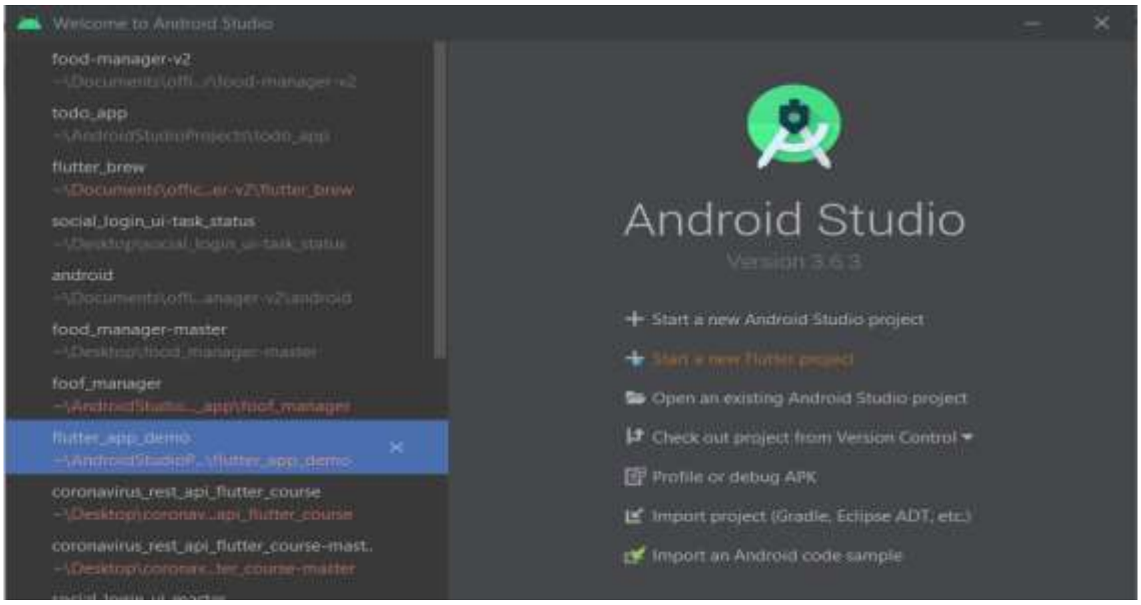
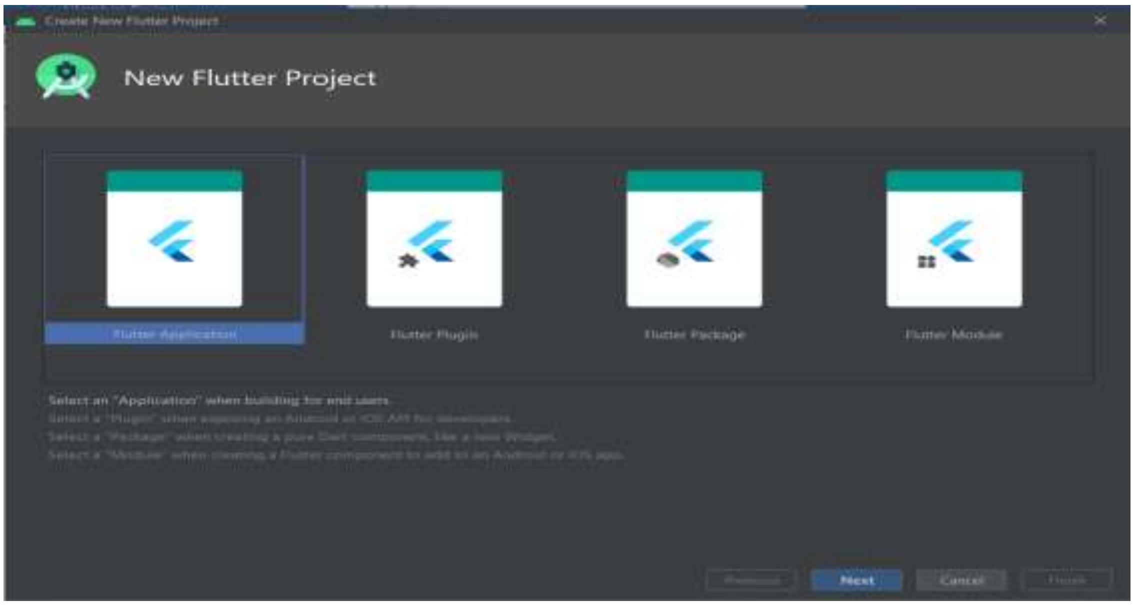

Welcome to Android Studio
food-manager-v2
~\Documents\offi...\food-manager-v2
todo_app
~\AndroidStudioProjects\todo_app
flutter_brew
~\Documents\offic...er-v2\flutter_brew
social_login_ui-task_status
~\Desktop\social_login_ui-task_status
android
~\Documents\offi...anager-v2\android
food_manager-master
~\Desktop\food_manager-master
foof_manager
~\AndroidStudio..._app\foof_manager
flutter_app_demo
~\AndroidStudioP...\flutter_app_demo
coronavirus_rest_api_flutter_course
~\Desktop\coronav...api_flutter_course
coronavirus_rest_api_flutter_course-mast...
~\Desktop\coronav...ter_course-master
Android Studio
Version 3.6.3
Start a new Android Studio project
Start a new Flutter project
Open an existing Android Studio project
Check out project from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample

Figure 2.9 Creating new project

Step 2

Select flutter application and click next.

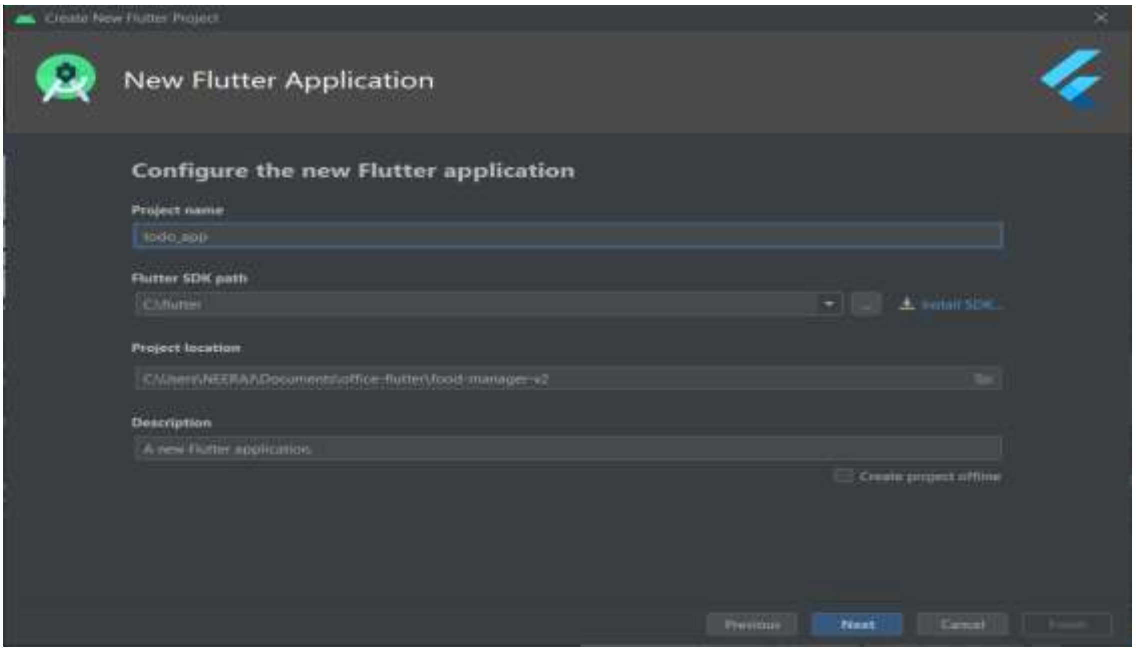


Select an "Application" when building for end users.
Select a "Plugin" when exposing an Android or iOS API for developers.
Select a "Package" when creating a pure Dart component, like a new Widget.
Select a "Module" when creating a Flutter component to add to an Android or iOS app.

Figure 2.10 Select Flutter Application

Step 3

Give your project a name, Flutter SDK path, location and click next.


Create New Flutter Project

New Flutter Application

Configure the new Flutter application

Project name: todo_app

Flutter SDK path: C:\flutter

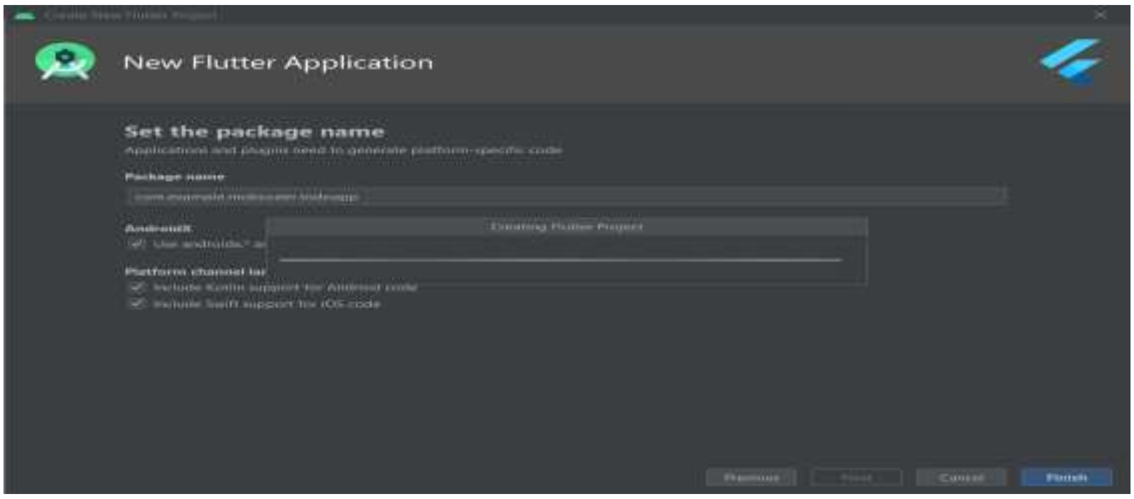
Project location: C:\Users\NEERAJ\Documents\office-flutter\food-manager-v2

Description: A new Flutter application.

Figure 2.11 Give your project a name

Step 4

Give your package name which should be unique and then click Finish. It will take few seconds and will prompt you on new screen with demo application.


Create New Flutter Project

New Flutter Application

Set the package name

Applications and plugins need to generate platform-specific code

Package name: com.example.[illegible].todoapp

AndroidX

✓ Use androidx.* ar

Platform channel lar

✓ Include Kotlin support for Android code

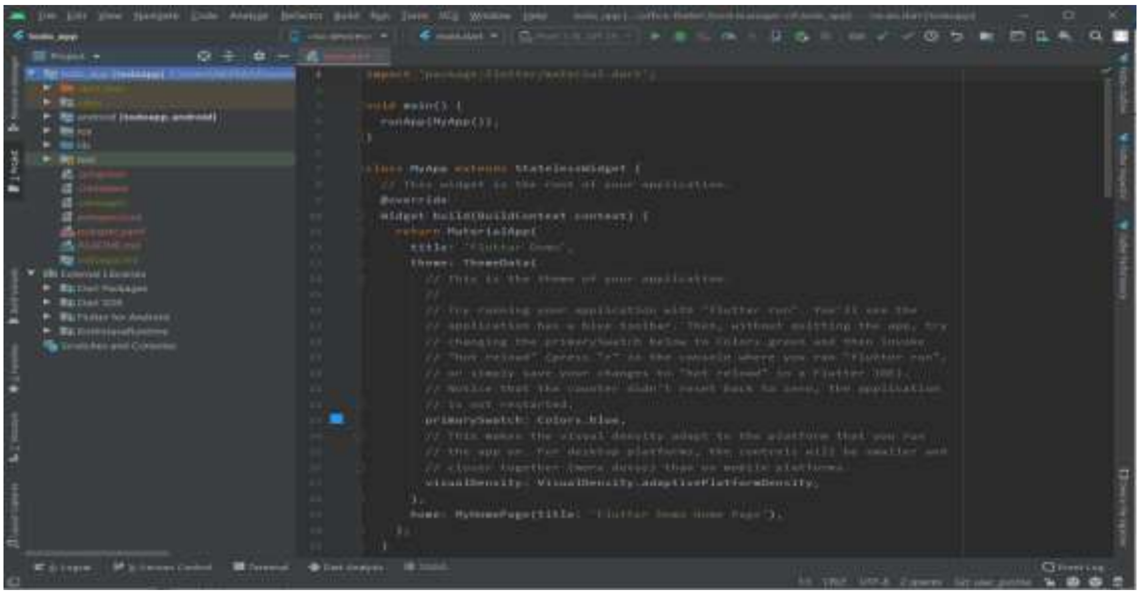
✓ Include Swift support for iOS code

Creating Flutter Project

Figure 2.12 Give a unique package name

Step 5

This is demo application screen with it’s



```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
        // This makes the visual density adapt to the platform that you run
        // the app on. For desktop platforms, the controls will be smaller and
        // closer together (more dense) than on mobile platforms.
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
```

Figure 2.13 Demo project

2.6 OPERATING SYSTEM FOR ANDROID DEVELOPMENT

This is probably the one topic you don’t have to worry about. Either you can pick the operating system on your computer used for development, or it is limited by the IT-policies of your employer. For most Android developers, any of the official supported operating system works fine. However, there are situations where the choice will matter.

Google supports Windows, Linux, and OS X for developing Android applications. Although Windows is officially supported by the Android SDK, you’ll have problems if you decide to do advanced development, especially when it comes to writing native applications or building your own custom ROM. The best choice is either Linux or OS X. If possible, try to have one of these as your primary operating system, and you’ll run into far fewer problems. Another reason for avoiding Windows on your Android development environment is that you won’t need to install new USB drivers for every Android device you work on.[4]

CHAPTER 3

TECHNICAL FEASIBILITY

3.1 FEASIBILITY STUDY

Feasibility is a measure of how beneficial the development of the information system will be to an organization. This is done by investigating the existing system in the area under investigation or general ideas about a new system. It is a test of a system proposal according to its workability, impact on the organization, ability to meet user needs and effective use of resources. A prefeasibility study seeks to validate a project and to expose organizational spots or problems that may condemn it before it sees the light of day, using a limited number of analytical tools. A prefeasibility study does not concentrate on the day-to-day operations; in other words, it does not investigate the specifics of tasks or the team member composition. This is something that the feasibility study does. From this perspective, a feasibility study is very pragmatic; once we have defined the project in approximate terms using the five frames of analysis, we can then examine how this is going to translate into daily operations, most evident during the transformation phase.[8]

As an example, let us take the case of the replacement of the Champlain Bridge that connects the island of Montréal to its south shore and the United States and which is nearing the end of its useful life. The 4.2-billion-dollar (C$) project will inevitably generate a perimeter of intense noise (machinery, etc.) and dust in an urban area that is quite populated. Natural ecosystems are threatened. A thorough environmental feasibility study is required. A feasibility analyst could linger on whether the construction is feasible, taking into account dust and displacements of rare animal and plant species as well as maximum noise levels. Of course, that would not be enough in this case: other aspects such as financial and technical concerns would also have to be addressed.[2]

3.2 TECHNICAL FEASIBILITY

In examining the Technical feasibility of the system, more importance is given to the hardware interaction part of the system. The assessments of technical feasibility center on the existing system and to what extent it can support the proposed addition. This was based on an outline design of system requirements in turns of inputs, files, programs, procedures, and staff. It involves financial considerations to accommodate technical enhancements.

In this project, the technical team can convert their ideas into a working model which shows that the project is TECHNICAL FEASIBLE. The hardware and software used in the project are easily available and easily used. The project is very easy to use by different users which also prove that the project is TECHNICALFEASIBLE.

3.3 TECHNOLOGY DESCRIPTION

Technology is the collection of techniques, skills, methods, and processes used in the production of goods or services or in the production of goods or services in the accomplishment of objectives, such as scientific investigation. Technology can be the knowledge of techniques, process, etc., or it can be embedded in machines, computers, devices, and factories which can operate by individuals without detailed knowledge of working on such things.

Technology can be defined as following ways:

- Tangible: blueprints, models, operating manuals, prototypes.
- Intangible: consultancy, problem-solving and training methods.
- High: entirely or almost entirely automated and intelligent technology that manipulates ever finer matter and ever powerful forces.
- Intermediate: semi-automated partially intelligent technology that manipulates refined matter and medium level forces.
- Low: labor intensive technology that manipulates only coarse or gross matter and weaker forces.

3.4 TECHNOLOGY USED IN THE PROJECCT

3.4.1 FLUTTER

Here are various technologies being used now. These technologies play vital role in the development of the solutions to the problems. In this project being used is Flutter, Dart language. Since this is era of mobile devices, everyone carries an Android device or an apple device. Flutter gives you the liberty to make applications for both platforms by using single code base.

Flutter is an open-source UI software development kit created by Google It is used to develop applications for Android, iOS, Windows, Mac, Linux, Google Fuchsia, and the web. The first version of Flutter was known as codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit, with the stated intent of being able to render consistently at 120 frames per second.[1]

3.4.1.1 FRAMEWORK ARCHITECTURE

The major components of flutter include:

- Dart platform
- Flutter engine
- Foundation library
- Design-specific widgets

3.4.1.1.1 DART PLATEFORM

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 the semi-official Flutter Desktop Embedding project, 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 can be reflected immediately in the running requiring a restart or any loss of state. This feature as implemented in Flutter has received widespread praise.

Release versions of Flutter apps are compiled with ahead-of-time (AOT) compilation on both Android and iOS, making Flutter's high performance on mobile devices possible.

3.4.1.1.2 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. Most developers will interact with Flutter via the Flutter Framework, which provides a modern, reactive framework, and a rich set of platform, layout, and foundation widgets.

3.4.1.1.3 FOUNDATION LIBRARY

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

3.4.1.1.4 WIDGETS

UI design in Flutter involves using composition to assemble / create "Widgets" from other Widgets. The trick to understanding this is to realize that any tree of components (Widgets) that is assembled under a single build() method is also referred to as a single Widget. This is because those smaller Widgets are also made up of even smaller Widgets, and each has a build () method of its own. This is how Flutter makes use of Composition.

The docs say: " A widget is an immutable description of part of a user interface." A human being will tell you it is a Blueprint, which is a much easier way to think about it. However, one also needs to keep in mind there are many types of Widgets in Flutter, and you cannot see or touch all of them. Text is a Widget, but so is its TextStyle, which defines things like size, color, font family and weight. There are Widgets that represent things, ones that represent characteristics (like TextStyle) and even others that do things, like FutureBuilder and StreamBuilder.

Complex widgets can be created by combining many simpler ones, and an app is actually just the largest Widget of them all (often called "MyApp"). The MyApp Widget contains all the other Widgets, which can contain even smaller Widgets, and together they make up your app.

However, the use of widgets is not strictly required to build Flutter apps. An alternative option, usually only used by people who like to control every pixel drawn to the canvas, is to use the Foundation library's methods directly. These methods can be used to draw shapes, text, and imagery directly to the canvas. This ability of Flutter has been utilized in a few frameworks, such as the open-source Flame game engine.

