VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report

on

"BARBER SHOP APP"

Submitted in partial fulfillment of the requirements as a part of the VIII semester of Degree of Bachelor of engineering in information science and engineering of Visvesvaraya Technological University, Belagavi

By

Sushmitha N J

1RN19IS409

Under the Guidance of

Mrs. Kusuma R

Assistant Professor

Department of ISE

Department of Information Science and Engineering

RNS Institute of Technology

Channasandra, Dr. Vishnuvardhan Road, RR Nagar Post,
Bengaluru – 560 098

2021 -2022

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, RR Nagar Post,

Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Barber Shop App*** has been successfully completed by **Sushmitha N J (1RN19IS409)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022.** The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mrs. Kusuma R	**Dr. Suresh L**	**Dr. M K Venkatesha**
Internship Guide Assistant Professor Department of ISE	Professor and HOD Department of ISE RNSIT	Principal RNSIT

External Viva

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

DECLARATION

I, **Sushmitha N J [USN:1RN19IS409]** students of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Barber Shop App***has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester degreeof* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :

SUSHMITHA N J

(1RN19IS409)

ABSTRACT

Searching out for the best barber shops or hair salons within your vicinity is a very tiresome task. Also, visiting a barbershop or salon without a prior appointment booking can keep you waiting in a queue thereby taking away your precious time. To solve this issue, we have developedan android and web based barbershop application. By using flutter.

There are several services available in barbershop so user can select the specified services and also, they can book appointments. Both the barbershop owners & users can access the application by registering themselves on the app by submitting the necessary details. The barbershop owners can update the list of services provided by them. The users can view the photos & services, based on which they can book an appointment as per their desired time & date.

This project uses Flutter as front-end tool Flutter is free and open-source UI software development kit introduced by google. The flutter development tools come with graphics library and material design allowing faster operations of the app and also giving the app a stunning look, irrespective of its operating platform! The biggest advantage of flutter is that it can be used to create cross-platform. For appointment scheduling, app will integrate with the calendar. The final deliverable will be a functioning mobile application that can handle all specified use cases. Some of the major use cases includes splash screen, booking page with includes services available in barber and services, appointment page which contains the appointment details of the particular person includes available time, and date.

ACKNOWLEDGMENT

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

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

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

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

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

I place my heartfelt thanks to **Mrs. Kusuma R** Assistant Professor, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for helping at all times.

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

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

Sushmitha N J
1RN19IS409

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 HISTORY

Flutter is an express stage structure that goes for growing elite portable applications. Flutter is openly discharged at 2016 with Google. Not exclusively can Flutter apps keep running on iOS Android, yet additionally Fuschia, Google's cutting edge working framework, picks Flutteras its app level structure. Flutter is special. As opposed to using web sees or depending on the device's OEM gadgets, Flutter renders each view parts utilizing its own superior rendering motor. This nature gives plausibility to assemble apps that are as superior as local apps can be. Flutter bolsters stateful harm reload while creating, which is look as a main consideration to support advancement cycle. Stateful harm reload is basically actualized by infusing refreshed source symbol into the working Dart virtual machine without modifying the internal design of the app, consequently all advances and activities of the app will be protected after harm reloading

.

In this mean, Flutter underpins utilizing shared bundles contributed by different designers to the Flutter and Dart biological systems. This permits to rapidly manufacture the application without creating everything sans preparation. Existing bundles empower many use categories, for instance, making system demands (http), custom route/course dealing with (Fluro), joining with gadget Android Programming Interfaces (such as universal resource locater launcher and battery), and utilizing outsider stage SDKs (such as Firebase) [2]. We must use Flutter because:

1) Flutter utilizations Dart, a quick OOP with a few valuable highlights, for example, generics, mixins, secludes, and discretionary static sorts.
2) Flutter has it is very own user interface parts, alongside a motor to deliver them on the iOS and android stages. The majority of those user interface parts, directly out of the container, fit in with the rules of Material Structure.
3) Flutter applications can be created utilizing IntelliJ that is fundamentally the same as Android Studio.

1.2 Methods and Material

1.2.1 Dart

In Flutter all apps are composed with Dart. So, Dart is a OOP that is created and kept up by Google. It is generally utilized within Google, it has been demonstrated to have the ability to create enormous web apps, for example, AdWords.

Dart was initially created as a substitution and successor of Java Script. Subsequently, it actualizes the majority of the significant attributes of Java Script's next standard (for example, watchwords "async" and "wait". Notcwithstanding, so as to draw in designers that are curious about Java Content, Dart has a Java such as linguistic structure. Likened to different frameworks that use responsive perspectives, Flutter app invigorates the see tree on each new casing.

Dart is a programming coding that we will utilize to build up our app in Flutter. Study it is not hard on the off chance that we have involvement with java content. We will rapidly fetch it. Dartis the customer enhanced language for quick applications on any stage do by Google. Superiorand efficiency in Flutter are accomplished by utilizing a few procedures:

1) Not at all like numerous other prominent versatile stages, Flutter makes not utilize java content at all. Dart is the coding. It assembles to parallel code, and that is the reason keeps running with the local execution of target java, quick, C.

2) Flutter does not utilize local user interface parts. That can sound clumsy at first. In any case, since segments are executed in Flutter itself, there is not correspondence see between the layer and the code. Because of this, diversions hit the best performance for their illustrations out of the cell phones, So catch, content, media components, foundation are altogether drawn by Flutter is designs motor. As a part, it ought to be referenced that the heap of the Flutter "Hi World" app is very little: android= 4Mb and iOS=2.5Mb.

3) Flutter utilizes a definitive methodology, motivated by the respond web system, to manufacture its user interface dependent on gadgets (named "parts" in the realm of the site). To fetch increasingly out of gadgets, they are delivered just when fundamental, for the most part when their case has been modifcated (simply such as the practical DOM accomplishes for us).

1.2.2 Flutter goal

- ✓ Beautiful fluid user interfaces.
- ✓ Run same user interface on numerous stages, ideal for brand-first plans.
- ✓ Superior applications that vibe regular on various stages.
- ✓ Be gainful.

1.3 Architecture overview

Messages and reactions are passed no concurrently, to guarantee the UI stays responsive. On the customer side, Method Channel (Programming interface) empowers sending msgs that relate to strategy calls. On the stage side, Method Channel at Android devices (Programming interface) and FlutterMethodChannel at iOS (Programming interface) empower getting strategy gets back to and sending an outcome. These classes permit to build up a stage module with almost no 'standard' symbol. Messages are sent between the customer (user interface) and host (stage) utilizing stage channel

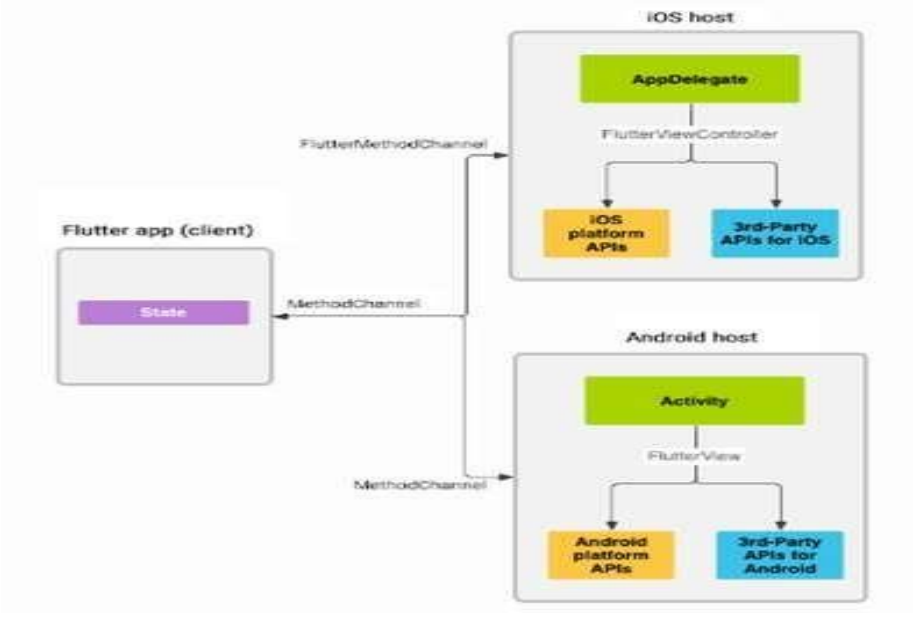

Fig 1.1: Architectural overview: platform channels

Architectural overview diagram shows us how exactly the platform channels work with both IOS as well as Android devices.

Widgets are the great significant components in a Flutter app. Widgets should be appealing and sensible in light of the fact that client (see and feel) them legitimately. Widgets don't just controland influence how the perspectives carry on, yet in addition handle and react to the client's activity.In this manner, it is urgent that Widget's requirement to perform quickly, including delivering andquickening. Rather than use again OEM Widgets, similarly as what Respond Local does, Fluttergroup chooses to give its very hold Widgets. This implies Flutter, as a stage, gets the chance tochoose how and when Widgets are delivered. As it were, Flutter shifts Widgets and deliverer fromframework level into the app itself, which enables them to be progressively adaptable andextensible. In any case hold the Widgets and deliverer inside the app makes the extent of usebigger.

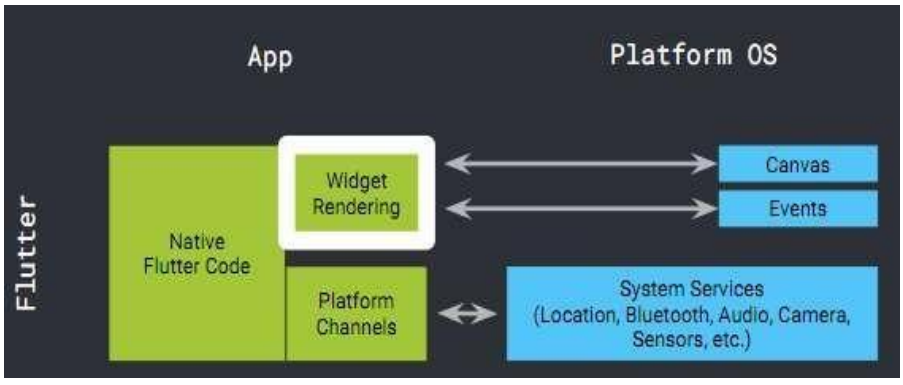

Fig 1.2: Widget Rendering

1.4 Flutter

One of the greatest upgrades in Flutter is the means by which it does design. Format decides the extend and location of widgets dependent on a lot of standards. Generally, layout utilizes a huge arrangement of standards that ability be connected to (for all intents and purposes) any widget. The principles actualize numerous layout strategies. To take for instance (Android XML).

It is a great deal of properties and characteristics, which are connected to all sight components. Every widget might have their very own property. Also, parent design models are now predefined, and we have to tail it orders. This outcome in less area for improvement and a ton of hacks as composing own design parent is dangerous and may not justified, despite any potential benefits. Another issue with customary layout the principles can collaborate (and even clash) with one another, and components frequently have many standards connected to them. This does layout moderate. Far and away more terrible, design execution is ordinarily of request N-squared, also as the quantity of components expands, layout backs off considerably more.

Flutter incorporates many widgets for making layout, segments as well as lines, matrices, records, and so forth. What's more, Flutter has a one-of-a-kind layout pattern we call the "bit design model" which is utilized for looking over. Layout with Flutter is so quick it very well may be utilized for looking over. Consider that for a minute. Looking over must be so immediate and fine that the client feels such as the screen picture is joined to their key as they pull it over the physical screen. By utilizing layout for looking over, Flutter can execute propelled sorts of looking with heaps of movement.

1.4.1 Flutter Installation

To install and run Flutter on the Windows system, you need first to meet these requirements for your development environment.

Operating System	Windows 7 or Later (I am Windows 10. You can also use Mac or Linux OS.).
Disk Space	400 MB (It does not include disk space for IDE/tools).
Tools	1. Windows PowerShell 2. Git for Windows 2.x (Here, Use Git from Windows Command Prompt option).
SDK	Flutter SDK for Windows
IDE	Android Studio (Official)

Table 1.1: Flutter Installation

Step 1: Download the installation bundle of the Flutter Software Development Kit for macOS. To download Flutter SDK.