3.4.1.1.5 DESIGN-SPECIFIC WIDGETS

The Flutter framework contains two sets of WIDGETS which 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.

3.4.1.2 FEATURS OF FLUTTER

- Fast Development
- Expressive and Flexible UI
- Native Performance

3.4.1.2.1 FAST DEVELOPMENT

Paint your app to life in milliseconds with Stateful Hot Reload. Use a rich set of fully customizable widgets to build native interfaces in minutes.

Flutter's *hot reload* helps you quickly and easily experiment, build UIs, add features, and fix bugs faster. Experience sub-second reload times without losing state on emulators, simulators, and hardware.

3.4.1.2.2 EXPRESSIVE AND FLEXIBLE UI

Quickly ship features with a focus on native end-user experiences. Layered architecture allows for full customization, which results in incredibly fast rendering and expressive and flexible designs.

Delight your users with Flutter's built-in beautiful Material Design and Cupertino (iOS-flavor) widgets, rich motion APIs, smooth natural scrolling, and platform awareness.

3.4.1.2.3 NATIVE PERFORMANCE

Flutter’s widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts, and your Flutter code is compiled to native ARM machine code using Dart’s native compilers.

Flutter’s widgets incorporate all critical platform differences such as scrolling, navigation, icons and fonts to provide full native performance on both iOS and Android.

3.4.1.3 CATEGORIES OF FLUTTER APPLICATION

- Mobile
- Web
- Desktop

3.4.1.4 ENVIRONMENT SETUP

STEP-1

To install and run Flutter, your development environment must meet these minimum requirements:

- **Operating Systems**: Windows 7 SP1 or later (64-bit)
- **Disk Space**: 400 MB (does not include disk space for IDE/tools).
- **Tools**: Flutter depends on these tools being available in your environment.
  - Windows PowerShell 5.0 or newer (this is pre-installed with Windows 10)
  - Git for Windows 2.x, with the **Use Git from the Windows Command Prompt** option.

  If Git for Windows is already installed, make sure you can run `git` commands from the command prompt or PowerShell.

STEP-2

1. Download the installation bundle from flutter.dev to get the latest stable release of the Flutter SDK.
2. tract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`; do not install Flutter in a directory like `C:\Program Files\` that requires elevated privileges).

STEP-3

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the PATH environment variable:

- From the Start search bar, enter 'env' and select **Edit environment variables for your account**.
- Under **User variables** check if there is an entry called **Path**:
  - If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
  - If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.

Note that you have to close and reopen any existing console windows for these changes to take effect.

STEP-4

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in **bold** text).

For example:

```
[-] Android toolchain - develop for Android devices
    • Android SDK at D:\Android\sdk
    ✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ
    • Try re-installing or updating your Android SDK,
      visit https://flutter.dev/setup/#android-setup for detailed instructions.
```

The following sections describe how to perform these tasks and finish the setup process. Once you have installed any missing dependencies, you can run the flutter doctor command again to verify that you've set everything up correctly.

3.4.1.5 ANDROID SETUP

1. Install Android Studio

1. Download and install Android Studio.
2. Start Android Studio and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.

2. Set up your Android device

To prepare to run and test your Flutter app on an Android device, you'll need an Android device running Android 4.1 (API level 16) or higher.

1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the Android documentation.
2. Windows-only: Install the Google USB Driver.
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID_HOME environment variable to that installation directory.

3. Set up the Android emulator

To prepare to run and test your Flutter app on the Android emulator, follow these steps:

1. Enable VM acceleration on your machine.
2. Launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device**. (The **Android** submenu is only present when inside an Android project.)
3. Choose a device definition and select **Next**.
4. Select one or more system images for the Android versions you want to emulate and select **Next**. An *x86* or *x86_64* image is recommended.
5. Under Emulated Performance, select **Hardware - GLES 2.0** to enable hardware acceleration.
6. Verify the AVD configuration is correct and select **Finish**.

   For details on the above steps, see Managing AVDs.

7. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

CHAPTER 4

BACKEND DESIGN

4.1 DATAFLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its process aspects. A DFD is often used as a preliminary stem to create an overview of the system, which can later be elaborated. DFDs can also use for the visualisation of data processing (Structure Design).

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel (which is shown on a flow chart).

Data flow diagram can be used in both Analysis and Design phase of the System Development Life Cycle (SDLC).

4.1.1 DFD COMPONENTS

DFD consists of processes, flows, warehouses, and terminators. There are several ways to view these DFD components.

4.1.1.1 PROCESS

The process (function, transformation) is part of a system that transforms inputs to outputs. The symbol of a process is a circle, an oval, a rectangle, or a rectangle with rounded corners (according to the type of notation). The process is named in one word, a short sentence, or a phrase that is clearly to express its essence.

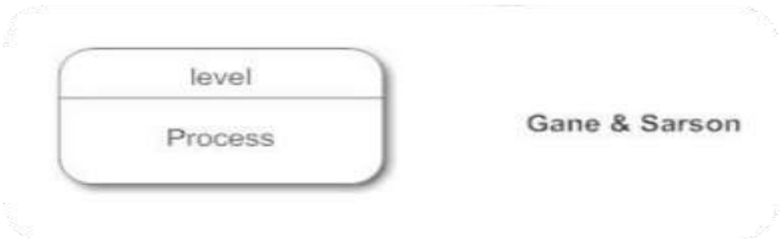


Figure 4.1 Process

4.1.1.2 DATA FLOW

Data flow (flow, dataflow) shows the transfer of information (sometimes also material) from one part of the system to another. The symbol of the flow is the arrow. The flow should have a name that determines what information (or what material) is being moved. Exceptions are flows where it is clear what information is transferred through the entities that are linked to these flows. Material shifts are modelled in systems that are not merely informative. Flow should only transmit one type of information (material). The arrow shows the flow direction (it can also be bi-directional if the information to/from the entity is logically dependent - e.g. question and answer). Flows link processes, warehouses, and terminators.

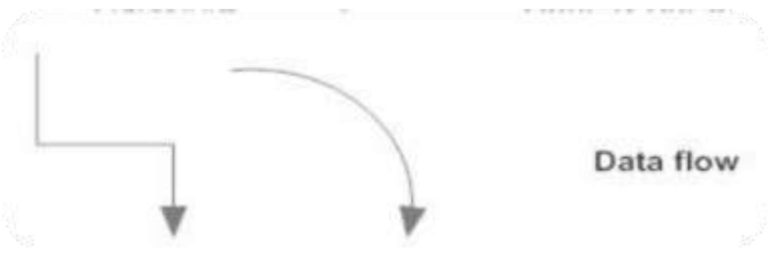


Figure 4.2 Dataflow

4.1.1.3 WAREHOUSE

The warehouse (datastore, data store, file, database) is used to store data for later use. The symbol of the store is two horizontal lines, the other way of view is shown in the DFD Notation. The name of the warehouse is a plural noun (e.g. orders) - it derives from the input and output streams of the warehouse. The warehouse does not have to be just a data file, for example, a folder with documents, a filing cabinet, and optical discs. Therefore, viewing the warehouse in DFD is independent of implementation. The flow from the warehouse usually represents the reading of the data stored in the warehouse, and the flow to the warehouse usually expresses data entry or updating (sometimes also deleting data). Warehouse is represented by two parallel lines between which the memory name is located (it can be modelled as a UML buffer node).

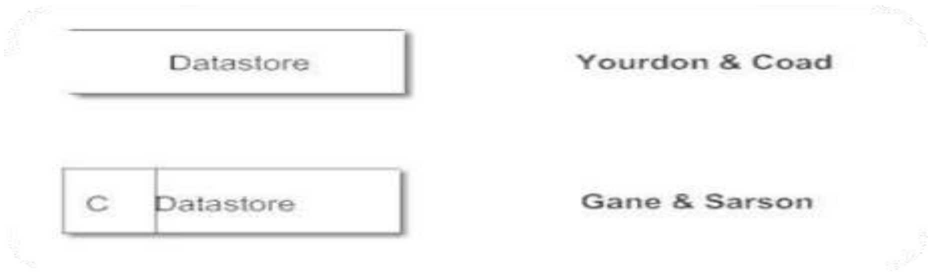


Figure 4.3 Database/Data Warehouse

4.1.1.4 TERMINATOR

The Terminator is an external entity that communicates with the system and stands outside of the system. It can be, for example, various organizations (e.g. a bank), groups of people (e.g. customers), authorities (e.g. a tax office) or a department (e.g. a human-resources department) of the same organization, which does not belong to the model system. The terminator may be another system with which the modelled system communicates.

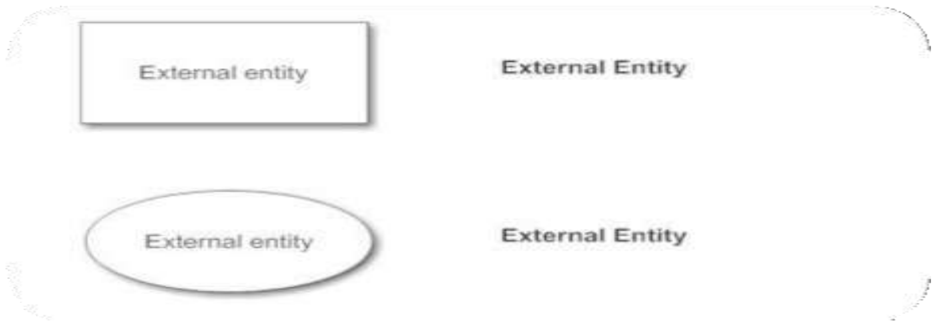


Figure 4.4 Terminator (External Entity)

4.2 DFD OF THE PROJECT

This project contains 0 (Context level) level and 1 level DFD.

4.2.1 LEVEL 0 DFD

A context diagram is a top level (also known as "Level 0") data flow diagram. It only contains one process node ("Process **0**") that generalizes the function of the entire system in relationship to external entities.

This level simply shows the admin who ultimately have complete access of the application in square box and complete application in circle.

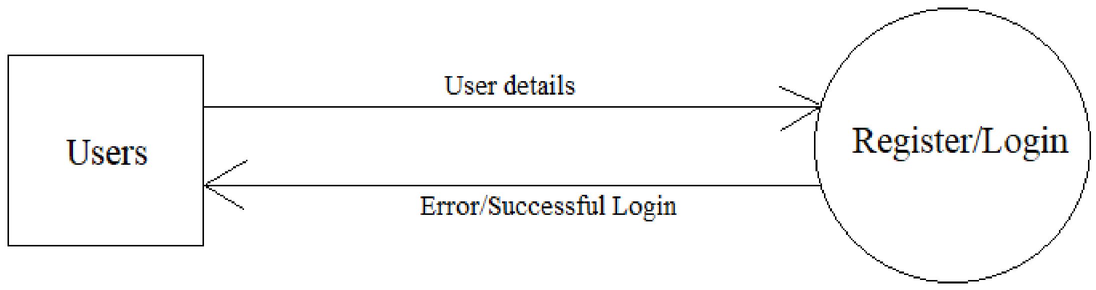
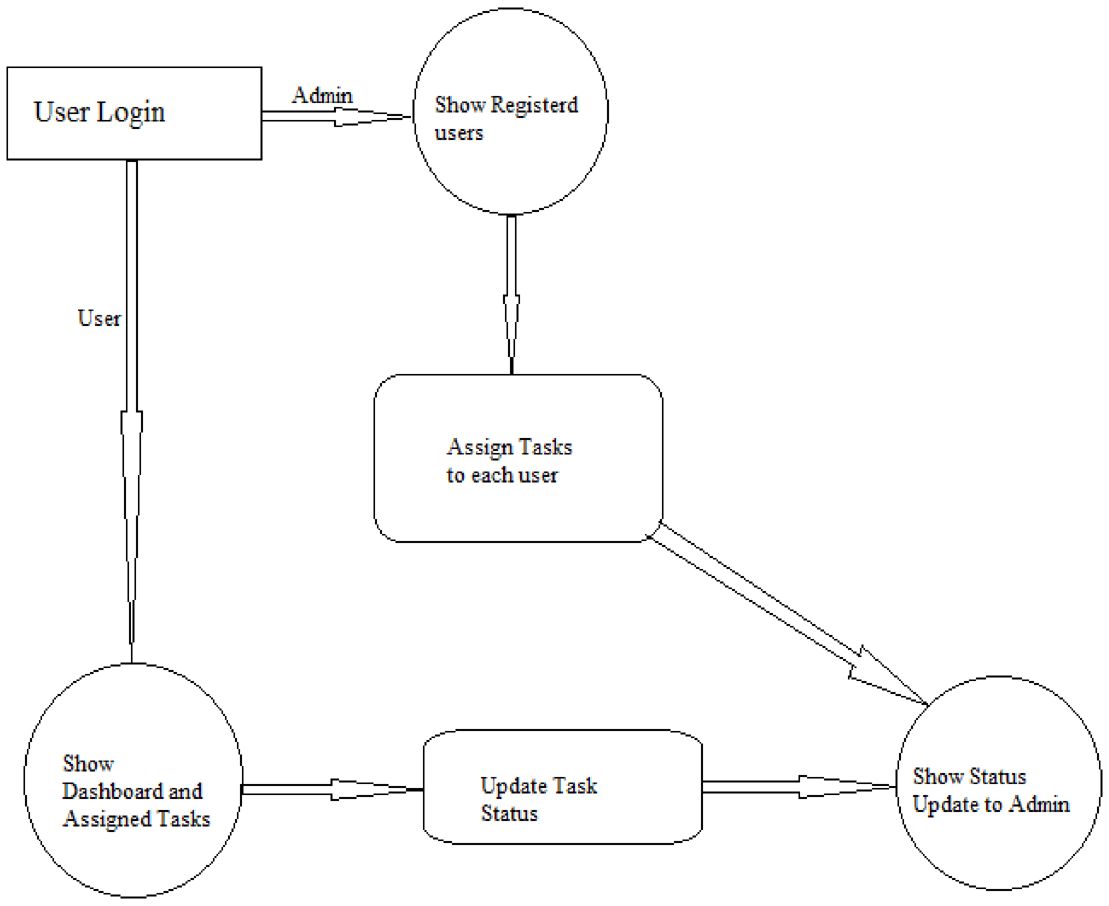


Figure 4.5 Level 0 DFD

4.2.2 LEVEL 1 DFD

The level 1 DFD shows how the system is divided into sub-system (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. This level will show all the process and database used in the application and the flow data between various processes.


User Login
Admin
Show Registerd users
User
Assign Tasks to each user
Show Dashboard and Assigned Tasks
Update Task Status
Show Status Update to Admin

Figure

4.6 Level 1 DFD

CHAPTER 5

FRONTEND DESIGN

5.1 SPLASH SCREEN

This first screen of the application it shows when a user opens an application which displays the logo and name of the application for very short period. In this screen, we are providing a very effective animation.

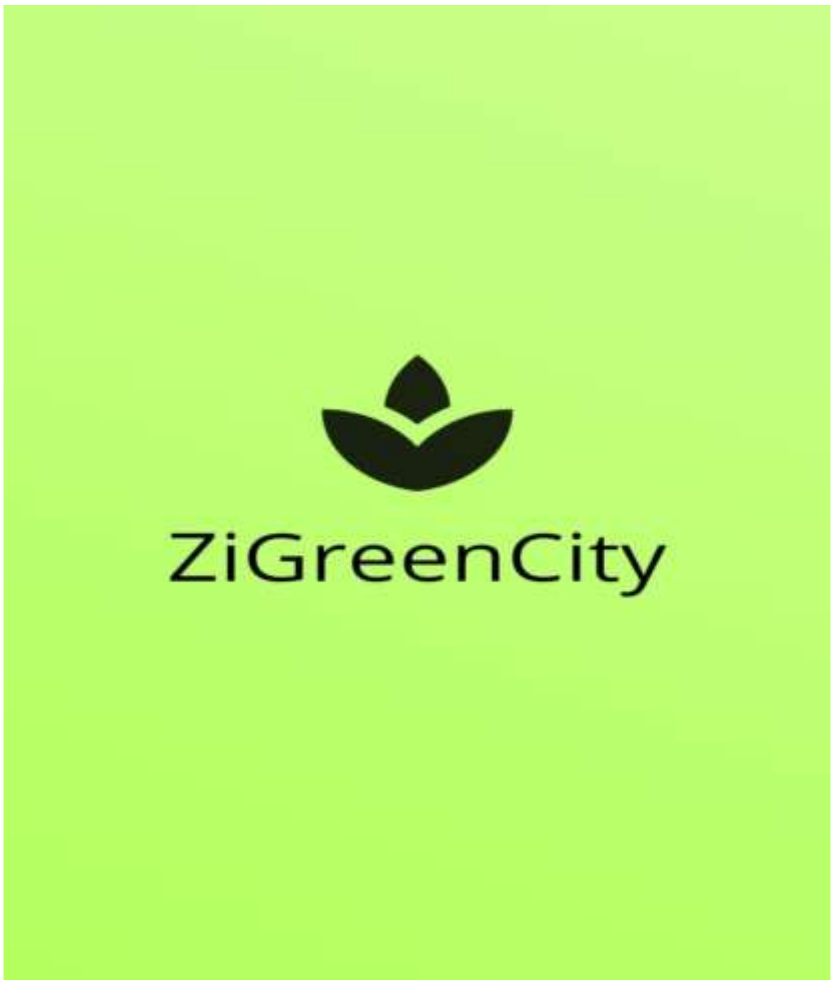


Figure 5.1 Splash Screen

5.2 LOGIN SCREEN

Login screen provides the facility to access the application with valid username and password details. For Login User must enter User Id and Password for Login and click Login button. Only authorized person can access the application.