Step 2: When your download is complete, extract the zip file and place it in the desired installation folder or location.

Step 3: To run the Flutter command, you need to update the system path to include the flutter bin directory.

Step 4: Next, enable the updated path in the current terminal window using the below command and then verify it also.

Step 5: Now, run the $ **flutter doctor** command. This command checks for all the requirements of Flutter app development and displays a report of the status of your Flutter installation.

Step 6: When you run the above command, it will analyze the system and the details of all missing tools, which required to run Flutter as well as the development tools that are available but not connected with the device.

Step 8: Next, you need to set up an iOS simulator or connect an iPhone device to the system for developing an iOS application.

Step 9: Again, set up an android emulator or connect an android device to the system for developing an android application.

Step 10: Now, install Flutter and Dart plugin for building Flutter application in Android Studio. These plugins provide a template to create a Flutter application, give an option to run and debug Flutter application in the Android Studio itself.

1.5 Android Architecture

Android architecture contains different number of components to support any android device needs. Android architecture or Android software stack is categorized into five parts:

1. Linux kernel
2. Native libraries (middleware)
3. Android Runtime
4. Application Framework
5. Applications

Let's see the android architecture first.

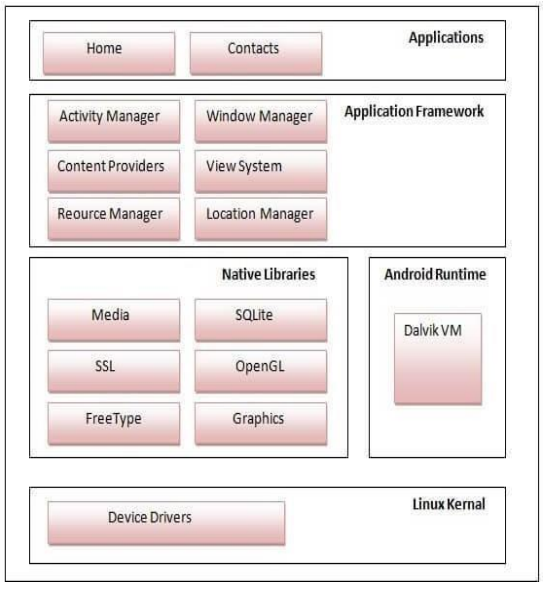

Figure 1.3: Android Architecture

1) Linux Kernel:

   It is the heart of android architecture that exists at the root of android architecture. Linux kernel is responsible for device drivers, power management, memory management, device management and resource access.

2) Native Libraries:

   On the top of Linux kernel, there are Native libraries such as Web Kit, OpenGL, Free Type, SQLite, Media, C runtime library (libc) etc. The Web Kit library is responsible for browser support, SQLite is for database, Free Type for font support, Media for playing and recording audio and video formats.

3) Android Runtime:

   In android runtime, there are core libraries and DVM (Dalvik Virtual Machine) which is responsible to run android application. DVM is like JVM but it is optimized for mobile devices. It consumes less memory and provides fast performance.

4) Application Framework:

   On the top of Native libraries and android runtime, there is android framework. Android framework includes Android API's such as UI (User Interface), telephony, resources, locations, Content Providers (data) and package managers. It provides a lot of classes and interfaces for android application development.

5) Applications:

   On the top of android framework, there are applications. All applications such as home, contact, settings, games, browsers are using android framework that uses android runtime and libraries. Android runtime and native libraries are using Linux kernel.

1.6 INTRODUCTION TO THE PROJECT

1.6.1 OVERVIEW OF THE PROJECT

Gone are the days when barber shop business was something only people took up to keep themselves busy in their free time. In the past couple of decades, it has seen a significant surge, to the point that today, the beauty and wellness sector is a full-fledged industry in itself! It would not be an overstatement to claim that the salon and spa industry is one of the most profitable business ventures in today's times, internationally as well as in India. As per a survey conducted by the cosmetic industry, it has experienced a growth of almost 60% in the last 5 years itself. In fact, it has been estimated that the hair and beauty industry is likely to generate revenue of more than 5 billion dollars by the end of the financial year 2019-2020.

Some of the factors that have contributed to the rapid growth of the beauty and wellness industry include increase in the urban population, increase in the population of working women in India, which gives them the financial freedom to indulge themselves, evolution of technology and science, rise in the salary standards because of an increase in international companies investing in India and a general tendency amongst men and women of all age groups wanting to look good, thanks to the increasing influence of social media, especially amongst the younger generation.

The beauty barber shop has become an almost iconic figure in Western culture and Southeast Asian culture as well as in modern generation. The beauty salon is where a woman goes to have their hair and nails done, but is also a center for community news confessions and general 'hen parties". As such, choosing a beauty salon is a process most women (and even men) will undergo. The first thing to start in choosing a salon is to evaluate what needs to be done. A woman who just wants her hair trimmed might go to a beauty salon than women who wants a permanent coloring, or more extensive services. If a woman wants her hair cut promptly and that day, she might be better off visiting to a walk-in salon, more willingly than trying to find her regular stylist. For other kind of services, a woman might want to think about a full-service in by-appointment-only beauty salon. This signifies that she will have a stylist who is expecting her and definitely knows what she wants to be done.

1.6.2 AIM OF THE PROJECT

Barber shop booking app is one where the user can book their slot and even, they can book thebest stylist available at there.

The main issue that most Salon owners generally go through is handling appointments. From sending emails, answering calls, to attending clients at the Salon, there are a lot of headaches. Many of the Salon owners fail to delight clients with their service, as they do not have an accurate and manageable system. There comes a time when Salon businesses experience potential clients' dissatisfaction.

General features of online salon booking system are Appointment management, point of sales and billing system, marketing campaigns, Reporting, staff management, customer Management etc.

Benefits of online booking system in the salon are:

1.6.3 Manifest a healthy relationship with the clients

1.6.4 Build and grow the salon revenue

1.6.5 Monetary savings

1.6.6 Centralized Information system

1.6.7 Email and SMS Remainders

1.6.8 Valuable Insight into the salon

CHAPTER 2

LITERATURE SURVEY

Paper[1] Flutter is a popular UI framework for developing mobile applications by Google. It has caught traction in recent years. However, Flutter developers have to deal with a state management issue when developing their applications. In order to solve this problem, multiple architectures have been developed. This paper proposes a new Flutter architecture based on the Clean Architecture by Uncle Bob. The Flutter Clean Architecture proposed in this paper is packaged and released through a Flutter package. The architecture is tested by developing a full application from scratch using the package and documenting the process. The Flutter Clean Architecture provides a solution to the state management problem as well as a potential overall choice for Flutter mobile application architecture.

Paper[2] Since barber shop is a service-sector, so the success of barber shop depends on the satisfaction of customer. Definition of customer satisfaction has been widely discussed from the view of many researchers & organizations who increasingly desire to measure it. The study of Social policy (2007) conceptualize that satisfaction is based on customer's experience of both contact with the org and personal outcome. The researchers defined satisfaction as a "highly personal assessment that is greatly influenced by individual expectations". Boulding et al (1933)and Yi and La (2004) conclude satisfaction into two general conceptualizations; Transaction- pacific satisfaction and cumulative satisfaction. Transaction-pacific satisfaction is a customer evaluation of her particular experience and reaction to a particular service encounter (Croning &Tailor, 1992; Boshoff & Gray.2004). Cumulative satisfaction refers to the customer overall evaluation of the consumption experience to date (Jones & Suh, 2000)

Paper[3] The traditional system of hair salon was manual and insecure because there was no any counting system of customers coming in the salon which creates sometimes major issues. The customer as well as the owner faces the problems.[2] These early systems are dependent on paper-pencil systems for billing purpose that means the records of bills of customer and the workers working in their salon are in written form. The records may get wrong due to anyone's mistake. There is difficulty in maintaining records of all these tasks manually. Hence, proposed system is the best solution of avoiding all these problems. Salon manager are often responsible scheduling staff members, training new front desk workers. [3]Since salon is a service sector, so the success of hair salon depends on the satisfaction of customer. The business of salon is totally dependent on customer satisfaction.

Paper[1]: Shady Boukhary∗ , Eduardo Colmenares†Department of Computer Science, Midwestern State University Wichita Falls, TX

Paper[2]: https://www.scribd.com/doc/30979857/Literature-Review

Paper[3]: Smart Hair Salon Management System Ms. Jadhav Vaishali1, Ms. Kumthekar Aarti V.2

CHAPTER 3

SYSTEM DESIGN

3.1 Home Screen

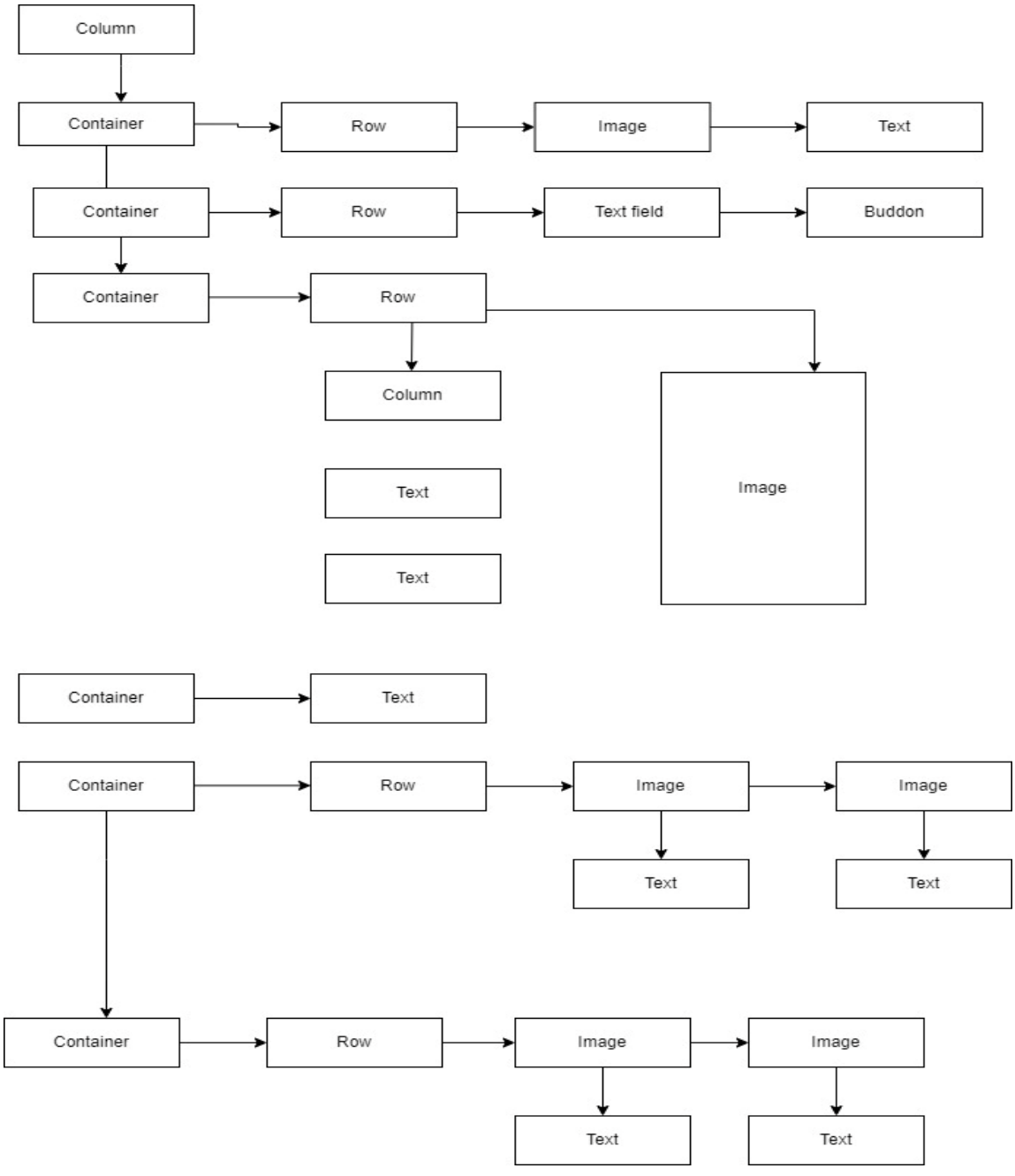

Figure 3.1: Home Screen Widget Tree

3.1 Widgets

Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the app. And the structure of the code of an app is a tree of widgets. Widgets describe what their view should look like given their current configuration and state. It includes a text widget, row widget, column widget, container widget, and many more.