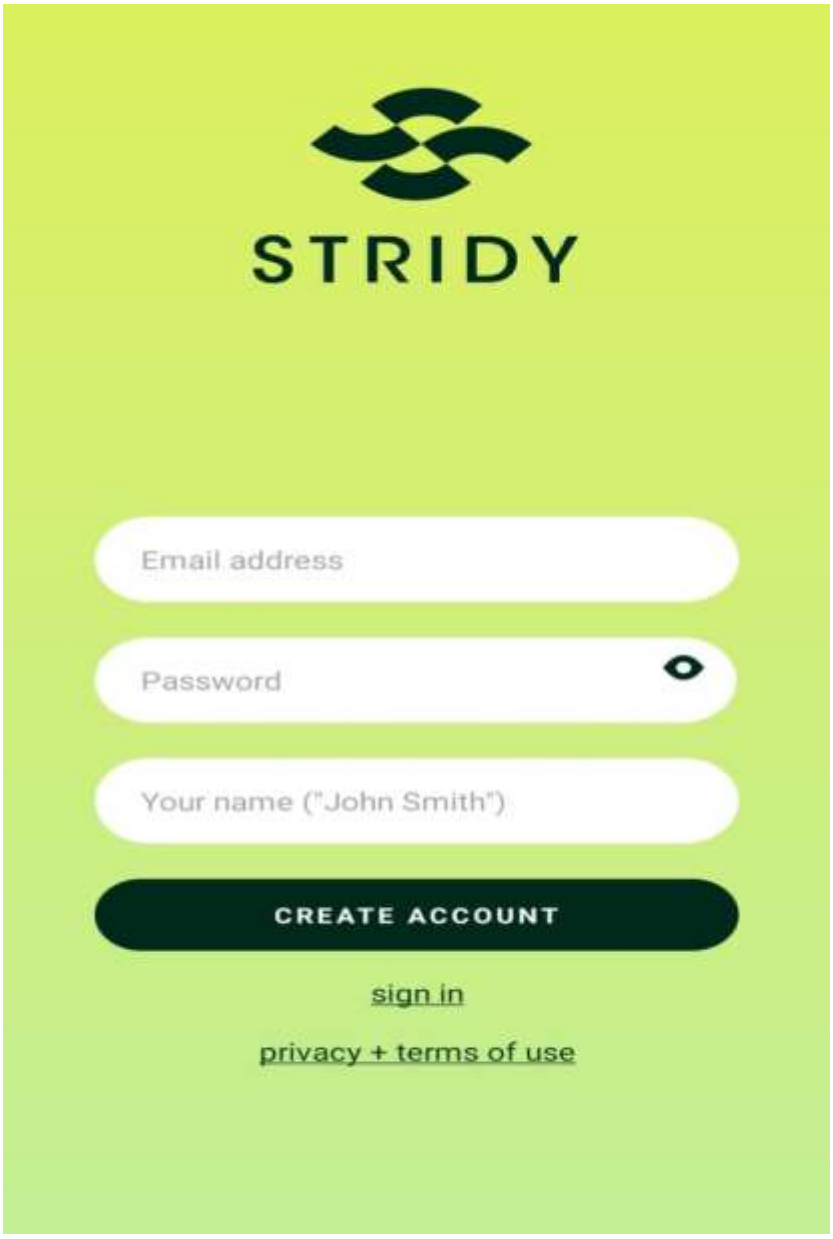


Figure 5.2 Login Screen

5.3 REGISTRATION SCREEN

The registration screen is for those users who are using this application for the first time. In registration few basic details about the user will be asked and stored into the database of the application.

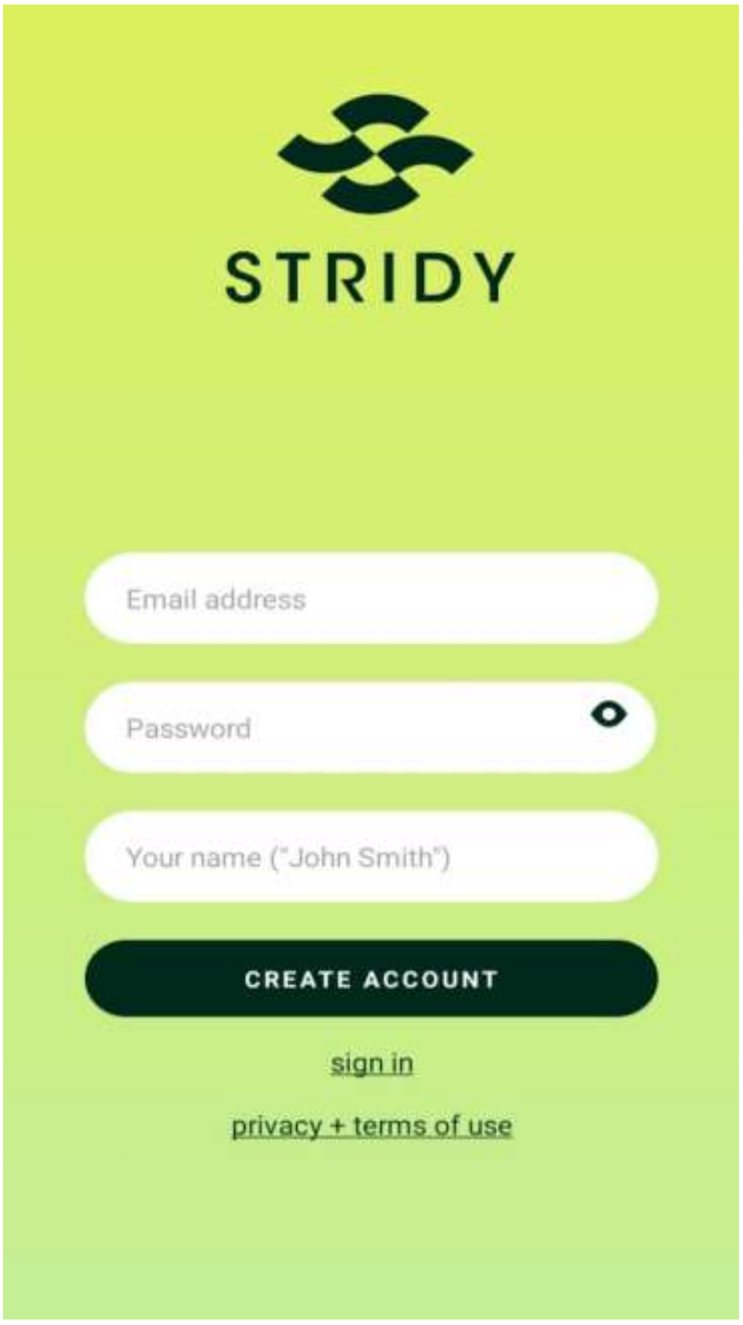


Figure 5.3 Registration Screen

5.4 Forget Password Screen

This is the first screen which appears when user forgot his password.

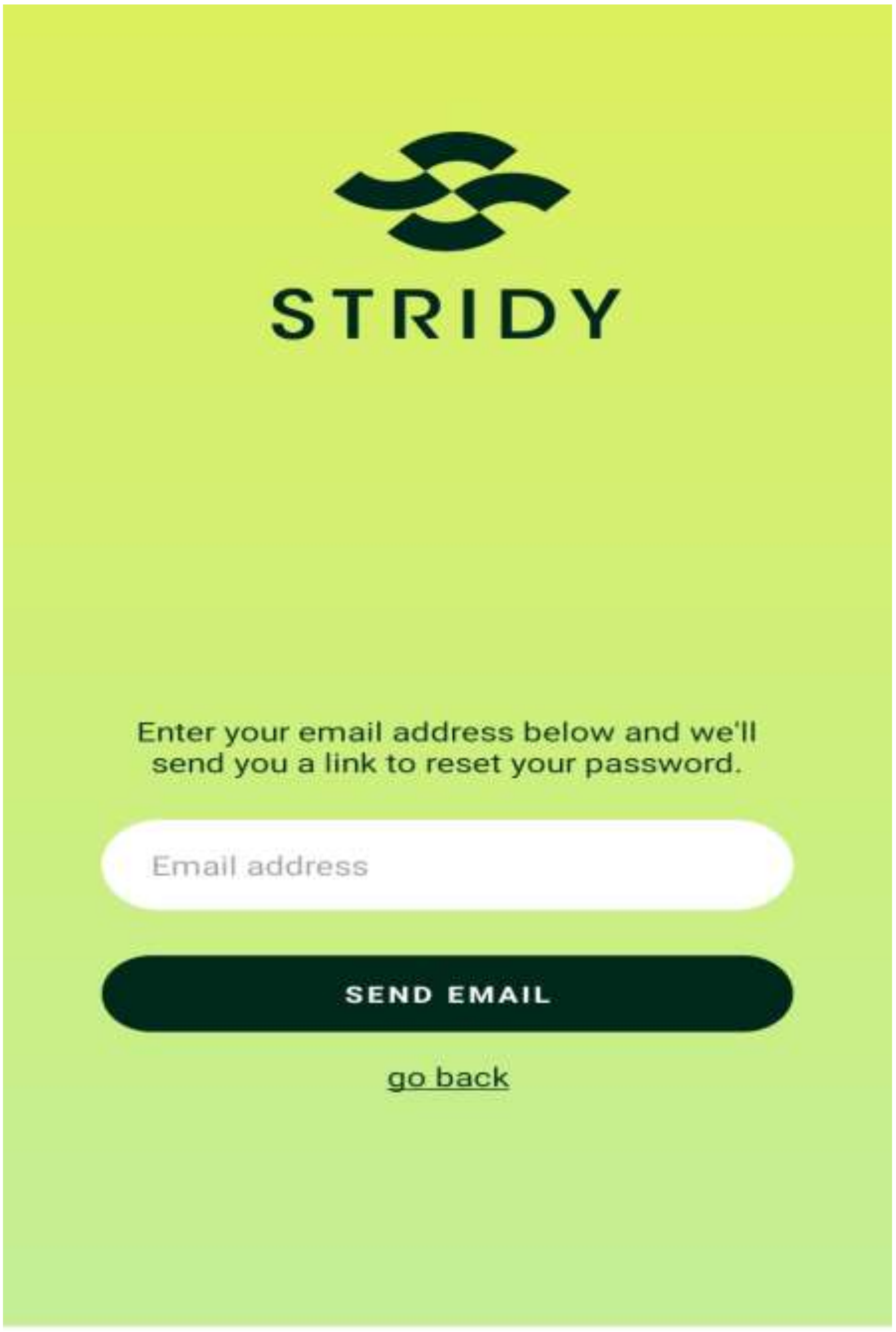

STRIDY
Enter your email address below and we'll send you a link to reset your password.
Email address
SEND EMAIL
go back

Figure 5.4 Forget Password

5.5 Profile Screen

This screen will show stats and achievements of registered users.

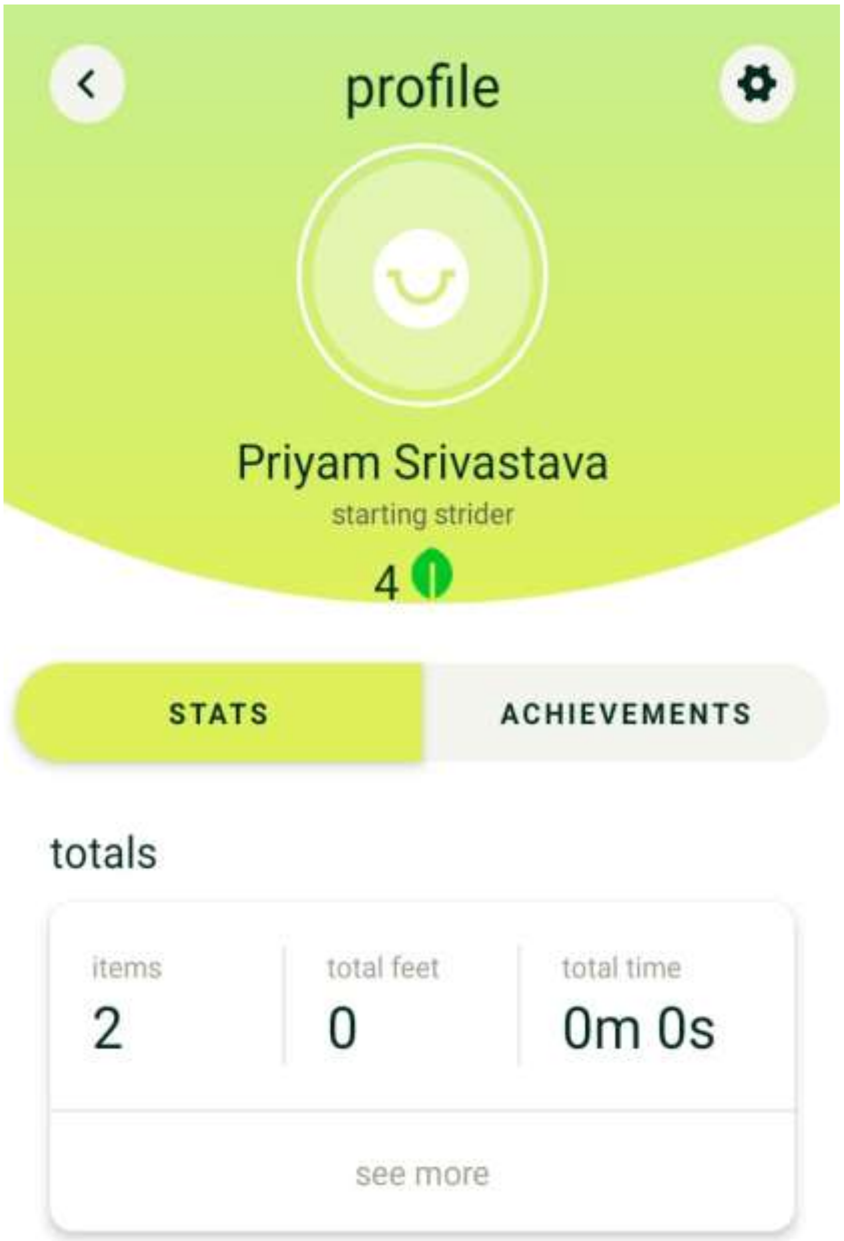


Figure 5.5 Profile Screen

5.6 Journey Screen

This screen will store keep record of when registered user started journey, when he collected item and when he ends his journey.

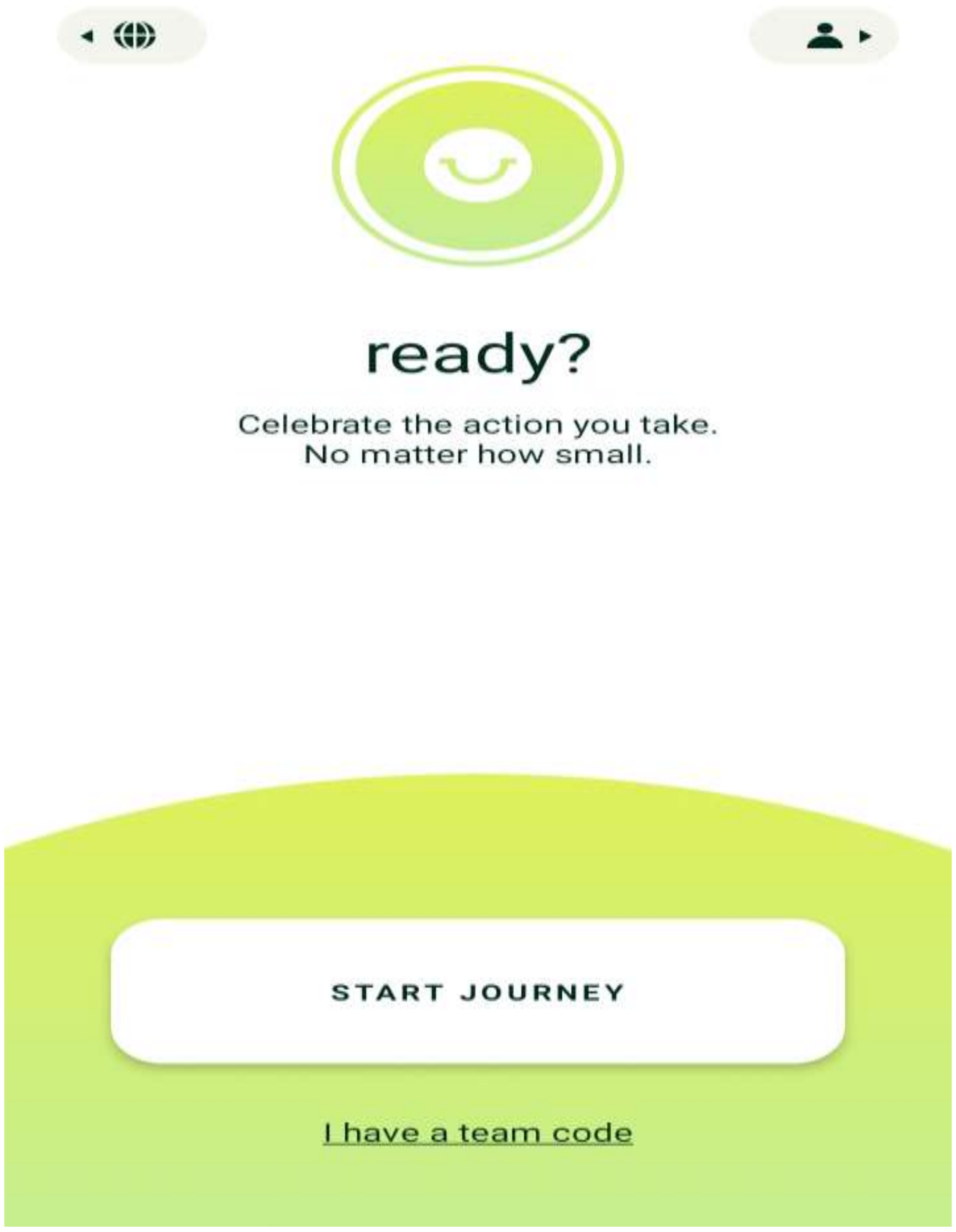
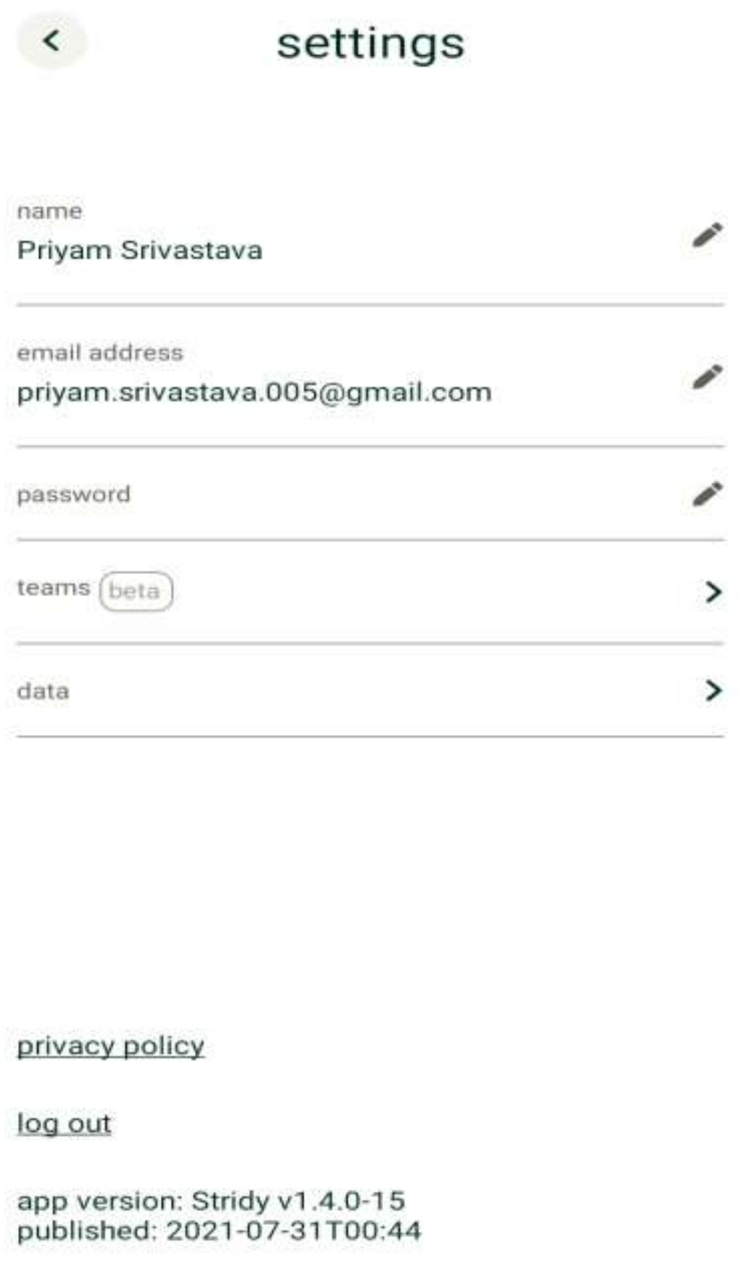


Figure 5.6 User List

5.7 Settings Screen

This screen will contain user details like name and email address, and here he can also change his password and also logout from app.



settings

name
Priyam Srivastava

email address
priyam.srivastava.005@gmail.com

password

teams beta

data

privacy policy

log out

app version: Stridy v1.4.0-15
published: 2021-07-31T00:44

Figure 5.7 User Profile

CHAPTER 6

CODING

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 via the semi-official Flutter Desktop Embedding project, 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 can be reflected immediately in the running app without requiring a restart or any loss of state. This feature as implemented in Flutter has received widespread praise.

6.1 SPLASH SCREEN

```
import 'package:flutter/material.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:flutter_template/viewmodel/splash/splash_viewmodel.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_progress_indicato
r.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:get_it/get_it.dart';

class SplashScreen extends StatefulWidget {
  static const String routeName = 'splash';

  const SplashScreen({
    Key? key,
  }) : super(key: key);

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

@visibleForTesting
class SplashScreenState extends State<SplashScreen> implements SplashNavigator {
  @override
  Widget build(BuildContext context) {
    return ProviderWidget<SplashViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilderWithViewModel: (context, viewModel, theme, _) => Scaffold(
        backgroundColor: theme.colorsTheme.backgroundDark,
        body:
        Stack(children: [
          Container(
            decoration: BoxDecoration(
                gradient: LinearGradient(
                    begin: Alignment.topRight,
                    end: Alignment.bottomLeft,
                    colors:
[theme.colorsTheme.lightGreen,theme.colorsTheme.darkGreen])),
          ),
          Center(child:Column(crossAxisAlignment: CrossAxisAlignment.center,
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Icon(Icons.spa,size: 100,),
              Container(height: ThemeDimens.padding8),
              Text("ZiGreenCity",style:theme.lightTextTheme.titleHugeStridy),
            ],
          )),
        ],)
        // const Center(
        //  child: FlutterTemplateProgressIndicator.light(),
        // ),
      ),
    );
  }

  @override
  void goToCleantheearth() => MainNavigatorWidget.of(context).goToCleantheearth();
  @override
  void goToLogin() => MainNavigatorWidget.of(context).goToLogin();
  @override
  void goToCreateAccScreen() =>
MainNavigatorWidget.of(context).goToCreateAccScreen();
  @override
  void goToCleanthemind() => MainNavigatorWidget.of(context).goToCleanthemind();
  @override
  void goToConnectwithcommunity() =>
MainNavigatorWidget.of(context).goToConnectwithcommunity();
}
```

6.2 LOGIN SCREEN

```
import
'package:flutter_template/widget/general/styled/flutter_template_progress_indicato
r.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:flutter_template/util/keys.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_button.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_input_field.dart'
;
import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';

class LoginScreen extends StatefulWidget {
  static const String routeName = 'login';

  const LoginScreen({Key? key}) : super(key: key);

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

@visibleForTesting
class LoginScreenState extends State<LoginScreen> with ErrorNavigatorMixin
implements LoginNavigator {
  @override
  Widget build(BuildContext context) {
    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            backgroundColor: theme.colorsTheme.backgroundDark,
            body: SafeArea(
              child: Container(
                width: double.infinity,
                padding: const EdgeInsets.all(ThemeDimens.padding16),
                child: Column(
                  children: [
                    Container(height: ThemeDimens.padding16),
                    Text(
                      'Login',
                      style: theme.lightTextTheme.titleNormal,
                      textAlign: TextAlign.center,
                    ),
                    Container(height: ThemeDimens.padding32),
                    Text(
                      'Just fill in some text. There is no validator for the
login',
                      style: theme.lightTextTheme.labelButtonSmall,
                    ),
                    Container(height: ThemeDimens.padding32),
                    FlutterTemplateInputField(
                      key: Keys.emailInput,
                      enabled: !viewModel.isLoading,
                      onChanged: viewModel.onEmailUpdated,
                      hint: 'Email',
                    ),
                    Container(height: ThemeDimens.padding16),
                    FlutterTemplateInputField(
                      key: Keys.passwordInput,
                      enabled: !viewModel.isLoading,
                      onChanged: viewModel.onPasswordUpdated,
                      hint: 'Password',
                    ),
                    Container(height: ThemeDimens.padding16),
                    if (viewModel.isLoading) ...{
                      const FlutterTemplateProgressIndicator.light(),
                    } else
                      FlutterTemplateButton(
                        key: Keys.loginButton,
                        isEnabled: viewModel.isLoginEnabled,
                        text: 'Login',
                        onClick: viewModel.onLoginClicked,
                      ),
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    );
  }

  @override
  void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
```

6.3 REGISTRATION SCREEN

```
import 'package:firebase_auth/firebase_auth.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_progress_indicato
r.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_button.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_input_field.dart'
;
import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';
final FirebaseAuth _auth = FirebaseAuth.instance;
class CreateAccScreen extends StatefulWidget {
  static const String routeName = 'createAcc';

  const CreateAccScreen({Key? key}) : super(key: key);

  @override
  CreateAccScreenState createState() => CreateAccScreenState();
}
```

```
@visibleForTesting
class CreateAccScreenState extends State<CreateAccScreen> with ErrorNavigatorMixin
implements LoginNavigator {
  final _formKey = new GlobalKey<FormState>();
  bool _autoValidate = false;
  bool emailSubmited = false;
  String _email='';
  bool _obscureText = true;
  FocusNode focusNode1 = FocusNode();
  FocusNode focusNode2 = FocusNode();
  FocusNode focusNode3 = FocusNode();
  final passwordController = TextEditingController();
  final emailController = TextEditingController();
  final nameController = TextEditingController();

  String validateEmail(String value) {
    String pattern =
r'^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-
Z]{2,}))$';
    RegExp regExp = new RegExp(pattern);
    if (value.length > 0 && !regExp.hasMatch(value)&& !emailSubmited) {
      return "Invalid Email";
    }
    else if(value.isEmpty)
      return "Email Field Can not be Empty";
    else {
      return "null";
    }
  }
  late bool _success;
  String _userEmail = '';
  @override
  void initState() {
    auth();
    super.initState();

  }
  Future<void> auth() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();
    // await FirebaseAuth.instance.useEmulator('http://localhost:9099');

  }
  bool _validateAndSave() {
    final form = _formKey.currentState;

    if (form!.validate()) {
      form.save();
      return true;
    }else {
      setState(() => _autoValidate = true);
    }
    return false;
  }
  Future<void> _register() async{
    final User? user = (await _auth.createUserWithEmailAndPassword(
      email: emailController.text,
      password: passwordController.text,
    )).user;
    if (user != null) {
      setState(() {
        _success = true;
        _userEmail = user.email!;
      });
    } else {
      _success = false;
    }
  }
  @override
  Widget build(BuildContext context) {

    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            // backgroundColor: theme.colorsTheme.secondary,
            body:

            Stack(children: [
              Container(
                decoration: BoxDecoration(
                    gradient: LinearGradient(
                        begin: Alignment.topRight,
                        end: Alignment.bottomLeft,
                        colors:
[theme.colorsTheme.lightGreen,theme.colorsTheme.darkGreen])),
              ),
              SingleChildScrollView(child:
              Center(
                child: Padding(
                  padding: const EdgeInsets.all(8.0),
                  child: Column(crossAxisAlignment: CrossAxisAlignment.center,
                    // mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      Container(height: ThemeDimens.padding16),
                      Container(

                        child: Form(
                          key: _formKey,
                          autovalidate: _autoValidate,
                          child: Column(
                            // shrinkWrap: true,
                            children: <Widget>[
                              Column(crossAxisAlignment:
CrossAxisAlignment.center,
                                children: [
                                  Container(height: ThemeDimens.padding32),
                                  Icon(Icons.spa,size: 40,),
                                  Container(
                                    child:  Text(
                                      'ZiGreenCity',
                                      style:
theme.lightTextTheme.titleHugeStridy,
                                    ),),
                                  Container(height: ThemeDimens.padding32),
                                    Container(height: ThemeDimens.padding32),
                                    Container(height: ThemeDimens.padding32),
                                    Container(
                                        width:
MediaQuery.of(context).size.width/1.5,
                                        height: 55.0,
                                        //padding:
EdgeInsets.only(left:width*0.04, right: width*0.04),
                                        child: TextFormField(

                                          focusNode: focusNode1,
                                          controller: emailController,
                                          onFieldSubmitted: (term) {
                                            focusNode1.unfocus();

FocusScope.of(context).requestFocus(focusNode2);
                                          },
                                          onChanged: (content) {
                                            setState(() {
                                              // enabled = true;
                                            });

                                          },
                                          autofocus: true,
                                          validator: (value) {
                                            if (value!.isEmpty) {
                                              return 'Please enter email';
                                            }
                                            return null;
                                          },
                                          cursorColor:Colors.lightGreen,
                                          decoration: InputDecoration(
                                            contentPadding: EdgeInsets.only(left:
11, right: 3, top: 14, bottom: 14),

                                            errorStyle: TextStyle(fontSize: 11,

height: 0.3),

                                            filled: true,
                                            fillColor: Colors.white,
                                            enabledBorder: OutlineInputBorder(
                                              borderSide: BorderSide(
                                                color:Colors.white,
                                              ),
                                              borderRadius:

BorderRadius.circular(28.0),

                                            ),
                                            focusedBorder: OutlineInputBorder(
                                              borderSide: BorderSide(
                                                color:Colors.white,
                                              ),
                                              borderRadius:

BorderRadius.circular(28.0),

                                            ),
                                            border: OutlineInputBorder(
                                              borderSide: BorderSide(
                                                color:Colors.grey,//this has no
effect

                                              ),
                                              borderRadius:

BorderRadius.circular(28.0),
                                            ),
                                            // border: InputBorder.none,
                                            hintText:"e.g.example@email.com",
                                            hintStyle: TextStyle(fontSize: 12.0,
color:Colors.grey),
//                  contentPadding: EdgeInsetsDirectional.only(start: 10),
                                          ),
                                          keyboardType:
TextInputType.emailAddress,
                                                )
                                            ),
                                            Container(height: ThemeDimens.padding32),
                                            Container(
                                                width:
MediaQuery.of(context).size.width/1.5,
                                                height: 55.0,
                                                //padding:
EdgeInsets.only(left:width*0.04, right: width*0.04),
                                                child: TextFormField(

                                          focusNode: focusNode2,
                                          controller:passwordController,
                                          onFieldSubmitted: (term) {
                                            focusNode2.unfocus();

FocusScope.of(context).requestFocus(focusNode3);
                                          },
                                          onChanged: (content) {
                                            setState(() {
                                              // enabled = true;
                                            });

                                          },
                                          autofocus: true,

                                          validator: (value) {
                                            if (value!.isEmpty) {
                                              return 'Please enter password';
                                            }
                                            return null;
                                          },
                                          cursorColor:Colors.lightGreen,
                                          decoration: InputDecoration(
                                            contentPadding: EdgeInsets.only(left:
11, right: 3, top: 14, bottom: 14),

                                            errorStyle: TextStyle(fontSize: 11,

height: 0.3),

                                            filled: true,
                                            suffixIcon: GestureDetector(
                                              onTap: () {
                                                setState(() {
                                                  _obscureText = !_obscureText;
                                                });
                                              },
                                              child: Icon(
                                                _obscureText ? Icons.visibility :
Icons.visibility_off,

                                                semanticLabel:
                                                _obscureText ? 'show password' :
```

```
'hide password',
                                            ),
                                          ),
                                          fillColor: Colors.white,
                                          enabledBorder: OutlineInputBorder(
                                            borderSide: BorderSide(
                                              color:Colors.white,
                                            ),
                                            borderRadius:
BorderRadius.circular(28.0),
                                          ),
                                          focusedBorder: OutlineInputBorder(
                                            borderSide: BorderSide(
                                              color:Colors.white,
                                            ),
                                            borderRadius:
BorderRadius.circular(28.0),
                                          ),
                                          border: OutlineInputBorder(
                                            borderSide: BorderSide(
                                              color:Colors.grey,//this has no
effect
                                            ),
                                            borderRadius:
BorderRadius.circular(28.0),
                                          ),
                                          // border: InputBorder.none,
                                          hintText:"Password",
                                          hintStyle: TextStyle(fontSize: 12.0,
color:Colors.grey),
//                  contentPadding: EdgeInsetsDirectional.only(start: 10),
                                        ),
                                        keyboardType:
TextInputType.emailAddress,
                                    )
                                ),
                                Container(height: ThemeDimens.padding32),
                                Container(
                                    width:
MediaQuery.of(context).size.width/1.5,
                                    height: 55.0,
                                    //padding:
EdgeInsets.only(left:width*0.04, right: width*0.04),
                                    child: TextFormField(

                                      focusNode: focusNode3,
                                      controller: nameController,
                                      onFieldSubmitted: (term) {
                                        // focusNode1.unfocus();
                                        //
FocusScope.of(context).requestFocus(focusNode2);
                                      },
                                      onChanged: (content) {
                                        setState(() {
                                          // enabled = true;
                                        });

                                      },
                                      autofocus: true,
                                          validator: (value) {
                                            if (value!.isEmpty) {
                                              return 'Please enter your name';
                                            }
                                            return null;
                                          },
                                          cursorColor:Colors.lightGreen,

                                          decoration: InputDecoration(
                                            contentPadding: EdgeInsets.only(left:
11, right: 3, top: 14, bottom: 14),
                                            errorStyle: TextStyle(fontSize: 11,
height: 0.3),

                                            filled: true,
                                            fillColor: Colors.white,
                                            enabledBorder: OutlineInputBorder(
                                              borderSide: BorderSide(
                                                color:Colors.white,
                                              ),
                                              borderRadius:
BorderRadius.circular(28.0),
                                            ),
                                            focusedBorder: OutlineInputBorder(
                                              borderSide: BorderSide(
                                                color:Colors.white,
                                              ),
                                              borderRadius:
BorderRadius.circular(28.0),
                                            ),
                                            border: OutlineInputBorder(
                                              borderSide: BorderSide(
                                                color:Colors.white,//this has no
effect
                                              ),
                                              borderRadius:
BorderRadius.circular(28.0),
                                            ),
                                            // border: InputBorder.none,
                                            hintText:'Your name ("John Smith")',
                                            hintStyle: TextStyle(fontSize: 12.0,
color:Colors.grey),
//                  contentPadding: EdgeInsetsDirectional.only(start: 10),
                                          ),

                                          keyboardType:
TextInputType.emailAddress,
                                        )
                                    ),
                                    Container(height: ThemeDimens.padding32),

                                      RaisedButton(
                                        onPressed:
                                            () async {
                                          if (_validateAndSave()) {
                                            print("reg");
                                          await _register();

MainNavigatorWidget.of(context).goToStartJourneyScreen();
```

```
                                          }


                                        },
                                        shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(80.0)),
                                        padding: EdgeInsets.all(0.0),
                                        child: Ink(
                                          decoration: BoxDecoration(
                                              gradient: LinearGradient(colors:
[Colors.black,Colors.black],
//                  Color(0xFF018151)
                                                begin: Alignment.centerLeft,
                                                end: Alignment.centerRight,
                                              ),
                                              borderRadius:
BorderRadius.circular(30.0)
                                          ),
                                          child: Container(
                                            constraints: BoxConstraints(maxWidth:
MediaQuery.of(context).size.width/1.5, minHeight: 50.0),
                                            alignment: Alignment.center,
                                            child: Text(
                                              "CREATE ACCOUNT",
                                              textAlign: TextAlign.center,

style:theme.lightTextTheme.titleListItem
                                            ),
                                          ),
                                        ),
//      ),
                                      ),

                                    Container(height: ThemeDimens.padding32),
                                    InkWell(
                                      splashColor: Colors.yellow,
                                      highlightColor: Colors.blue,
                                      child: Text("Sign
in",style:theme.lightTextTheme.bodyNormal),
                                      onTap: () {

MainNavigatorWidget.of(context).goToStridyLoginScreen();

                                      },
                                    ),
                                    Container(height: ThemeDimens.padding32),
                                    InkWell(
                                      splashColor: Colors.yellow,
                                      highlightColor: Colors.blue,
                                      child: Text("privacy+ terms of
use",style:theme.lightTextTheme.bodyNormal),
                                      onTap: () {
                                        //
MainNavigatorWidget.of(context).goToSignUpScreen();

                                      },
                                    )
                                  ],
                              ),


                            ],
                          ),
                        ))
                    ],
                  ),
                ),
              ),
              )
            ],)

          ),
        ),
      ),
    );
  }

  @override
  void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
```