Description of the Widgets Used:

- Scaffold – Implements the basic material design visual layout structure.
- AppBar – To create a bar at the top of the screen.
- Text - To write anything on the screen.
- Container – To contain any widget.
- Center – To provide center alignment to other widgets.

3.1.1 Scaffold

Scaffold is a class in flutter which provides many widgets or we can say APIs like Drawer, Snack Bar, BottomNavigationBar, FloatingActionButton, AppBar etc. Scaffold will expand or occupy the whole device screen. It will occupy the available space. Scaffold will provide a framework to implement the basic material design layout of the application.

3.1.2 AppBar

AppBar is usually the topmost component of the app (or sometimes the bottom-most), it contains the toolbar and some other common action buttons. As all the components in a flutter application is a widget or a combination of widgets. So AppBar is also a built-in class or widget in flutter which gives the functionality of the AppBar out of the box. The AppBar widget is based on *Material Design* and much of the information is already provided by other classes like Media Query, *Scaffold* as to where the content of the AppBar should be placed. Though the AppBar class is very flexible and can be easily customized, we can also use SilverAppBar widget which gives scrollable functionality to the app bar.

3.1.3 Text

The Rich Text widget is used to display text that uses various different styles. The displayed text is described using a tree of Text Span objects, each of which has its own associated style that is used for that *subtree*. Depending on the layout constraints the text might break across multiple lines or might all be displayed on the same line.

3.1.4 Container

Container class in flutter is a convenience widget that combines common painting, positioning, and sizing of widgets. A Container class can be used to store one or more widgets and position it on the screen according to our convenience. Basically, a container is like a box to store contents. A basic container element that stores a widget has a margin, which separates the present container with other contents. The total container can be given a border of different shapes, for example, rounded rectangles, etc. A container surrounds its child with padding and then applies additional constraints to the padded extent.

3.1.5 MaterialApp

MaterialApp is a predefined class in a flutter. It is likely the main or core component of flutter. We can access all the other components and widgets provided by Flutter SDK. Text widget, Dropdownbuttonwidget, AppBar widget, Scaffold widget, ListView widget, Stateless widget, Icon Button widget, TextField widget, Padding widget, ThemeData widget,etc. are the widgets that can be accessed using MaterialApp class. There are many more widgets that are accessed using MaterialApp class. Using this widget, we can make anattractive app.

CHAPTER 4

IMPLEMENTATION

4.1 Requirement Specifications

4.1.1 Hardware Requirements

- **CPU:** Intel i5 8th gen and above
- **RAM:** 8 GB
- **HDD:** 40 GB

4.1.2 Software Requirements

- **Operating System:** Windows 10 and above
- **IDE:** Visual Studio Code | Android studio Emulator
- **Front-end Language**: Dart

4.1.3 Flutter

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

The major components of Flutter include:

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

4.1.3.1 Dart Platform

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

4.1.3.2 Flutter Engine

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

4.1.3.3 Foundation Library

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

4.1.3.4 Design Specific Widgets

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

4.2 Implementation Details

4.2.1 Installation of Visual Studio Code

- Download VS code from https://code.visualstudio.com/download.
- Download the Visual Studio Code installer for suitable OS. Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). Then, run the file – it will only take a minute
- Accept the agreement and click "next."
- After accepting all the requests press finish button. By default, VS Code installs under: the desires path the user wishes to download

4.3 Discussion of code Segment

4.3.1 Login page Code

```
import 'package:barber_shop/home_screen.dart';
import 'package:flutter/material.dart';
import 'contants.dart';

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

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

class _LoginScreenState extends State<LoginScreen> {
  @override
  Widget build(BuildContext context) {
    return Material(
      color: primaryColor,
      child: Column(
        children: [
          Container(
            margin: EdgeInsets.fromLTRB(0, 40, 0, 0),
            height: 200,
            width: 200,
            child: FittedBox(
              child: Image.asset(
                'lib/assets/images/logo.png',
                fit: BoxFit.fill,
              ),
            ),
          ),
          Container(
            alignment: Alignment.center,
            margin: EdgeInsets.all(20),
            height: 40,
width: double.infinity,
            child: Text(
              'Welcome to Barder Shop',
              style: TextStyle(
                color: secondaryColor,
                fontSize: 30,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
                filled: true,
```

```
                contentPadding: EdgeInsets.all(15),
                fillColor: secondaryColor,
              ),
            ),
          ),
          Container(
            width: 500,
            height: 50,
            margin: EdgeInsets.fromLTRB(40, 20, 40, 20),
            child: ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(builder: (context) => HomeScreen()),
                );
              },
              style: TextButton.styleFrom(
                  primary: Colors.white,
                  backgroundColor: pinkColor,
                  shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(10),
                      side: BorderSide(width: 0))),
              child: Text(
                'Login',
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 18,
                  color: Colors.white,
                ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

4.3.2 Home screen code

```
import 'package:barber_shop/appointment_screen.dart';
import 'package:flutter/material.dart';
import 'contants.dart';

class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);
```

```
@override
Widget build(BuildContext context) {
  return Material(
    color: primaryColor,
    child: Column(
      children: [
        Container(
          child: Row(
            children: [
              Container(
                margin: EdgeInsets.fromLTRB(20, 60, 20, 0),
                height: 40,
                width:  40,
                child: FittedBox(
                  child: Image.asset('lib/assets/images/account.png'),
                  fit: BoxFit.fill,
                ),
              ),
        Container(
          child: Row(
            children: [
              Container(
                margin: EdgeInsets.fromLTRB(20, 10, 10, 10),
                width: 230,
                child: TextField(
                  decoration: InputDecoration(
                    border: OutlineInputBorder(
                      borderRadius: BorderRadius.zero,
                      borderSide: BorderSide(
                        width: 0,
                        style: BorderStyle.none,
                      ),
                    ),
                    filled: true,
                    contentPadding: EdgeInsets.all(15),
                    fillColor: secondaryColor,
                  ),
                ),
              ),
              Container(
                width: 100,
                height: 50,
                margin: EdgeInsets.fromLTRB(10, 10, 10, 10),
                child: ElevatedButton(
                  onPressed: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) => HomeScreen()),
```