6.4 FORGET PASSWORD SCREEN

```
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';
import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';

class PasswordScreen extends StatefulWidget {
  static const String routeName = 'password';

  const PasswordScreen({Key? key}) : super(key: key);

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

@visibleForTesting
class PasswordScreenState extends State<PasswordScreen> with ErrorNavigatorMixin
implements LoginNavigator {
  final _formKey = new GlobalKey<FormState>();
  bool _autoValidate = false;
  bool emailSubmited = false;
  String _email='';
  bool _obscureText = true;
  FocusNode focusNode1 = FocusNode();
  FocusNode focusNode2 = FocusNode();
  final passwordController = TextEditingController();
  final emailController = TextEditingController();
  String validateEmail(String value) {
    String pattern =
r'^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-
Z]{2,}))$';
    RegExp regExp = new RegExp(pattern);
    if (value.length > 0 && !regExp.hasMatch(value)&& !emailSubmited) {
      return "Invalid Email";
    }
    else if(value.isEmpty)
      return "Email Field Can not be Empty";
    else {
      return "null";
    }
  }
  @override
  Widget build(BuildContext context) {

    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            backgroundColor: theme.colorsTheme.secondary,
            body:
            Stack(children: [
              Container(
                decoration: BoxDecoration(
                    gradient: LinearGradient(
                        begin: Alignment.topRight,
                        end: Alignment.bottomLeft,
                        colors:
[theme.colorsTheme.lightGreen,theme.colorsTheme.darkGreen])),
              ),
              SingleChildScrollView(child:
              Center(
                child: Padding(
                  padding: const EdgeInsets.all(8.0),
                  child: Column(crossAxisAlignment: CrossAxisAlignment.center,
                    // mainAxisAlignment: MainAxisAlignment.center,
                    children: [
                      Container(height: ThemeDimens.padding16),
                      Container(

                          child: Form(
                            key: _formKey,
                            autovalidate: _autoValidate,
                            child: Column(
                              // shrinkWrap: true,
                              children: <Widget>[
                                Column(crossAxisAlignment:
CrossAxisAlignment.center,
                                  children: [
                                    Container(height: ThemeDimens.padding32),
                                    Icon(Icons.spa,size: 40,),
                                    Container(
                                      child:  Text(
                                        'ZiGreenCity',
                                        style:
```

```
theme.lightTextTheme.titleHugeStridy,
                                          ),),
                                        Container(height: ThemeDimens.padding32),
                                        Center(child: Text("Enter your email address
below and \nwe'll send you a link to reset your password.",)),
                                        Container(height: ThemeDimens.padding32),
                                        Container(height: ThemeDimens.padding32),
                                        Container(
                                            width:
MediaQuery.of(context).size.width/1.5,
                                            height: 55.0,
                                            //padding:
EdgeInsets.only(left:width*0.04, right: width*0.04),
                                            child: TextFormField(

                                              focusNode: focusNode1,
                                              controller: emailController,
                                              onFieldSubmitted: (term) {
                                                focusNode1.unfocus();

FocusScope.of(context).requestFocus(focusNode2);
                                              },
                                              onChanged: (content) {
                                                setState(() {
                                                  // enabled = true;
                                                });

                                              },
                                              autofocus: true,

                                              // validator:validateEmail,
//                  (value) => value.isEmpty ? 'Email can\'t be empty' : null,
//                                             onSaved: (value) => _email =
value.trim(),
                                              cursorColor:Colors.lightGreen,
                                              decoration: InputDecoration(
                                                contentPadding: EdgeInsets.only(left:
11, right: 3, top: 14, bottom: 14),
                                                errorStyle: TextStyle(fontSize: 11,
height: 0.3),
                                                filled: true,
                                                fillColor: Colors.white,
                                                enabledBorder: OutlineInputBorder(
                                                  borderSide: BorderSide(
                                                    color:Colors.white,
                                                  ),
                                                  borderRadius:
BorderRadius.circular(28.0),
                                                ),
                                                focusedBorder: OutlineInputBorder(
                                                  borderSide: BorderSide(
                                                    color:Colors.white,
                                                  ),
                                                  borderRadius:
BorderRadius.circular(28.0),
                                                ),
                                                border: OutlineInputBorder(
                                                  borderSide: BorderSide(
                                                    color:Colors.grey,//this has no
effect
                                                  ),
                                                  borderRadius:
BorderRadius.circular(28.0),
                                                ),
                                                // border: InputBorder.none,
                                                hintText:"e.g.example@email.com",
                                                hintStyle: TextStyle(fontSize: 12.0,
color:Colors.grey),
//                 contentPadding: EdgeInsetsDirectional.only(start: 10),
                                              ),
                                              keyboardType:
TextInputType.emailAddress,
                                            )
                                        ),
                                        Container(height: ThemeDimens.padding32),

                                        Container(height: ThemeDimens.padding32),

                                        Container(height: ThemeDimens.padding32),
                                        AbsorbPointer(
                                          // absorbing: !enabled,
                                          child:
                                          RaisedButton(
                                            onPressed:
                                                () async {



MainNavigatorWidget.of(context).goToStartJourneyScreen();



                                            },
                                            shape:
RoundedRectangleBorder(borderRadius: BorderRadius.circular(80.0)),
                                            padding: EdgeInsets.all(0.0),
                                            child: Ink(
                                              decoration: BoxDecoration(
                                                  gradient: LinearGradient(colors:
[Colors.black,Colors.black],
//                 Color(0xFF018151)
                                                    begin: Alignment.centerLeft,
                                                    end: Alignment.centerRight,
                                                  ),
                                                  borderRadius:
BorderRadius.circular(30.0)
                                              ),
                                              child: Container(
                                                constraints: BoxConstraints(maxWidth:
MediaQuery.of(context).size.width/1.5, minHeight: 50.0),
                                                alignment: Alignment.center,
                                                child: Text(
                                                    "SEND EMAIL",
                                                    textAlign: TextAlign.center,

style:theme.lightTextTheme.titleListItem
                                                ),
                                              ),
                                            ),
//      ),
                                          ),
                                        ),
                                        Container(height: ThemeDimens.padding32),
                                        InkWell(
                                          splashColor: Colors.yellow,
                                          highlightColor: Colors.blue,
                                          child: Text("Go
Back",style:theme.lightTextTheme.bodyNormal),
                                          onTap: () {

MainNavigatorWidget.of(context).goToStridyLoginScreen();

                                          },
                                        ),

                                      ],

                                    ),


                                  ],
                                ),
                              ))
                            ],
                          ),
                        ),
                      ),
                    )
                  ],)

                ),
              ),
            ),
          );
      }

      @override
      void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
```

6.5 PROFILE SCREEN

```
import 'dart:math';

import
'package:flutter_template/widget/general/styled/flutter_template_progress_indicato
r.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:flutter_template/util/keys.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_button.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_input_field.dart'
;
import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';

class UserProfileScreen extends StatefulWidget {
  static const String routeName = 'userProfile';

  const UserProfileScreen({Key? key}) : super(key: key);

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

@visibleForTesting
class UserProfileScreenState extends State<UserProfileScreen> with
ErrorNavigatorMixin implements LoginNavigator {
  final _formKey = new GlobalKey<FormState>();
  bool _autoValidate = false;
  bool emailSubmited = false;
  String _email='';
  bool _obscureText = true;
  FocusNode focusNode1 = FocusNode();
  FocusNode focusNode2 = FocusNode();
  final passwordController = TextEditingController();
  final emailController = TextEditingController();

  @override
  Widget build(BuildContext context) {

    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            backgroundColor: theme.colorsTheme.secondary,
            body:
            Container(
              color: Colors.white,
              child: Column(
                children: [

                  ClipPath(
                    clipper:CurvedBottomClipper(),
                    child: Container(
                      decoration: BoxDecoration(
                          gradient: LinearGradient(
                              begin: Alignment.topRight,
                              end: Alignment.bottomLeft,
                              colors:
[theme.colorsTheme.lightGreen,theme.colorsTheme.darkGreen])),
                      height: 400.0,
                      child: Center(
                          child: Padding(
                            padding: EdgeInsets.only(bottom: 50),
                            child: Column(
                              children: [
                                Container(height: ThemeDimens.padding32),
                                Container(height: ThemeDimens.padding32),
                                Row(crossAxisAlignment: CrossAxisAlignment.start,
                                  mainAxisAlignment:
MainAxisAlignment.spaceAround,
                                  children: [
                                    Container(
                                      decoration: new BoxDecoration(

                                        color:  Colors.white,
                                        shape: BoxShape.circle,
                                        // border: new Border.all(
                                        //   color: Colors.green,
                                        //   width: 2.5,
                                        // ),
                                      ),
                                      // color:  Colors.white,
                                      child: Padding(
                                        padding: const EdgeInsets.all(8.0),
                                        child: Center(
                                          child:Icon(Icons.arrow_back_ios),
                                        ),
                                      ),
                                    ),
                                    Text(
                                      'profile',
                                        style:theme.lightTextTheme.labelButtonBig
                                    ),

                                    Container(
                                      decoration: new BoxDecoration(

                                        color:  Colors.white,
                                        shape: BoxShape.circle,
                                        // border: new Border.all(
                                        //   color: Colors.green,
                                        //   width: 2.5,
                                        // ),
                                      ),
                                      // color:  Colors.white,
                                      child: Padding(
                                        padding: const EdgeInsets.all(8.0),
                                        child: Center(
                                          child:IconButton(
                                            icon: Icon(Icons.settings),
                                            onPressed: (){
MainNavigatorWidget.of(context).goToSettingscreen();
                                              },
                                          )
                                        ),
                                      ),
                                    ),
                                  ],
                                ),
                                Container(height: ThemeDimens.padding32),
                                Container(
                                  decoration: new BoxDecoration(

                                    color:theme.colorsTheme.darkGreen,
                                    shape: BoxShape.circle,
                                    border: new Border.all(
                                      color: Colors.lime,
                                      width: 2.5,
                                    ),
                                  ),
                                  child:  Padding(
                                    padding: const EdgeInsets.all(46.0),
                                    child: Container(
                                      decoration: new BoxDecoration(

                                        color:  Colors.white,
                                        shape: BoxShape.circle,
                                        // border: new Border.all(
                                        //   color: Colors.green,
                                        //   width: 2.5,
                                        // ),
                                      ),
                                      // color:  Colors.white,
                                      child: new Center(
                                        child:Icon(Icons.face),
                                      ),
                                    ),
                                  ),
                                ),
                                Container(height: ThemeDimens.padding8),
                                Text("user
name",style:theme.lightTextTheme.bodyBig),
                                // Container(height: ThemeDimens.padding4),
                                Text("starting
strider",style:theme.lightTextTheme.bodySmall),
                                Container(height: ThemeDimens.padding8),
                                Text("0",style:theme.lightTextTheme.bodyBig),

                              ],
                            ),
                          )),
                      ),
                    ),
                    Container(height: ThemeDimens.padding32),
                    Container(height: ThemeDimens.padding32),
                    Text("nothing to see here",style:theme.lightTextTheme.bodyBig),
                    Container(height: ThemeDimens.padding32),
                    Padding(
                      padding: const EdgeInsets.all(8.0),
                      child: RaisedButton(
                        onPressed:
                            () {

MainNavigatorWidget.of(context).goToEndJourneyScreen();
                        },
                        shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(80.0)),
                        padding: EdgeInsets.all(0.0),
                        child: Ink(
                          decoration: BoxDecoration(
                            gradient: LinearGradient(colors:
[Colors.white,Colors.white],
//                  Color(0xFF018151)
                              begin: Alignment.centerLeft,
                              end: Alignment.centerRight,
                            ),
                            borderRadius: BorderRadius.circular(30.0),
                          boxShadow: [
                            BoxShadow(
                              color: Colors.grey,
                              blurRadius: 1.0,
                            ),
                          ],
                          ),
                          child: Container(
                          constraints: BoxConstraints(maxWidth:
MediaQuery.of(context).size.width/1.5, minHeight: 80.0),
                          alignment: Alignment.center,
                          child: Text(
                              "START YOUR JOURNEY NOW",
                              textAlign: TextAlign.center,
                              style:theme.darkTextTheme.titleListItem
                          ),
                          ),
                        ),
//      ),
                      ),
                    ),

                  ],
                ),
              ),

            ),
          ),
        ),
      );
  }

  @override
  void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
class CurvedBottomClipper extends CustomClipper<Path> {
  @override
  Path getClip(Size size) {
    // I've taken approximate height of curved part of view
    // Change it if you have exact spec for it
    final roundingHeight = size.height * 3 / 5;

    // this is top part of path, rectangle without any rounding
    final filledRectangle =
    Rect.fromLTRB( size.width,  size.height - roundingHeight,0,0);

    // this is rectangle that will be used to draw arc
    // arc is drawn from center of this rectangle, so it's height has to be twice
roundingHeight
```

```
    // also I made it to go 5 units out of screen on left and right, so curve will
have some incline there
    final roundingRectangle = Rect.fromLTRB(
        -5, size.height - roundingHeight * 2, size.width + 5, size.height);

    final path = Path();
    path.addRect(filledRectangle);

    // so as I wrote before: arc is drawn from center of roundingRectangle
    // 2nd and 3rd arguments are angles from center to arc start and end points
    // 4th argument is set to true to move path to rectangle center, so we don't
have to move it manually
    path.arcTo(roundingRectangle, pi, -pi, true);
    path.close();

    return path;
  }

  @override
  bool shouldReclip(CustomClipper<Path> oldClipper) {
    // returning fixed 'true' value here for simplicity, it's not the part of
actual question, please read docs if you want to dig into it
    // basically that means that clipping will be redrawn on any changes
    return true;
  }
}

class RoundedClipper extends CustomClipper<Path> {
  @override
  Path getClip(Size size) {
    var path = Path();
    path.lineTo(0.0, size.height);
    path.quadraticBezierTo(
        size.width / 2,
        size.height - 100,
        size.width,
        size.height
    );
    path.lineTo(size.width, 0.0);
    path.close();
    return path;
  }

  @override
  bool shouldReclip(CustomClipper<Path> oldClipper) => false;
}
```

6.5 JOURNEY SCREEN

i. Start Journey

```
import 'dart:math';

import
'package:flutter_template/widget/general/styled/flutter_template_progress_indicato
r.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:flutter_template/util/keys.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_button.dart';
import
'package:flutter_template/widget/general/styled/flutter_template_input_field.dart'
;
import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';

class StartJourneyScreen extends StatefulWidget {
  static const String routeName = 'startJourney';

  const StartJourneyScreen({Key? key}) : super(key: key);

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

@visibleForTesting
class StartJourneyScreenState extends State<StartJourneyScreen> with
ErrorNavigatorMixin implements LoginNavigator {
  final _formKey = new GlobalKey<FormState>();
  bool _autoValidate = false;
  bool emailSubmited = false;
  String _email='';
  bool _obscureText = true;
  FocusNode focusNode1 = FocusNode();
  FocusNode focusNode2 = FocusNode();
  final passwordController = TextEditingController();
  final emailController = TextEditingController();

  @override
  Widget build(BuildContext context) {

    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            backgroundColor: theme.colorsTheme.secondary,
            body:
            Container(
              decoration: BoxDecoration(
                  gradient: LinearGradient(
                      begin: Alignment.topRight,
                      end: Alignment.bottomLeft,
                      colors:
[theme.colorsTheme.lightGreen,theme.colorsTheme.darkGreen])),
              child: Column(
                children: [

                  ClipPath(
                    clipper:RoundedClipper(),
                    child: Container(
                      color: Colors.white,
                      height: 400.0,
                      child: Center(
                          child: Padding(
                            padding: EdgeInsets.only(bottom: 50),
                            child: Column(
                              children: [
                                Container(height: ThemeDimens.padding32),
                                Container(height: ThemeDimens.padding32),
                                Row(crossAxisAlignment: CrossAxisAlignment.start,
                                  mainAxisAlignment:
MainAxisAlignment.spaceAround,
                                  children: [
                                    Column(
                                      children: [
                                        Container(
                                          decoration: BoxDecoration(
                                            color:Colors.grey.withOpacity(0.5),
                                            borderRadius:
BorderRadius.all(Radius.circular(20.0)),

                                          ),
                                          child:Padding(
                                            padding: const EdgeInsets.all(4.0),
                                            child: Row(crossAxisAlignment:
CrossAxisAlignment.start,
                                              children: [

                                                Icon(Icons.arrow_left),

Icon(Icons.supervised_user_circle),

                                              ],
                                            ),
                                          ),

                                        ),
                                        Container(height: ThemeDimens.padding12),
                                        Container(
                                            decoration: BoxDecoration(
                                              color:Color(0xFFFFFFFF),
                                              borderRadius:
BorderRadius.all(Radius.circular(10.0)),
                                              boxShadow: [
                                                BoxShadow(
                                                  color: Colors.grey,
                                                  blurRadius: 1.0,
                                                ),
                                              ],
                                            ),
                                            child:InkWell(onTap: (){


MainNavigatorWidget.of(context).goToUserProfileScreen();
                                            },
                                              child: Padding(
                                                padding: const
EdgeInsets.all(12.0),
                                                child: Text("community"),
                                              ),)

                                        )
                                      ],
                                    ),
                                    Container(
                                      decoration: new BoxDecoration(

                                        color:theme.colorsTheme.lightGreen,
                                        shape: BoxShape.circle,
                                        border: new Border.all(
                                          color: Colors.lime,
                                          width: 2.5,
                                        ),
                                      ),
                                      child:  Padding(
                                        padding: const EdgeInsets.all(46.0),
                                        child: Container(
                                          decoration: new BoxDecoration(

                                            color:  Colors.white,
                                            shape: BoxShape.circle,
                                            // border: new Border.all(
                                            //   color: Colors.green,
                                            //   width: 2.5,
                                            // ),
                                          ),
                                          // color:  Colors.white,
                                          child: new Center(
                                            child:Icon(Icons.face),
                                          ),
                                        ),
                                      ),
                                    ),
                                    Column(
                                      children: [
                                        Container(
                                          decoration: BoxDecoration(
                                            color:Colors.grey.withOpacity(0.5),
                                            borderRadius:
```

```
BorderRadius.all(Radius.circular(20.0)),

                                          ),
                                          child:Padding(
                                            padding: const EdgeInsets.all(4.0),
                                            child: Row(crossAxisAlignment:
CrossAxisAlignment.start,

                                              children: [
                                                Icon(Icons.person),
                                                Icon(Icons.arrow_right),

                                              ],
                                            ),
                                          ),

                                        ),
                                        Container(height: ThemeDimens.padding12),
                                        Container(
                                            decoration: BoxDecoration(
                                              color:Color(0xFFFFFFFF),
                                              borderRadius:
BorderRadius.all(Radius.circular(10.0)),

                                              boxShadow: [
                                                BoxShadow(
                                                  color: Colors.grey,
                                                  blurRadius: 1.0,
                                                ),
                                              ],
                                            ),
                                            child:InkWell(onTap: (){


MainNavigatorWidget.of(context).goToUserProfileScreen();
                                            },
                                              child: Padding(
                                                padding: const
EdgeInsets.all(12.0),

                                                child: Text("user profile"),
                                              ),)

                                        )
                                      ],
                                    ),
                                  ],
                                ),
                                Container(height: ThemeDimens.padding32),
                                Text("ready?", textAlign: TextAlign.center,
                                    style:theme.lightTextTheme.labelButtonBig),
                                Container(height: ThemeDimens.padding32),
                                Text("Celebrate the action you take.",
style:theme.lightTextTheme.bodyBig),
                                Text("No matter how small."
,style:theme.lightTextTheme.bodyBig),

                              ],
                            ),
                          )),
                    ),
                  ),
                  Container(height: ThemeDimens.padding32),
                  Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: RaisedButton(
                      onPressed:
                          () {
MainNavigatorWidget.of(context).goToEndJourneyScreen();
                      },
                      shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(80.0)),
                      padding: EdgeInsets.all(0.0),
                      child: Ink(
                        decoration: BoxDecoration(
                            gradient: LinearGradient(colors:
[Colors.white,Colors.white],
//                  Color(0xFF018151)
                              begin: Alignment.centerLeft,
                              end: Alignment.centerRight,
                            ),
                            borderRadius: BorderRadius.circular(30.0)
                        ),
                        child: Container(
                          constraints: BoxConstraints(maxWidth:
MediaQuery.of(context).size.width/1.5, minHeight:80.0),
                          alignment: Alignment.center,
                          child: Text(
                            "START JOURNEY",
                            textAlign: TextAlign.center,
                              style:theme.lightTextTheme.labelButtonBig
                          ),
                        ),
                      ),
//       ),
                    ),
                  ),
                  Container(height: ThemeDimens.padding32),
                  Text("I have a team code",style:theme.lightTextTheme.bodyBig),
                ],
              ),
            ),

          ),
        ),
      ),
    );
}

@override
void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
class CurvedBottomClipper extends CustomClipper<Path> {
@override
Path getClip(Size size) {
  // I've taken approximate height of curved part of view
  // Change it if you have exact spec for it
  final roundingHeight = size.height * 3 / 5;

  // this is top part of path, rectangle without any rounding
  final filledRectangle =
  Rect.fromLTRB( size.width,  size.height - roundingHeight,0,0);

  // this is rectangle that will be used to draw arc
  // arc is drawn from center of this rectangle, so it's height has to be twice
roundingHeight
  // also I made it to go 5 units out of screen on left and right, so curve will
have some incline there
  final roundingRectangle = Rect.fromLTRB(
        -5, size.height - roundingHeight * 2, size.width + 5, size.height);

  final path = Path();
  path.addRect(filledRectangle);

  // so as I wrote before: arc is drawn from center of roundingRectangle
  // 2nd and 3rd arguments are angles from center to arc start and end points
  // 4th argument is set to true to move path to rectangle center, so we don't
have to move it manually
  path.arcTo(roundingRectangle, pi, -pi, true);
  path.close();

  return path;
}

@override
bool shouldReclip(CustomClipper<Path> oldClipper) {
  // returning fixed 'true' value here for simplicity, it's not the part of
actual question, please read docs if you want to dig into it
  // basically that means that clipping will be redrawn on any changes
  return true;
}
}

class RoundedClipper extends CustomClipper<Path> {
@override
Path getClip(Size size) {
  var path = Path();
  path.lineTo(0.0, size.height);
  path.quadraticBezierTo(
        size.width / 2,
        size.height - 100,
        size.width,
        size.height
    );
  path.lineTo(size.width, 0.0);
  path.close();
  return path;
}

@override
bool shouldReclip(CustomClipper<Path> oldClipper) => false;
}
```

ii. Resume Journey

```
import 'dart:math';

import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';
import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';

class ResumeJourneyScreen extends StatefulWidget {
  static const String routeName = 'resumeJourney';

  const ResumeJourneyScreen({Key? key}) : super(key: key);

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

@visibleForTesting
class ResumeJourneyScreenState extends State<ResumeJourneyScreen> with
ErrorNavigatorMixin, SingleTickerProviderStateMixin implements LoginNavigator {
  late TabController _controller;


double.parse(widget.startLng)),
  //       PointLatLng(double.parse(widget.stopLat),
double.parse(widget.stopLng)),
  //       travelMode: TravelMode.walking,
  //       wayPoints: [PolylineWayPoint(location: "Sabo, Yaba Lagos Nigeria")]);
  //   if (result.points.isNotEmpty) {
  //     result.points.forEach((PointLatLng point) {
  //       polylineCoordinates.add(LatLng(point.latitude, point.longitude));
  //     });
  //   }
  //   _addPolyLine();
  // }



  @override
  int activeTabIndex = 1;
  bool bottle=false;
  bool can=false;
  bool cup=false;
  bool wrapper=false;
  bool mask=false;
  bool other=false;
  bool more=false;
  @override
  void initState() {
    super.initState();
    _controller = TabController(
      length: 2,
      initialIndex: 1,
      vsync: this,
    );
    _controller.addListener(() {
      setState(() {
        activeTabIndex = _controller.index;
      });
    });

    // _addMarker(LatLng(double.parse(originLatitude),
double.parse(widget.startLng)), "origin",
    //     BitmapDescriptor.defaultMarker);
    //
    // /// destination marker
    // _addMarker(LatLng(double.parse(widget.stopLat),
double.parse(widget.stopLng)), "destination",
    //     BitmapDescriptor.defaultMarkerWithHue(90));
    // _getPolyline();
  }


  int _itemCount = 1;
  int _itemCountCan = 1;
  int _itemCountCup = 1;
  @override
  Widget build(BuildContext context) {

    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            // backgroundColor: theme.colorsTheme.secondary,
            appBar:AppBar(
              backgroundColor:theme.colorsTheme.lightGreen,
              elevation: 0,
              title:Center(child: Text('RESUME
JOURNEY',style:theme.lightTextTheme.labelButtonBig)),
              actions: [

              ],
            ),
            bottomNavigationBar:Container(
              height: 50,
              child: Row(crossAxisAlignment: CrossAxisAlignment.start,
                mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [
                  Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: RaisedButton(
                      onPressed:
                          () {
                       //
MainNavigatorWidget.of(context).goToFirebaseStoreScreen();
                      },
                      shape: RoundedRectangleBorder(borderRadius:
```

```
BorderRadius.circular(80.0)),
                      padding: EdgeInsets.all(0.0),
                      child: Ink(
                        decoration: BoxDecoration(
                            gradient: LinearGradient(colors:
[theme.colorsTheme.lightGreen,theme.colorsTheme.darkGreen],
//                  Color(0xFF018151)
                              begin: Alignment.centerLeft,
                              end: Alignment.centerRight,
                            ),
                            borderRadius: BorderRadius.circular(30.0)
                        ),
                        child: Container(
                          constraints: BoxConstraints(maxWidth:
MediaQuery.of(context).size.width/1.5, minHeight: 5.0),
                          alignment: Alignment.center,
                          child: Text(
                              "SAVE & FINISH",
                              textAlign: TextAlign.center,
                              style:theme.lightTextTheme.labelButtonBig
                          ),
                        ),
                      ),
//      ),
                    ),
                  ),
                  Container(
                    decoration: new BoxDecoration(
                      boxShadow: [
                        BoxShadow(
                          color: Colors.grey.withOpacity(0.2),
                          blurRadius: 1.0,
                        ),
                      ],
                      color:  Colors.grey.withOpacity(0.2),
                      shape: BoxShape.circle,
                    ),

                    child: InkWell(
                      onTap: (){
                        MainNavigatorWidget.of(context).goToAddPickupScreen();
                      },
                      child: Padding(
                        padding: const EdgeInsets.all(4.0),
                        child: new Center(
                          child:Icon(Icons.delete),
                        ),
                      ),
                    ),
                  ),
                ],
              ),
            ),
            body:
            SingleChildScrollView(
              child: Container(
                color: Colors.white,
                child: Padding(
                  padding: const EdgeInsets.all(16.0),
                  child: Column(crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      Container(height: ThemeDimens.padding16),
                      Text('results',style:theme.lightTextTheme.titleHugeStridy),
                      Container(height: ThemeDimens.padding32),
                      Text('summary',style:theme.lightTextTheme.labelButtonBig),
                      Container(height: ThemeDimens.padding32),
                      Container(
                        decoration: new BoxDecoration(
                          color:  Colors.white,
                          shape: BoxShape.circle,
                          boxShadow: [
                            BoxShadow(
                              color: Colors.grey.withOpacity(0.1),
                              blurRadius: 2.0,
                            ),
                          ],
                          border: new Border.all(
                            color: Colors.grey,
                            width: 1.5,
                          ),
                        ),
                        child:  Padding(
                          padding: const EdgeInsets.all(50.0),
                          child: Column(
                            children: [

                              Container(height: ThemeDimens.padding4),
                              Row(crossAxisAlignment: CrossAxisAlignment.center,
                                mainAxisAlignment: MainAxisAlignment.center,
                                children: [
                                  Text("+0
",style:theme.lightTextTheme.labelButtonBig),
                                  Icon(Icons.cloud_circle)
                                ],
                              ),
                              Container(height: ThemeDimens.padding4),
                              Text("earned",style:theme.lightTextTheme.bodySmall),

                            ],
                          ),
                        ),
                      ),
                      Container(height: ThemeDimens.padding32),
                      Text("totals",style:theme.lightTextTheme.labelButtonBig),
                      Container(height: ThemeDimens.padding16),
                  Container(
                    child: Column(
                      children: [
                        Container(height: ThemeDimens.padding16),
                        Row(crossAxisAlignment: CrossAxisAlignment.start,
                          mainAxisAlignment: MainAxisAlignment.spaceAround,
                          children: [
                            Column(children: [
                              Text("items"),
                              Text("1",style:theme.lightTextTheme.labelButtonBig),
                            ],),
                            Column(children: [
                              Text("total feet"),
                              Text("1",style:theme.lightTextTheme.labelButtonBig),
                            ],),
                            Column(children: [
                              Text("total time"),
                              Text("1
min",style:theme.lightTextTheme.labelButtonBig),
                            ],),
                          ],
                        ),
                        Visibility(visible: more,
                          child: Column(
                            children: [
                              Divider(color: Colors.grey,),
                              Row(crossAxisAlignment: CrossAxisAlignment.start,
                                mainAxisAlignment: MainAxisAlignment.spaceAround,
                                children: [

                                  Column(children: [

Text("1",style:theme.lightTextTheme.labelButtonBig),
                                    Text("bottle"),
                                  ],),
                                  Column(children: [


Text("1",style:theme.lightTextTheme.labelButtonBig),
                                    Text("can"),
                                  ],),
                                  Column(children: [


Text("1",style:theme.lightTextTheme.labelButtonBig),
                                    Text("cup"),
                                  ],),
                                ],
                              ),
                              Container(height: ThemeDimens.padding8),
                              Row(crossAxisAlignment: CrossAxisAlignment.start,
                                mainAxisAlignment: MainAxisAlignment.spaceAround,
                                children: [
                                  Column(children: [


Text("0",style:theme.lightTextTheme.labelButtonBig),
                                    Text("wrapper"),
                                  ],),
                                  Column(children: [


Text("1",style:theme.lightTextTheme.labelButtonBig),

                                    Text("mask"),
                                  ],),
                                  Column(children: [


Text("1",style:theme.lightTextTheme.labelButtonBig),
                                    Text("other"),
                                  ],),
                                ],
                              ),
                            ],
                          ),
                        ),

                        Divider(color: Colors.grey,),

                        Container(height: ThemeDimens.padding16),
                        InkWell(onTap: (){
                          setState(() {
                            more=!more;
                          });
                        },
                            child: Text(more==true?"See less":"see more")),
                        Container(height: ThemeDimens.padding16),
                      ],
                    ),
                    decoration: BoxDecoration(
                      color:Color(0xFFFFFFFF),
                      borderRadius: BorderRadius.all(Radius.circular(10.0)),
                      boxShadow: [
                        BoxShadow(
                          color: Colors.grey,
                          blurRadius: 1.0,
                        ),
                      ],
                    ),),
                      Container(height: ThemeDimens.padding32),
                      Text("map",style:theme.lightTextTheme.labelButtonBig),
                      Container(height: ThemeDimens.padding32),
                      Text("items",style:theme.lightTextTheme.labelButtonBig)
                    ],
                  ),
                ),
              ),
            ),

          ),
        ),
      ),
    );
  }

  @override
  void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
```

iii. Collected Items

```
import 'dart:io';
import 'dart:math';
import 'package:permission_handler/permission_handler.dart';
import 'package:image_cropper/image_cropper.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';
import 'package:image_picker/image_picker.dart';
import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';
import 'package:flutter/material.dart';
import 'package:flutter_template/widget/general/status_bar.dart';

import 'package:get_it/get_it.dart';
import 'package:provider/provider.dart';

class CollectedItemScreen extends StatefulWidget {
  static const String routeName = 'collectedItem';

  const CollectedItemScreen({Key? key}) : super(key: key);

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

@visibleForTesting
class CollectedItemScreenState extends State<CollectedItemScreen> with
ErrorNavigatorMixin, SingleTickerProviderStateMixin implements LoginNavigator {
  late TabController _controller;
  @override
  int activeTabIndex = 0;
  bool buttonVisible = true;
  bool imageVisible = false;
  bool bottle=false;
  bool can=false;
  bool cup=false;
  bool wrapper=false;
  bool mask=false;
  bool other=false;
  bool small=false;
  bool medium=false;
  bool large=false;
  bool extraL=false;
  bool xxLarge=false;
  // File _image;
  // var _image;
  void open_camera(BuildContext context)
  async {
    await Permission.camera.request();

    //
    // // _image = await ImagePicker.pickImage(source: ImageSource.camera);
    // if (_image != null) {
    //   setState(() {
    //     // _cropImage(context);
    //
    //   });
    // }
    // buttonVisible=false;
    // imageVisible=true;
  }

  // Future<Null> _cropImage(BuildContext context) async {
  //
  //   File croppedFile = await ImageCropper.cropImage(
  @override
  void initState() {
    super.initState();
    _controller = TabController(
      length: 2,
      initialIndex: 1,
      vsync: this,
    );
    _controller.addListener(() {
      setState(() {
        activeTabIndex = _controller.index;
      });
    });
  }



  int _itemCount = 1;
  int _itemCountCan = 1;
  int _itemCountCup = 1;
  @override
  Widget build(BuildContext context) {

    return ProviderWidget<LoginViewModel>(
      create: () => GetIt.I()..init(this),
      childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
        builder: (context, viewModel, child) => StatusBar.light(
          child: Scaffold(
            // backgroundColor: theme.colorsTheme.secondary,
            appBar:AppBar(
              backgroundColor:Colors.white ,
              elevation: 0,
              title:Center(child: Text('collected items',
style:theme.lightTextTheme.labelButtonBig)),

            ),

            body:
            SingleChildScrollView(
              child: Container(
                color: Colors.white,
                child: Column(
                  children: [
                    Container(
                      decoration: new BoxDecoration(color:Colors.white),
                      child: TabBar(
                        controller: _controller,
                        labelColor: Colors.black12,
                        unselectedLabelColor: Colors.black12,
                        indicatorSize: TabBarIndicatorSize.label,
                        indicatorColor: Colors.transparent,
                        tabs: [
                          Tab(
                            child: Container(
                              decoration: BoxDecoration(
                                borderRadius: BorderRadius.horizontal(
                                    left: Radius.circular(50), right:
Radius.zero),
                                border: _controller.index == 1
                                    ? Border.all(color: Colors.black12, width: 2)
                                    : Border.all(color: Colors.transparent, width:
2),

                                color: _controller.index == 0
                                    ? theme.colorsTheme.lightGreen
                                    : Colors.white,
                              ),
                              child: Align(
                                alignment: Alignment.center,
                                child:
Text("TOTALS",style:theme.lightTextTheme.labelButtonBig),
                              ),
                            ),
                          ),
                          Tab(
                            child: Container(
                              decoration: BoxDecoration(
                                borderRadius: BorderRadius.horizontal(
                                    left: Radius.zero, right:
Radius.circular(50)),
                                border: _controller.index == 1
                                    ? Border.all(color: Colors.transparent, width:
2)

                                    : Border.all(color: Colors.black12, width: 2),
                                color: _controller.index == 1
                                    ? theme.colorsTheme.lightGreen
                                    : Colors.white,
                              ),
                              child: Align(
                                alignment: Alignment.center,
                                child: Text("IMAGES",
style:theme.lightTextTheme.labelButtonBig),
                              ),
                            ),
                          ),
                        ],
                      ),
                    ),
                    // Container(height: ThemeDimens.padding32),

                    Padding(
                      padding: const EdgeInsets.all(16.0),
                      child:
                      Container(
                        // decoration: BoxDecoration(
                        //   color:Color(0xFFFFFFFF),
                        //   borderRadius:
BorderRadius.all(Radius.circular(10.0)),
                        //   boxShadow: [
                        //     BoxShadow(
                        //       color: Colors.grey,
                        //       blurRadius: 1.0,
                        //     ),
                        //   ],
                        // ),
                        height:_controller.index == 1? 250.0:250,
                        child:
                        TabBarView(
                          controller: _controller,
                          children: <Widget>[
                            Container(
                              child: Row(crossAxisAlignment:
CrossAxisAlignment.start,
                                // mainAxisAlignment:
MainAxisAlignment.spaceAround,
                                children: [
                                  Stack(
                                    alignment: Alignment.topRight,
                                    children: [
                                      InkWell(
                                        onTap: (){
                                          setState(() {

                                          });
                                        },
                                        child: Container(
                                          decoration: new BoxDecoration(

                                            color:  Colors.grey.withOpacity(0.5),
                                            shape: BoxShape.circle,
                                            border: new Border.all(
                                              color:
can==true?theme.colorsTheme.lightGreen:Colors.grey.withOpacity(0.5),
                                              width: 2.5,
                                            ),
                                          ),
                                          // color:  Colors.white,
                                          child: Padding(
                                            padding: const EdgeInsets.all(8.0),
                                            child: new Center(
                                              child:Icon(Icons.wallet_giftcard),
                                            ),
                                          ),
                                        ),
                                      ),
                                      Padding(
                                        padding: const EdgeInsets.only(left:10.0),
                                        child: Container(

                                          decoration: BoxDecoration(
                                            border: Border.all(width:
1,color:theme.colorsTheme.lightGreen),

                                            shape: BoxShape.circle,
                                            // You can use like this way or like
the below line

                                            //borderRadius: new
BorderRadius.circular(30.0),

                                            color: theme.colorsTheme.lightGreen,
                                          ),
                                          child:

                                          Padding(
                                            padding: const EdgeInsets.all(4.0),
                                            child: Center(child:
Text('$_itemCount')),
                                          ),
                                        ),
                                      ),
                                    ],
                                  ),
                                  SizedBox(width: 5,),
                                  Stack(
                                    alignment: Alignment.topRight,
                                    children: [
                                      InkWell(
                                        onTap: (){
                                          setState(() {
                                            can=!can;

                                          });
                                        },
                                        child: Container(
                                          decoration: new BoxDecoration(

                                            color:  Colors.grey.withOpacity(0.5),
                                            shape: BoxShape.circle,
                                            border: new Border.all(
                                              color:
can==true?theme.colorsTheme.lightGreen:Colors.grey.withOpacity(0.5),
                                              width: 2.5,
                                            ),
                                          ),
                                          // color:  Colors.white,
                                          child: Padding(
                                            padding: const EdgeInsets.all(8.0),
                                            child: new Center(
                                              child:Icon(Icons.wallet_giftcard),
                                            ),
                                          ),
                                        ),
                                      ),
                                      Padding(
                                        padding: const EdgeInsets.only(left:10.0),
                                        child: Container(

                                          decoration: BoxDecoration(
                                            border: Border.all(width:
1,color:theme.colorsTheme.lightGreen),

                                            shape: BoxShape.circle,
                                            // You can use like this way or like
the below line

                                            //borderRadius: new
BorderRadius.circular(30.0),

                                            color: theme.colorsTheme.lightGreen,
                                          ),
                                          child:

                                          Padding(
                                            padding: const EdgeInsets.all(4.0),
                                            child: Center(child:
Text('$_itemCount')),

                                          ),

                                        ),
                                                  ),
                                                ],
                                              ),
                                            ],
                                          ),
                                        ),
                                        Container(
                                          child: Row(crossAxisAlignment:
CrossAxisAlignment.start,
                                            // mainAxisAlignment:
MainAxisAlignment.spaceAround,
                                            children: [
                                              Stack(
                                                alignment: Alignment.topRight,
                                                children: [
                                                  InkWell(
                                                    onTap: (){
                                                      setState(() {

                                                      });
                                                    },
                                                    child: Container(
                                                      decoration: new BoxDecoration(

                                                        color:  Colors.grey.withOpacity(0.5),
                                                        shape: BoxShape.circle,
                                                        border: new Border.all(
                                                          color:
can==true?theme.colorsTheme.lightGreen:Colors.grey.withOpacity(0.5),
                                                          width: 2.5,
                                                        ),
                                                      ),
                                                      // color:  Colors.white,
                                                      child: Padding(
                                                        padding: const EdgeInsets.all(8.0),
                                                        child: new Center(
                                                          child:Icon(Icons.wallet_giftcard),
                                                        ),
                                                      ),
                                                    ),
                                                  ),
                                                  Padding(
                                                    padding: const EdgeInsets.only(left:10.0),
                                                    child: Container(

                                                      decoration: BoxDecoration(
                                                        border: Border.all(width:
1,color:theme.colorsTheme.lightGreen),

                                                        shape: BoxShape.circle,
                                                        // You can use like this way or like
the below line

                                                        //borderRadius: new
BorderRadius.circular(30.0),

                                                        color: theme.colorsTheme.lightGreen,
                                                      ),
                                                      child:

                                                      Padding(
                                                        padding: const EdgeInsets.all(4.0),
                                                        child: Center(child:
Text('$_itemCount')),
                                                      ),

                                                    ),
                                                  ),
                                                ],
                                              ),
                                              SizedBox(width: 5,),
                                              Stack(
                                                alignment: Alignment.topRight,
                                                children: [
                                                  InkWell(
                                                    onTap: (){
                                                      setState(() {
                                                        can=!can;

                                                      });
                                                    },
                                                    child: Container(
                                                      decoration: new BoxDecoration(

                                                        color:  Colors.grey.withOpacity(0.5),
                                                        shape: BoxShape.circle,
                                                        border: new Border.all(
                                                          color:
can==true?theme.colorsTheme.lightGreen:Colors.grey.withOpacity(0.5),
                                                          width: 2.5,
                                                        ),
                                                      ),
                                                      // color:  Colors.white,
                                                      child: Padding(
                                                        padding: const EdgeInsets.all(8.0),
                                                        child: new Center(
                                                          child:Icon(Icons.wallet_giftcard),
                                                        ),
                                                      ),
                                                    ),
                                                  ),
                                                  Padding(
                                                    padding: const EdgeInsets.only(left:10.0),
                                                    child: Container(

                                                      decoration: BoxDecoration(
                                                        border: Border.all(width:
1,color:theme.colorsTheme.lightGreen),

                                                        shape: BoxShape.circle,
                                                        // You can use like this way or like
the below line

                                                        //borderRadius: new
BorderRadius.circular(30.0),

                                                        color: theme.colorsTheme.lightGreen,
                                                      ),
                                                      child:

                                                      Padding(
                                                        padding: const EdgeInsets.all(4.0),
                                                        child: Center(child:
Text('$_itemCount')),
                                                        ),

                                                      ),
                                                    ),
                                                  ],
                                                ),
                                              ],
                                            ),
                                          ),
                                        ],
                                      ),
                                    ),
                                  ),
                                  Container(height: ThemeDimens.padding32),

                                  Container(height: ThemeDimens.padding32),


                                ],
                              ),
                            ),
                          ),

                        ),
                      ),
                    ),
                  );
                }

                @override
                void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
```