```
                );
              },
              style: TextButton.styleFrom(
                  primary: Colors.white,
                  backgroundColor: pinkColor,
                  shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(10),
                      side: BorderSide(width: 0))),
              child: Text(
                'Search',
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize: 18,
                  color: Colors.white,
                ),
              ),
            ),
          ],
          GestureDetector(
            child: Container(
              margin: EdgeInsets.fromLTRB(30, 10, 10, 10),
              padding: EdgeInsets.all(20),
              height: 150,
              width: 150,
              decoration: BoxDecoration(
                  borderRadius: BorderRadius.circular(10), // radius of 10
                  color: secondaryColor // green as background color
                  ),
              child: Stack(
                children: [
                  Container(
                    margin: EdgeInsets.fromLTRB(20, 0, 20, 0),
                    height: 80,
                    width: 80,
                    child: FittedBox(
                      child: Image.asset("lib/assets/images/h2.png"),
                    ),
                  ),
                  Container(
                    margin: EdgeInsets.fromLTRB(10, 80, 10, 0),
                    alignment: Alignment.center,
                    height: 40,
                    width: 100,
                    child: Text(
                      "Hair care",
                      style: TextStyle(
                        color: primaryColor,
                        fontSize: 20,
                        fontWeight: FontWeight.w900,
                      ),
```

```
                        ),
                      ),
                    ],
                  ),
                ),
                onTap: () {
                  Navigator.push(
                    context,
                    MaterialPageRoute(
                        builder: (context) => AppointmentScreen()),
                  );
                },
              ),
            ],
          ),
        ),
      ],
    ),
  );
 }
}
```

4.3.3 Appointment page

```
import 'package:barber_shop/booking_con.dart';
import 'package:flutter/material.dart';
import 'contants.dart';

class AppointmentScreen extends StatefulWidget {
  @override
  State<AppointmentScreen> createState() => _AppointmentScreenState();
}

class _AppointmentScreenState extends State<AppointmentScreen> {
  bool _flag1 = true;
  bool _flag2 = true;
  bool _flag3 = true;
  bool _flag4 = true;
  bool _flag5 = true;
  bool _flag6 = true;
  bool _flag7 = true;
  bool _flag8 = true;
  bool _flag9 = true;
  bool _flag10 = true;
  bool _flag11 = true;

  @override
  Widget build(BuildContext context) {
    return Material(
```

```
color: primaryColor,
child: Column(
  mainAxisAlignment: MainAxisAlignment.start,
  children: [
    Container(),
    Container(
      margin: EdgeInsets.fromLTRB(30, 40, 30, 10),
      child: Text(
        "Appointment",
        style: TextStyle(
            fontSize: 30,
            color: secondaryColor,
            fontWeight: FontWeight.bold),
      ),
    ),
    Container(
      width: 380,
      margin: EdgeInsets.fromLTRB(20, 0, 0, 0),
      child: Row(
        children: [
          Container(
            margin: EdgeInsets.fromLTRB(0, 0, 20, 0),
            height: 40,
            width: 180,
            child: Row(
              children: [
                Container(
                  height: 40,
                  width: 40,
                  child: FittedBox(
                    child: Image.asset('lib/assets/images/account.png'),
                    fit: BoxFit.fill,
                  ),
                ),
                Container(
                  child: Column(
                    children: [
                      Container(
                        margin: EdgeInsets.fromLTRB(5, 0, 0, 0),
                        height: 20,
                        width: 130,
                        child: Text(
                          "Surya",
                          style: TextStyle(
                            fontSize: 18,
                            fontWeight: FontWeight.bold,
                            color: secondaryColor,
                          ),
                        ),
                      ),
                    ),
```

```
                    ],
                  ),
                ),
              ],
            ),
          )
        ],
      ),
    ),
    SizedBox(
      height: 20,
    ),
    Container(
      margin: EdgeInsets.fromLTRB(20, 0, 0, 0),
      child: Row(
        children: [
          Container(
            margin: EdgeInsets.fromLTRB(0, 0, 20, 0),
            height: 40,
            width: 180,
            child: Row(
              children: [
                Container(
                  child: Column(
                    children: [
                      Container(
                        margin: EdgeInsets.fromLTRB(5, 0, 0, 0),
                        height: 20,
                        width: 130,
                        child: Text(
                          "Samarth",
                          style: TextStyle(
                            fontSize: 18,
                            fontWeight: FontWeight.bold,
                            color: secondaryColor,
                          ),
                        ),
                      ),
                      Container(
                        margin: EdgeInsets.fromLTRB(5, 0, 50, 0),
                        child: Row(
                          children: [
                            Container(
                              child: Icon(
                                Icons.star,
                                color: secondaryColor,
                                size: 20,
                              ),
                            ),
```

```
                            Container(
                              child: Icon(
                                Icons.star,
                                color: secondaryColor,
                                size: 20,
                              ),
                            ),
                          ),
                        ),
                      ],
                    ),
                  ),
                ),
              ],
            ),
          ),
        Container(
          margin: EdgeInsets.fromLTRB(20, 10, 20, 30),
          height: 180,
          width: double.infinity,
          decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(30), // radius of 10
              color: secondaryColor // green as background color
              ),
          child: Column(
            children: [
              Container(
                margin: EdgeInsets.fromLTRB(10, 10, 150, 10),
                child: Text(
                  'Choose the data',
                  style: TextStyle(
                      color: primaryColor,
                      fontSize: 24,
                      fontWeight: FontWeight.w900),
                ),
              ),
              Container(
                margin: EdgeInsets.fromLTRB(20, 0, 10, 0),
                width: double.infinity,
                height: 110,
                child: Row(
                  children: [
                    Container(
                      child: Stack(
                        children: [
                          Container(
                            height: 110,
                            width: 50,
                            child: ElevatedButton(
```