iv. End Journey

```
import 'dart:async';
import 'package:flutter/material.dart';
import 'package:carp_background_location/carp_background_location.dart';
import 'package:flutter_template/database/database_table.dart';
import 'package:flutter_template/widget/provider/provider_widget.dart';
import 'package:flutter_template/navigator/main_navigator.dart';
import 'package:flutter_template/styles/theme_dimens.dart';

import 'package:flutter_template/navigator/mixin/error_navigator.dart';
import 'package:flutter_template/viewmodel/login/login_viewmodel.dart';

import 'package:flutter_template/widget/general/status_bar.dart';
import 'package:pedometer/pedometer.dart';
import 'package:get_it/get_it.dart';
import 'package:permission_handler/permission_handler.dart';
import 'package:provider/provider.dart';
enum LocationStatus { UNKNOWN, RUNNING, STOPPED }
String formatDate(DateTime d) {
  return d.toString().substring(0, 19);
}
```

```
String dtoToString(LocationDto dto) =>
    'Location ${dto.latitude}, ${dto.longitude} at
${DateTime.fromMillisecondsSinceEpoch(dto.time ~/ 1)}';

Widget dtoWidget(LocationDto dto) {
  if (dto == null)
    return Text("No location yet");
  else
    return Column(
      children: <Widget>[
        Text(
          '${dto.latitude}, ${dto.longitude}',
        ),
        Text(
          '@',
        ),
        Text('${DateTime.fromMillisecondsSinceEpoch(dto.time ~/ 1)}')
      ],
    );
}
class EndJourneyScreen extends StatefulWidget {
  static const String routeName = 'endJourney';

  const EndJourneyScreen({Key? key}) : super(key: key);

  @override
  EndJourneyScreenState createState() => EndJourneyScreenState();
}
String formatTime(int milliseconds) {
  var secs = milliseconds ~/ 1000;
  var hours = (secs ~/ 3600).toString().padLeft(2, '0');
  var minutes = ((secs % 3600) ~/ 60).toString().padLeft(2, '0');
  var seconds = (secs % 60).toString().padLeft(2, '0');

  return "$hours:$minutes:$seconds";
}

@visibleForTesting
class EndJourneyScreenState extends State<EndJourneyScreen> with
ErrorNavigatorMixin implements LoginNavigator {
  late Stopwatch _stopwatch;
  late Timer _timer;
  // List<String> db = [];
 late List db;
  String logStr = '';
  LocationDto? lastLocation;
  DateTime? lastTimeLocation;
  Stream<LocationDto>? locationStream;
  StreamSubscription<LocationDto>? locationSubscription;
  LocationStatus _status = LocationStatus.UNKNOWN;
late DateTime startJourneyTime;
  late DateTime endJourneyTime;
  late Stream<StepCount> _stepCountStream;
  late Stream<PedestrianStatus> _pedestrianStatusStream;
  String _status1 = '?';
    late  int _steps;

  @override
  void initState() {
    super.initState();

    // Subscribe to stream in case it is already running
    LocationManager().interval = 1;
    LocationManager().distanceFilter = 0;
    LocationManager().notificationTitle = 'CARP Location Example';
    LocationManager().notificationMsg = 'CARP is tracking your location';
    locationStream = LocationManager().locationStream;
    locationSubscription = locationStream!.listen(onData);
    _stopwatch = Stopwatch();
    _stopwatch.start();
    startJourneyTime= DateTime.now();
    initPlatformState();
    _timer = new Timer.periodic(new Duration(milliseconds: 30), (timer) {
      setState(() {});
    });
    LocalDatabase().initDB();

    LocalDatabase().getSteps().then((v) => print(v));
  }

  void onGetCurrentLocation() async {
    LocationDto dto = await LocationManager().getCurrentLocation();
    print('Current location: $dto');
  }

  void onData(LocationDto dto) {
    print(dtoToString(dto));
    setState(() {
      if (_status == LocationStatus.UNKNOWN) {
        _status = LocationStatus.RUNNING;
      }
      lastLocation = dto;
      lastTimeLocation = DateTime.now();
    });
    LocalDatabase().addSteps(_steps,_status1,dto.latitude,dto.longitude);
  }

  void start() async {
    // Subscribe if it hasn't been done already
    if (locationSubscription != null) {
      locationSubscription!.cancel();
    }
    locationSubscription = locationStream!.listen(onData);
    await LocationManager().start();
    setState(() {
      _status = LocationStatus.RUNNING;
    });
  }

  void stop() async {
    setState(() {
      _status = LocationStatus.STOPPED;
    });
    locationSubscription!.cancel();
    await LocationManager().stop();
  }

  Widget stopButton() {
```

```
  Function f = stop;
  String msg = 'STOP';

  return SizedBox(
    width: double.maxFinite,
    child: RaisedButton(
      child: Text(msg),
      onPressed: (){
        stop();
      },
    ),
  );
}

Widget startButton() {
  Function f = start;
  String msg = 'START';
  return SizedBox(
    width: double.maxFinite,
    child: RaisedButton(
      child: Text(msg),
      onPressed:  (){
        start();
      },
    ),
  );
}

Widget status() {
  String msg = _status.toString().split('.').last;
  return Text("Status: $msg");
}

Widget lastLoc() {
  return Text(
      lastLocation != null
          ? dtoToString(lastLocation!)
          : 'Unknown last location',
      textAlign: TextAlign.center);
}

Widget getButton() {
  return RaisedButton(
    child: Text("Get Current Location"),
    onPressed: onGetCurrentLocation,
  );
}



void onStepCount(StepCount event) {
  print(event);
  setState(() {
    _steps = event.steps;
  });
}

void onPedestrianStatusChanged(PedestrianStatus event) {
  print(event);
  setState(() {
    _status1 = event.status;
  });
}

// void onPedestrianStatusError(error) {
//   // print('onPedestrianStatusError: $error');
//   setState(() {
//     _status1 = 'Pedestrian Status not available';
//   });
//   print(_status1);
// }
//
// void onStepCountError(error) {
//   // print('onStepCountError: $error');
//   setState(() {
//     _steps = 'Step Count not available';
//   });
// }

void initPlatformState()async {
  if (await Permission.activityRecognition.request ().isGranted){
  _pedestrianStatusStream = Pedometer.pedestrianStatusStream;
  _pedestrianStatusStream.listen(onPedestrianStatusChanged);


  _stepCountStream = Pedometer.stepCountStream;
  _stepCountStream.listen(onStepCount);
  }
  if (!mounted) return;
}

@override
void dispose() {
  _timer.cancel();
  super.dispose();
}
void handleStartStop() {
  if (_stopwatch.isRunning) {
    startJourneyTime= DateTime.now();
    _stopwatch.stop();
  }
  // else {
  //   _stopwatch.start();
  // }

  setState(() {});
}

@override
Widget build(BuildContext context) {
  LocalDatabase().getSteps().then((v) => setState(() => db = v));
  return ProviderWidget<LoginViewModel>(
    create: () => GetIt.I()..init(this),
    childBuilder: (context, theme, _) => Consumer<LoginViewModel>(
      builder: (context, viewModel, child) => StatusBar.light(
        child: Scaffold(
          backgroundColor: theme.colorsTheme.secondary,
          bottomNavigationBar:Container(
              height: 100,
              width:  MediaQuery.of(context).size.width,
              decoration: BoxDecoration(
                borderRadius: BorderRadius.vertical(
                    top: Radius.circular(50), bottom: Radius.zero),
                border: Border.all(color: Colors.white, width: 2),


                color: Colors.white,
              ),
              child: Padding(
                padding: const EdgeInsets.all(20.0),
                child: InkWell(
                    onTap: handleStartStop,
                    child: Text(_stopwatch.isRunning ? 'END JOURNEY' : 'END
JOURNEY',textAlign: TextAlign.center,style:theme.lightTextTheme.labelButtonBig)),
              ),
            ),
            body:
            db.isEmpty ? Container() : Container(
              child: ListView.builder(
                  itemCount: db.length,
                  itemBuilder: (BuildContext context, int index) {
                    return
                      Card(

                        color: Colors.grey.shade300,
                        child: Padding(
                          padding: const EdgeInsets.all(8.0),
                          child: ListTile(

                            title: InkWell(onTap:(){

                            },
                                child: Text(db[index].toString())),

                          ),
                        ),
                      );
                    // Text(dbSteps[index].toString());
                  }
              ),
            ),

          ),
        ),
      ),
    );
  }

  @override
  void goToHome() => MainNavigatorWidget.of(context).goToHome();
}
```

6.6 SETTINGS SCREEN

```
import 'package:flutter/material.dart';

class Settingsscreen extends StatefulWidget {
  static const String routeName = 'settings';

  const Settingsscreen({Key? key}) : super(key: key);

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

@visibleForTesting
class SettingsscreenState extends State<Settingsscreen>{
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        padding: EdgeInsets.only(left: 16, top: 25, right: 16),
        child: GestureDetector(
          onTap: () {
            FocusScope.of(context).unfocus();
          },
          child: ListView(
            children: [
              // backgroundColor: Theme.of(context).scaffoldBackgroundColor,
              // elevation: 1,
              Row(
                mainAxisAlignment: MainAxisAlignment.start,
                children: [
                  IconButton(
                    icon: Icon(
                      Icons.arrow_back_ios,
                      color: Colors.green,
                    ),
                    onPressed: () {},
                  ),


                  Center(
                    child: Text(
                      "             Settings",
                      style: TextStyle(fontSize: 25, fontWeight: FontWeight.w500),
                    ),),],),
              SizedBox(
                height: 15,
              ),
              /*Center(
                child: Stack(
                  children: [
                    Container(
                      width: 130,
                      height: 130,
                      decoration: BoxDecoration(
                          border: Border.all(
                              width: 4,
                              color: Theme.of(context).scaffoldBackgroundColor),
                          boxShadow: [
                            BoxShadow(
                                spreadRadius: 2,
                                blurRadius: 10,
                                color: Colors.black.withOpacity(0.1),
                                offset: Offset(0, 10))
                          ],
                          //shape: BoxShape.circle,
                         // image: DecorationImage(
                             // fit: BoxFit.cover,
                             // image: NetworkImage(
                               //
"https://images.pexels.com/photos/3307758/pexels-photo-
3307758.jpeg?auto=compress&cs=tinysrgb&dpr=3&h=250",
                              )),
                        // ),
                   // ),
                   /* Positioned(
                        bottom: 0,
                        right: 0,
                        child: Container(
                          height: 40,
                          width: 40,
                          decoration: BoxDecoration(
                            shape: BoxShape.circle,
                            border: Border.all(
                              width: 4,
                              color: Theme.of(context).scaffoldBackgroundColor,
                            ),
                            color: Colors.green,
                          ),
                          child: Icon(
                            Icons.edit,
                            color: Colors.white,
                          ),
                        )),*/
                  ],
                ),
              ),*/
              SizedBox(
                height: 35,
              ),
              buildTextField('name', "Priyam Srivastava", false),
              buildTextField("e-mail", "priyam@gmail.com", false),
              buildTextField("password", "********", true),
              buildTextField("teams", " ",false),
              buildTextField("data", " ",false),

              SizedBox(
                height: 35,
              ),
              // Row
              Column(
                crossAxisAlignment: CrossAxisAlignment.start,
                children: [
                  new InkWell(
                    child: new Text('Privacy Policy',
                      style: TextStyle(
                        decoration: TextDecoration.underline,
                      ),),
                   // onTap: () =>
launch('https://docs.flutter.io/flutter/services/UrlLauncher-class.html')
                  ),
                  SizedBox(
                      height: 10
                  ),
                  new InkWell(
                    child: new Text('Logout',
                      style: TextStyle(
                        decoration: TextDecoration.underline,
                      ),
                    ),

                    //onTap: () =>
launch('https://docs.flutter.io/flutter/services/UrlLauncher-class.html')
                  ),

                  SizedBox(
                      height: 20
                  ),
                  Text(
                      "app version: Stridy v1.3.0-14"
                  ),
                  SizedBox(
                      height: 10
                  ),
                  Text(
                      "published: "
                  )

                ],
              )
            ],
          ),
        ),
      ),
    );

  }

  Widget buildTextField(
      String labelText, String placeholder, bool isPasswordTextField) {
    return Padding(
      padding: const EdgeInsets.only(bottom: 35.0),
      child: TextField(
        //obscureText: isPasswordTextField ? showPassword : false,
        decoration: InputDecoration(
            suffixIcon: isPasswordTextField
                ? IconButton(
              onPressed: () {
                setState(() {
                  //showPassword = !showPassword;
                });
              },
              icon: Icon(
                Icons.remove_red_eye,
                color: Colors.grey,
              ),
            )
                : null,
            contentPadding: EdgeInsets.only(bottom: 3),
            labelText: labelText,
            floatingLabelBehavior: FloatingLabelBehavior.always,
            hintText: placeholder,
            hintStyle: TextStyle(
              fontSize: 16,
              fontWeight: FontWeight.bold,
              color: Colors.black,
            )),
      ),
    );
  }
}
```

CHAPTER 7

TESTING

Testing is an investment conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risk of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs and verifying that the software product is fir for use.

Quality plays a crucial role in the success of a business. It is the degree of excellence which ensures some certain standards in a product. Based on quality, products can be categorized in two types, good quality products and bad quality products. A good quality product meets all the requirements of the customers or users whereas a bad or poor-quality product does not meet the expectations of the users.

The quality of a piece of software is based on the requirement specification document which is created before designing the software and can be defined as a blueprint of the software. To ensure high level of quality of the software you need to do rigorous and effective software testing. The testing not only ensures that the software is consistent and bug-free, but it also checks whether the software meets the standards of quality and functionality as per expectations of the users before the software is released into the market. Software testing can be done through some testing models. Among these models, the verification and validation model (V model) is the most popular model of testing.

In this chapter, you learn about quality its importance, as well as different methods used for maintaining quality. The chapter also discuss software quality metrics, software quality management system, software quality assurance, and quality standards. It also explains the psychology of software testing, the fundamental concepts of software testing and different levels of software testing. Towards the end, the chapter discusses the W model of software testing.[11]

FOR TESTING WE NEED TEST STRATEEGY AND TEST DESIGN

7.1 Definition of Quality

Quality is a general term that defines the degree, level, or extent up to which a product or service meets the requirements and specifications of the intended customers. The quality of a product/service is directly related to the satisfaction level of the customers. Quality depends on the following three aspects:

Customer requirement: Plays a crucial role in producing a product or in providing a service. A customer or user would buy a product or service if the product fulfills all the requirements. The product can be said of good quality in case it satisfies all the needs of customers.

Customer satisfaction: Implies that the quality of a product can be evaluated by the level of satisfaction that users derive from it. In other words, the better the quality of a product is, the more satisfied the users would be. Therefore, we can call customer satisfaction an accurate measure of quality.

Level of excellence: Ensures the degree of excellence in a product service. If the product fulfills all the requirements of the customers, it is considered of high-quality product. Based on availability of the product in the market, a customer can decide if the product is of good quality or poor quality.

After defining quality, let's discuss its importance.[9]

7.2 Importance of Quality

Quality is of paramount importance for the success of an organization. The level of quality of products or services largely determines the volume of customers it attracts as well as the satisfaction level of the customers. By offering good quality products or services, an organization ensures a high degree of satisfaction among customers. And satisfied customers would continue doing business with the organization and would contribute to its growth and prosperity. On the other hand, poor quality products/services would discourage customers and that would have a negative impact on the organization. Therefore, quality is a key to the survival of an organization.

We know that the quality of products is important for an organization because it:

1. Helps in maintaining high customer satisfaction as customers go for good quality products. Customer satisfaction, if nurtured properly, transforms into customer loyalty.
2. Determines the brand value of a product. The brand value of the product is based on the quality maintained by the company. If an organization is producing high quality products, the brand value of the product would increase.
3. Attracts the customers towards the features of a product. Usually, the customers invest in good quality and reliable products.[9]

7.3 Managing Quality

Quality is managed at each level of the production process in an organization. It is not controlled by any individual person but a group working as a chain. In this chain, every person has a specific role to play. If a person does not perform to expectations, the quality is likely to suffer.