```
                    onPressed: () => setState(() => _flag1 = !_flag1),
                    child: Container(
                      child: Text(
                        _flag1 ? '' : '',
                        style: TextStyle(
                            color: primaryColor,
                            fontSize: 16,
                            fontWeight: FontWeight.bold),
                      ),
                    ),
                    style: ElevatedButton.styleFrom(
                         primary: _flag1
                             ? Colors.white
                             : pinkColor, // This is what you need!
                        ),
                  ),
                ),
                Container(
                    child: Column(
                  children: [
                    Container(
                      margin: EdgeInsets.fromLTRB(8, 20, 0, 10),
                      child: Text(
                        'Mon',
                        style:  TextStyle(
                            color: primaryColor,
                            fontSize: 18,
                            fontWeight: FontWeight.w900),
                      ),
                    ),
                    Container(
                      child: Text(
                        '19',
                        style:  TextStyle(
                            color: primaryColor,
                            fontSize: 22,
                            fontWeight: FontWeight.w900),
                      ),
                    ),
                  ],
                )),
              ],
            ),
          ),
          SizedBox(
            width: 10,
          ),
          Container(
            child: Stack(
```

```
                children: [
                  Container(
                    height: 110,
                    width: 50,
                    child: ElevatedButton(
                      onPressed: () => setState(() => _flag3 = !_flag3),
                      child: Container(
                        child: Text(
                          _flag3 ? '' : '',
                          style: TextStyle(
                              color: primaryColor,
                              fontSize: 16,
                              fontWeight: FontWeight.bold),
                        ),
                      ),
                      style: ElevatedButton.styleFrom(
                            primary: _flag3
                                ? Colors.white
                                : pinkColor, // This is what you need!
                          ),
                    ),
                  ),
                ],
              ),
            ),
            SizedBox(
              width: 10,
            ),
            Container(
              child: Stack(
                children: [
                  Container(
                    height: 110,
                    width: 50,
                    child: ElevatedButton(
                      onPressed: () => setState(() => _flag4 = !_flag4),
                      child: Container(
                        child: Text(
                          _flag4 ? '' : '',
                          style: TextStyle(
                              color: primaryColor,
                              fontSize: 16,
                              fontWeight: FontWeight.bold),
                        ),
                      ),
                     style: ElevatedButton.styleFrom(
                            primary: _flag4
                                ? Colors.white
                                : pinkColor, // This is what you need!
                          ),
```

```
                ),
              ),
            ],
          ),
        ),
        SizedBox(
          width: 10,
        ),
        Container(
          child: Stack(
            children: [
              Container(
                height: 110,
                width: 50,
                child: ElevatedButton(
                  onPressed: () => setState(() => _flag5 = !_flag5),
                  child: Container(
                    child: Text(
                      _flag5 ? '' : '',
                      style: TextStyle(
                          color: primaryColor,
                          fontSize: 16,
                          fontWeight: FontWeight.bold),
                    ),
                  ),
                  style: ElevatedButton.styleFrom(
                        primary: _flag5
                            ? Colors.white
                            : pinkColor, // This is what you need!
                      ),
                ),
              ),
              Container(
                  child: Column(
                children: [
                  Container(
                    margin: EdgeInsets.fromLTRB(15, 20, 0, 10),
                    child: Text(
                      'Fri',
                      style:  TextStyle(
                          color: primaryColor,
                          fontSize: 18,
                          fontWeight: FontWeight.w900),
                    ),
                  ),
                ],
              )),
            ],
          ),
```

```
                ),
              ],
            ),
          ),
        ],
      ),
    ),
    Container(
      margin: EdgeInsets.fromLTRB(20, 0, 20, 10),
      height: 200,
      width: double.infinity,
      decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(30), // radius of 10
          color: secondaryColor // green as background color
          ),
      child: Column(
        children: [
          Container(
            margin: EdgeInsets.fromLTRB(10, 20, 150, 10),
            child: Text(
              'Choose the time',
              style: TextStyle(
                  color: primaryColor,
                  fontSize: 24,
                  fontWeight: FontWeight.w900),
            ),
          ),
        ],
      ),
    ),
    Container(
      width: 500,
      height: 50,
      margin: EdgeInsets.fromLTRB(40, 10, 40, 0),
      child: ElevatedButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => BookCon()),
          );
        },
        style: TextButton.styleFrom(
            primary: Colors.white,
            backgroundColor: pinkColor,
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10),
                side: BorderSide(width: 0))),
        child: Text(
          'Book Appointment',
```