To manage quality an organization has to:

1. **Understand the requirements of the user**: When a customer or user purchases a product, he/she expects it to be of a certain quality. It is one of the prerequisites of the user and.is taken very seriously by organizations. To provide quality products to its customers, an organization has to first have a clear understanding of the customers requirements. Therefore, understanding the requirements of the users is the first step towards managing quality.
2. **Focus on product design**: While designing a product an organization should maintain the quality because the design helps us in determining both appearance and functionalities of the product. Therefore, the organization should focus on the product design to ensure good quality product.
3. **Make the product cost-effective**: While purchasing a product, a customer always has three things in mind: price, functionality, and quality. For a product to be

successful, it has to offer the maximum number of functionalities at a reasonable price and without compromising on quality. A customer always looks for cheaper options in the market.

4. **Make verification before delivery**: Before delivering the product an organization should verify whether the product is according to the requirements of customer or not. The product is of a good quality if it meets all the requirements,

5. **Ensure customer satisfaction**: An organization should analyze the satisfaction level of the customers on the basis of the feedback taken from the market. It should try to find out if the customers are satisfied with the quality of the product. If they are not satisfied it means that the product is not as per their requirements.[11]

7.4 MOBILE TESTING

Mobile computing is probably the biggest technology change that has greatly impacted the tester's realm of operations. Moving away from a desktop or laptop-based web application testing to a device or simulator/emulator-driven testing of web, native, or hybrid applications is a large change the testers are still getting used to. From a physical impact as well, testing on mobile devices with a hand-held has been a big change. Obviously, it is just not a device that has brought in the change. The device in combination with an application development mind-set and the overall application development process has created a very large test impact. Mobile applications have brought in a lot of focus on nonfunctional testing elements such as performance, security, and usability. Newer business models around m-commerce and content digitization have all triggered new testing opportunities. In fact, it is interesting to see business strategies such as discounts being offered only on the mobile app, mobile shopping spree, and large retailers such as Flipkart in India working toward a mobile-only presence. Crowdsourced testing strategy has become more prevalent given the need to test applications on a large range of mobile devices across realistic user scenarios. The tester himself is now encouraged to think like an end user. Mobile testing is forcing testers to think deeper and bring out optimization opportunities—for example, we have a homegrown tool at QA InfoTech, called Mobile Application Strategy Tool. This is a tool that will help you consider varied parameters and weigh in on them to make suggestions on what kind of testing should be done to maximize overall test coverage with a minimal set of tests to be run. This will also help you make a call on where the tests should be run (whether on physical devices or on simulators through a crowd team).[10]

7.5 TEST STRATEGY

When a test effort was initially conceptualized as an independent activity to be performed by testers who were not involved in the development effort, the need for a test strategy came in. This is a document that talks about the product, its architecture, what are the varied test areas (for which test plans would subsequently be created), entry and exit criteria for the overall release, resource mapping identifying testers who would be working on varied components, what the group's overall test and defect management practices would be, etc. This has served as a very valuable document in the waterfall days, helping the testers pause and understand the large context of what they were involved in, how each other's work interlaced, what are some of the common practices in the team, etc.[7]

7.5.1 Universal Rules

In the testing profession there is no silver bullet and no snake oil. Each and every project we are working on is completely different from all the others, even if they look very similar. You always need to approach them individually. Each project will require its own analysis, test approach and selection of tools and techniques. The real tester's wisdom is to know exactly what and when to do in given circumstances. However, regardless of the problem you are working on, there are some universal rules that apply to all testing projects. Here they are:[2]

1. Know your client's needs.
2. Know where you are.
3. Know your enemies (faults).
4. Be reasonable and professional.
5. Be optimal.
6. Don't give up.
7. Think

7.6 TEST DESIGN

A lot of emphasis was placed on designing test cases. While some teams left test cases at scenarios level, most teams would take in the scenarios from the test plans to elaborate them into a set of individual test cases of varying levels of detail. Some test cases were so detailed that a new tester coming into the test effort can merely review the test cases to understand application workflows. While such granularity helped ensure no confusion on what tests were executed and also to train new people, it also meant that so much more time was being spent on writing test cases, during which the tester could have actually tested the product. Maintenance was also a huge overhead where even for a small feature change, the tester had to spend a lot of time updating the test cases.[7]

7.7 Approach Overview

7.7.1 Pre-processing.

Since the range invariants bug localization method needs to analyze the value information of variables when the program executes, it is necessary to instrument the program source code before the program is executed. And then execute the instrumented program according to the given inputs to obtain a series of test cases that contain execution path information and variables value information. [6]

7.7.2 Test Case Optimization.

Our method is known the error program, inputs and expected output results. Distinguished successful test cases and failure test cases by checking whether the execution results are the same as the expected results. In general, there are accidental correct test cases (CC) in successful test cases. If the number of such accidental correct test cases is large, the scope of the range invariants is increased, and the possibility of violating the range invariants is reduced, so that there are many false negatives. Therefore, we reduce the scope of range invariants by deleting accidental correct test cases, thereby increasing the possibility of violating invariants during the bug localization phase, thereby reducing false negatives. In this paper, we use the algorithm proposed by Masri to delete accidental correct test cases. And CCT is used to represent the set of successful test cases identified as CC. [12]

7.7.3 Screening Key Variables.

In this paper, we first optimize the two variable evolution pattern detectors proposed in reference, and obtain two dynamic filtering mechanisms. Then we combine two dynamic filtering mechanisms and a static reduction mechanism to filter the variables in the program and obtain the set of key variables in the program. [6]

7.7.4 Bug Localization.

First, we analyze the values of key variables in successful test cases, and obtain the range invariants of key variables statistically. Then, Software Bug Localization Based on Key Range Invariants 23 detect failure test cases. If the value of the key variable in failure test cases is not within the range of the corresponding invariant, the flag is violated. That is, the statement containing the variable is marked as a suspicious statement.[6]

7.8 UI TESTING

User interface (UI) is a high return area when it comes to engaging the crowd. The crowd, especially an end user base-representing crowd, can provide strong UI feedback that aligns with its preferences. Since it aligns with the crowd's preferences and touches a softer and intangible aspect of the end user, the crowd can potentially add more value in UI testing than even the internal test team. And since UI is an area of focus primarily for end user facing pages, screens, and sections, this is an area where the crowd adds very high value through its tests and feedback. Domain expertise is very important in this case, compared to the crowd's end user representation.[5]

7.9 PERFORMANCE TESTING

This is an area that is given a lot of attention and focus in a test strategy in parallel to the core functional aspects. Typically, there is a separate team that focuses on performance testing the application from various angles, including load, stress, capacity and scalability, long haul, competitive baselining, etc. While the crowd cannot partake in all activities of performance testing due to infrastructure and system access limitations, if the product company diligently plans on how to use the crowd in this space, it can get some very valuable real time feedback on the application's performance, which only the crowd can provide while still in the QA phase.[5]

CHAPTER 8

ANDROID SECURITY

8.1 ANDROID SECURITY MODEL

Android developers have included security in the design of the platform itself. This is visible in the two-tiered security model used by Android applications and enforced by Android. Android, at its core, relies on one of the security features provided by Linux kernel—running each application as a separate process with its own set of data structures and preventing other processes from interfering with its execution.

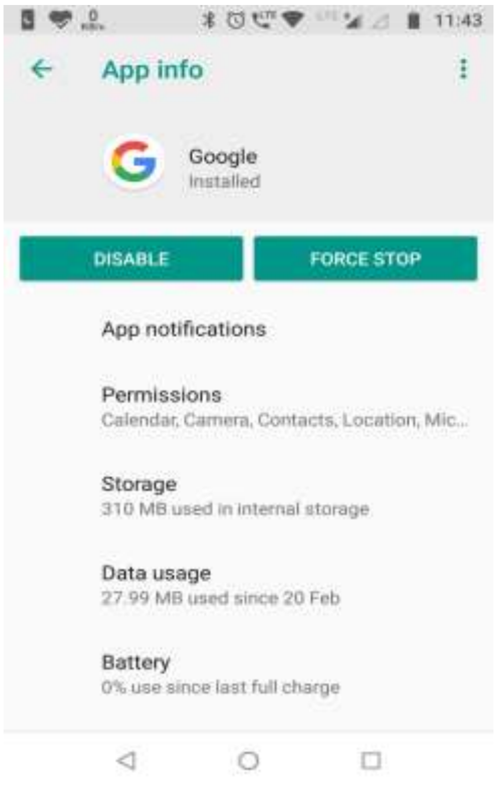
The application layer, Android uses finer-grained permissions to allow (o disallow) applications or components to interact with other applications/components or critical resources. User approval is required before an application can get access to critical operations (e.g., making calls, sending SMS messages). Applications explicitly request the permissions they need in order to execute successfully. By default, no application has permission to perform any operations that might adversely impact other applications, the user's data, or the system. Examples of such operations Include sending SMS messages, reading contact information, and accessing the Web. Playing music files or viewing pictures do not fall under such operations, and, thus, an application does not need to explicitly request permissions for these. Application-level permissions provide a means to get access to restricted content and APIs. Each Android application (or component) runs in a separate Dalvik Virtual Machine (VM)—a sandbox. However, the reader should not assume that this sandbox enforces security. The Dalvik VM is optimized for running on embedded devices efficiently, with a small footprint. It is possible to break out of this sandbox VM, and, thus, it cannot be relied on to enforce security. Android per-mission checks are not implemented inside the Dalvik VM but, rather, inside the Linux kernel code and enforced at runtime. Access to low-level Linux facilities is provided through user and group ID enforcement, whereas additional fine-grained security features are provided through Manifest permissions.[1]

8.2 PERMISSION ENFORCEMENT-LINUX

When a new application is installed on the Android platform, Android assigns it a unique user id (UID) and a group id (GID). Each installed application has a set of data structures and files that are associated with its UID and GID. Permissions to access these structures and files are allowed only to the application itself (through its ID) or to the superuser (root). However, other applications do not have elevated superuser privileges (nor can they get them) and, thus, can-not access other applications' files. If an application needs to share information with other application(s) or component(s), the MAC security model is enforced at the application layer (discussed in the next section).It is possible for two applications to share the same UID or run in the same process. This can be the case if two applications have been signed by the same key (see application signing in Chapter 3). This should underscore the importance of signing keys safely for developers. Android applications run in separate processes that are owned by their respective UID and thus sandboxed from each other. This enables applications to use native code (and native libraries) without worrying about security implications. Android takes care of it.[3]

8.3 ANDROID'S MANIFEST PERMISSION

The Linux kernel sandboxes different applications and prevents them from accessing other applications' data or user information, or from performing operations such as accessing the Internet, making phone calls, or receiving SMS messages. If an application needs to perform the aforementioned operations (e.g., Internet access), read the user's information (e.g., contacts), or talk to other applications (e.g., communicate with the e-mail application), the application needs to specifically request these permissions (MAC model). Applications declare these permissions in their configuration file (Manifest.xml). When an application is installed, Android prompts the user to either allow or reject requested permissions A user cannot select certain permissions—that is, allow access to the Internet and reject SMS access. The application requests a set of permissions, and the users either approve or deny all of them. Once he user has approved these permissions, Android (through the Linux kernel) will grant access to the requested operations or allow interaction with different applications/components. Please note that once the user has approved permissions, he cannot revoke them. The only way to remove the permissions is to uninstall the application. This is because Android does not have the means to grant permissions at runtime, as it will lead to less user-friendly applications. Android permissions are also displayed to the end-user when downloading applications from the “official” Android market (see Figure 4.5). However, this might not always be the case, as there are quite a few sources for Android applications. If the user just downloads .apk files, a warning about security implications will only be displaced during runtime.[3]



Permissions
Calendar, Camera, Contacts, Location, Mic...

Storage
310 MB used in internal storage

Data usage
27.99 MB used since 20 Feb

Battery
0% use since last full charge

Figure 7.1 Manifest permission

8.4 MOBILE SECURITY ISSUES

The Android platform suffers from “traditional” security concerns, just like any other mobile OS. The issues discussed below are common to all mobile platforms, not just the Android. Some of these issues are also found on traditional devices (laptops), whereas some are specific to mobile devices.[1]

8.4.1 DEVICES

Many of us have, at some point, lost a cellular device. Before the advent of smartphones, it meant losing one’s contact information. On a typical (Android) smartphone today, however, the following is true for most of us:– E-mails saved on the mobile device– Auto sign-in to Facebook, Twitter, YouTube, Flickr, and more– Bank account information– Location and GPS data– Health data Unless the device is encrypted, the loss of a cell phone implies a potential data disclosure risk, as well. Plug in a cellphone to a computer, and various tools (including forensic tools) will do the rest.[1]

8.4.2 PATCHING

Android's latest version is 3.2. However, most devices in use today are running anything from Android 1.5 to Android 2.3, with 2.2 and 2.3 being the most popular releases. Furthermore, these devices are updated/modified by the respective manufacturers. Thus, it is difficult to apply patches in a timely manner given the lack of uniformity of the OS used. Compare this to the iPhone, where IOS 3 and IOS 4 are the only versions available today.[1]

8.4.3 EXTERNAL STORAGE

Removable external storage compounds the data security issue. It is much easier to lose SD cards than to lose a cell phone. In most cases, data is not encrypted, thus giving very easy access to the user's data. SD cards also travel through multiple devices, thus increasing the risk of malicious software ending up on the device. Finally, removable storage is often more fragile, which can lead to data loss/corruption.[1]

8.4.4 KEYBOARDS

Although a very popular feature, touch screen keyboards can give goose bumps to a security professional. They provide a perfect opportunity for shoulder surf-ng, if you are accessing sensitive data in a train or in a coffee shop. Tablets are even worse culprits, with full-size soft keyboards and letters being reflected to the user in plaintext for few seconds. Smudges on the screen may also aid an attacker.[1]

8.4.5 DATA PRIVACY

One of the most popular applications on Android is Google Maps. Many other applications are also interactive and can use the user's location information. They can store this information in its cache, display ads based on this data, or show us the nearest coffee shot. Bottom line: This data is available for any application that has the right permissions. Over a period of time, this data can reveal sensitive information about a user's habits, essentially acting as a GPS tracking in the background.[1]

8.5 APPLICATION SECURITY

Mobile applications are still vulnerable to the same attacks as traditional, full-fledged information technology (IT) applications. SQL Inject (SQLi), Cross-Site Request Forgery (XSRF), and Cross-Site Scripting (XSS) are not only possible on mobile platforms and applications but can lead to more serious attacks, given the nature of data available on a mobile device. Weak Secure Sockets Layer (SSL) or lack of encryption, phishing, authentication bypass, and session fixation are all issues likely to be present in mobile applications.

8.6 LEGACY CODE

Much of the underlying code used by cell phones for GSM or CDMA communication has not changed much over the years. These device drivers were written without security practices in mind and thus are vulnerable to old-school attacks (e.g., buffer overflows). New devices continue to rely on this code. In fact, new code is being added on the top of existing code.

8.7 RECENT ANDROID ATTACKS - A WALKTHROUGH

In the first week of March 2011, a malware Droid Dream—hit the Android platform. Android is a much more open platform compared to iOS and, thus, has a lenient marketplace policy. Google does not tightly control applications that show up in the market. In fact, Google does not even control all channels of distribution, unlike Apple. Various ways to get applications on Android are as follows:– Official Android market (Google)– Secondary Android markets (e.g., Amazon)– Regional Android markets and app stores (e.g., China, Korea)– Sites providing apk files to users Similar to other Android malware, such as Geinimi and HongTouTou, Droid Dream was "hidden" or "obfuscated" inside a legitimate-looking application. Regular users having no reasons to distrust the Android market downloaded the application and ended up having an infected device. After the outbreak of this malware, Google took an extraordinary step—the remote wiping of devices that were infected (approximately 50 applications were considered to be malicious). Droid Dream and its variants gained access to sensitive user and device information and even obtained root access. For a complete list of malicious applications on the list, perform a search on Google for "MYOURNET."

CHAPTER 9

LIMITATIONS

9.1 LIMITATIONS

The limitations of the study are those characteristics of design or methodology that impacted or influenced the application or interpretation of the results of your study. They are the constraints on generalizability and utility of findings that are the result of the ways in which you chose to design the study and/or the method used to establish internal and external validity.

Every project in this world have some limitations, similarly our project also has some of them:

- Internet based, without internet user cannot access its tools.
- Server Dependent
- User Details must be correct

9.2 FUTURE SCOPE

The future scope of the project are as follows:

- Providing better User Experience
- Additional features like chat support
- Reduce Load Time

9.3 FUTURE ENHANCEMENT

- Provide Notification to the user
- More reliability on application
- Graphics enhancements

CHAPTER 10

CONCLUSION

Working on the project was a good experience. Working together in a team helped us to communicate better. We understood the importance of planning and designing as a part of software development.

Developing the project has helped us to gain some experience on real time development procedures.

This is an application for those users that requires assigning daily basis task to students or to employees. With the help of this application the physical reporting to mentor for getting new tasks and for updating about task status is eliminated.

CHAPTER 11

BIBLIOGRAPHY

1. Adam Roman (Thinking-Driven testing: A Universal Testing Framework)

1st Edition

Publisher: Springer, Cham

Pages; 99-143

https://link.springer.com/chapter/10.1007/978-3-319-73195-7_3

2. Anmol Misra (Android Security-Attacks and Defenses)

1st Edition

Publisher: Auerbach Publications

Pages:71-73

https://www.taylorfrancis.com/books/9780429111600

3. Erik Hellman (Android Programming)

Publisher: Wiley

https://ebooks.wileyindia.com/epubreader/android-programming

4. Jerome DiMarzio (Beginning Android Programming with Android Studio)

4th Edition

Publisher: Wiley

https://ebooks.wileyindia.com/epubreader/beginning-android-programming-studio

5. Lin Ma (Software Bug Localization Based on Key Range Invariants)

Research paper from: School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310019, Zhejiang, China

https://link.springer.com/content/pdf/10.1007%2F978-3-030-04272-1_2.pdf

6. Mukesh Sharma (Leveraging the Wisdom of the Crowd in Software Testing)

1st Edition

Publisher: Auerbach Publications

https://www.taylorfrancis.com/books/9780429189722

7. Oliver Mesly (Project Feasibility)

1st Edition

Publisher: Boca Raton

https://www.taylorfrancis.com/books/9781315295251

8. Rajesh K Maurya (Software testing, ISBN 9789350044001,

   Publisher: Wiley

   https://ebooks.wileyindia.com/pdfreader/software-testing)

9. Rajini Padmanabhan (Leveraging the Wisdom of the Crowd in Software Testing)

   1st Edition

   Publisher: Auerbach Publications

   https://www.taylorfrancis.com/books/9780429189722

10. Swati R Maurya (Software testing, ISBN 9789350044001,

    Publisher: Wiley

    https://ebooks.wileyindia.com/pdfreader/software-testing)

11. Zuohua Ding (Software Bug Localization Based on Key Range Invariants)
    Research paper from: School of Information Science and Technology, Zhejiang Sci-Tech

    University, Hangzhou 310019, Zhejiang, China

    https://link.springer.com/content/pdf/10.1007%2F978-3-030-04272-1_2.pdf

12. https://flutter.dev/

13. https://pub.dev/

14. https://en.wikipedia.org/

15. https://stackoverflow.com/

16. https://medium.com/

17. https://github.com/