```
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  fontSize:  24,
                  color: Colors.white,
                ),
              ),
            ),
          ),
        ],
      ),
    );
  }
}
```

This the appointment page code where it contains the details of the user who can book the appointment, by seeing rating of barber with timings and date.

CHAPTER 5

TESTING

5.1 Introduction

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

1. A software configuration that includes a software requirement specification, a design specification and source code.

2. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

5.2 Levels of Testing

5.2.1 Unit Testing

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

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

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

The units are combined and tested. Its main objective is to verify whether the major.

5.2.4 Validation Testing

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

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

5.2.5 Output Testing

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

5.2.6 User Acceptance Testing

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

UAT is done in the final phase of testing.

	Description	Input data	Expected Ouput	Actual Ouput	Status
1	Splash screen	More than 4 min	Accept and display the booking page	Booking page	Pass
2	Splash screen	Less than 4 min	Does not accept	Booking page is not displayed	Fail
3	Services	Selecting more than 1 services	Book appointment button get activated	Book now	Pass
4	Services	No selection	Deactivate the Book now button	Book Now button does not work	Pass

Table 5.1: Test Cases

CHAPTER 6

RESULTS

6.1 Splash screen

Figure 6.1: Splash screen

The Figure 6.1 shows the Splash screen and login screen of the application. This splash screen is visible for 4 seconds and later login page will be displayed.

6.2 Login screen

Figure 6.2: Login screen.

The Figure 6.2 shows Welcome to barber shop and login screen of the application.

By typing user name and password user can login

6.3 Home screen

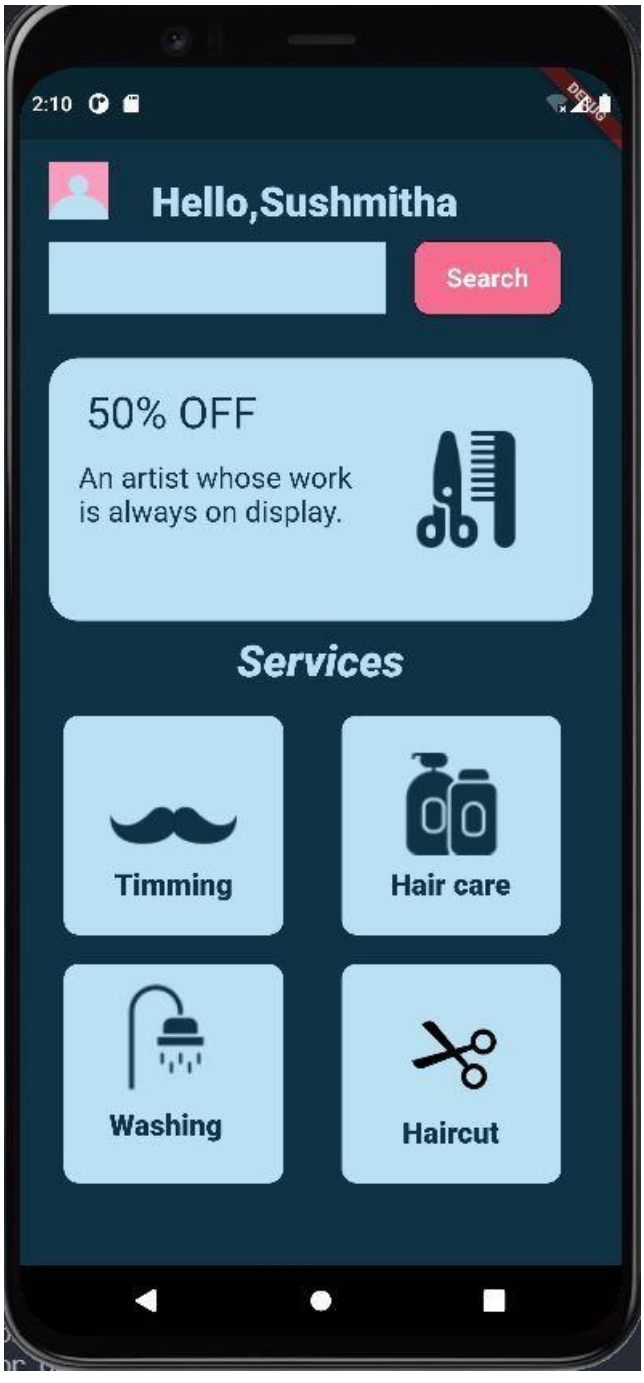

Figure 6.3: Booking screen with services.

Figure 6.3 contains Trimming, Hair Care, Washing and HairCut services user can select particularService then it will moove to next appointment page.

6.4 Appointment screen

Figure 6.4: Appointment screen

In Figure 6.4 User can select date and time, by clicking book appointment they can take appointment

6.5 Booking status

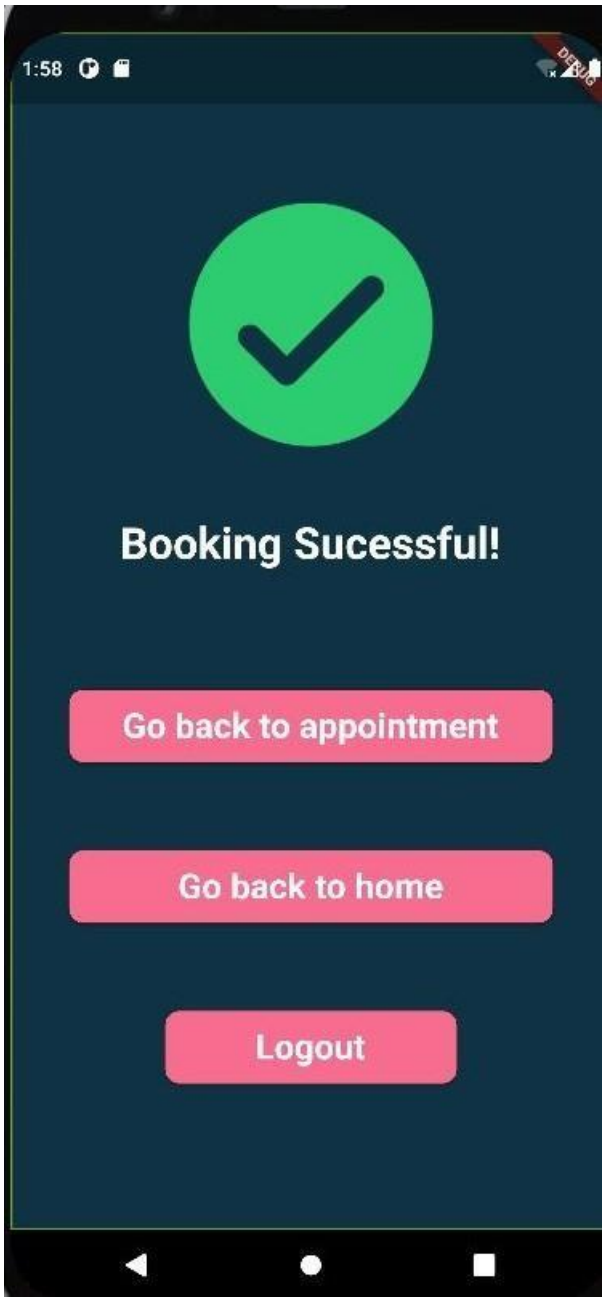

Figure 6.5: Booking status screen

Figure 6.5 displays Booking status and options with respect to next operations

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

Online booking is a faster, easier way for clients as well as Barber shop to take work and manage it according to their availability. Gone are the days where you had to manually manage sheets and registers to keep your calendar updated.

In this project we have created an application with an user friendly and creative user interface for the users so that the user will be happy and can easily use our application in an color full way.

Future Enhancements

- Secured payment system with various payment methods.
- Implementing the email or message service as the remainder of appointment
- Implementing the same features for more the one Barber shop.
- And also reschedule can be implemented.

REFERENCES

[1] https://flutter.dev/

[2] https://www.javatpoint.com/flutter

[3] https://www.geeksforgeeks.org/flutter-tutorial/

[4] https://medium.com/flutter

[5] https://www.pub.dev

[6] https://www.dribble.com