# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

**An Internship Project Report on**

## *"SALON BOOKING APP"*

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

By

NEHA B S
1RN18IS068

**Under the Guidance of**

**Mr. R Rajkumar**
Associate Professor
Department of ISE


## Department of Information Science and Engineering

## RNS Institute of Technology

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

**2020 -2021**

# 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 ***Online Food Recipe*** has been successfully completed by **Neha B S (1RN18IS068)** a bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 8th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

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

**External Viva**

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

# DECLARATION

We, **NEHA B S [USN: 1RN18IS068]** student of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Salon Booking app*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date :09-01-2022

**NEHA B S**

**(1RN18IS068)**

# ABSTRACT

Flutter is an open-source cross-platform mobile application development SDK created by Google. It is highly user-friendly and builds high-quality mobile applications. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia,Web platform.

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is an object-oriented, class-based, garbage-collected language with C -style syntax. Dart can compile to either native code or JavaScript.

Salon booking app is a mobile-based salon app with appointment scheduling functionality. ItConnects clients ,salons ,and stylists in an online community allowing users to browse salons and stylists ,book or cancel appointments. Users can also write and read reviews of salons and stylists. Salons can specify the stylists that works at their salons, as well as the services they offer. Salons can also book appointments for customers and can view and print schedules in convenient formats.

This project uses Flutter as front end tool. For appointment scheduling, app will integrate with the calender. 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 salon shop, gallery with has the hair styles that they had made before and writing and reading reviews for specified salons and/or services , appointment page which contains the appointment details of the particular person includes available time, total amount, user name .

# 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.

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

We place our heartfelt thanks to **Sowmya S K** Professor and HOD, 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, CEO, Enmaz** , 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.

**NAME: NEHA B S**
**USN: 1RN18IS068**

# Table of Contents

# LIST OF FIGURES

# ABBREVIATION

| | |
|---|---|
| FIRE | Flutter Italian Registry |
| FK | Flutter Kick |
| FSS | Flutter Suppression System |
| FCL | Flutter cycle Length |

# Chapter 1

# INTRODUCTION TO FLUTTER

## 1.1 HISTORY

Flutter is a 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 Flutter as 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 InteliJ that is fundamentally the same as Android Studio.

### 1.2. 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 nonconcurrently, to guarantee the UI stays responsive. On the customer side, MethodChannel (Programming interface) empowers sending msgs that relate to strategy calls. On the stage side, MethodChannel 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

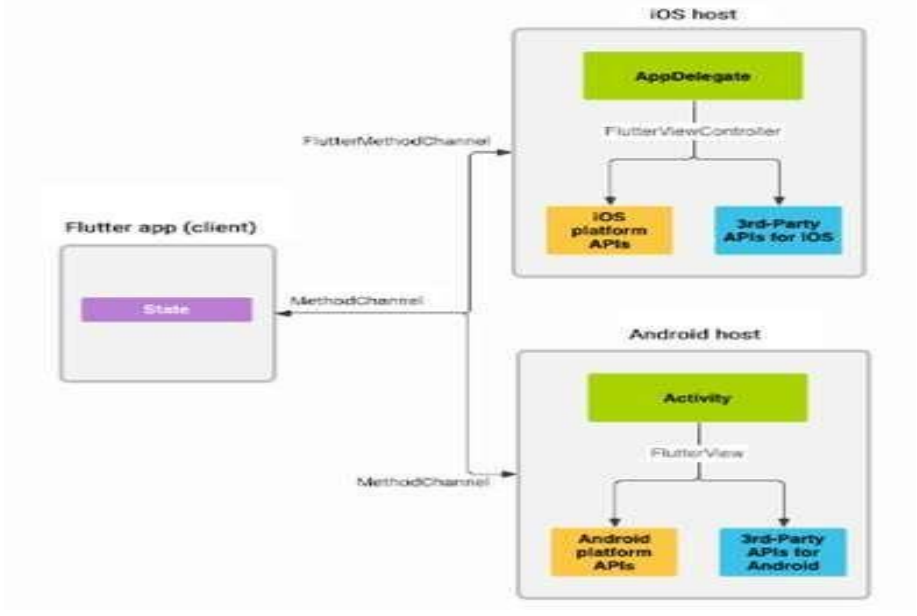


Fig 1. Architectural overview: platform channels

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 control and influence how the perspectives carry on, yet in addition handle and react to the client's activity. In this manner, it is urgent that Widgets requirement to perform quickly, including deliverring and quickening. Rather than use again OEM Widgets, similarly as what Respond Local does, Flutter group chooses to give its very hold Widgets. This implies Flutter, as a stage, gets the chance to choose how and when Widgets are delivered. As it were, Flutter shifts Widgets and deliverer from framework level into the app itself, which enables them to be

progressively adaptable and extensible. In any case hold the Widgets and deliverer inside the app makes the extent of use bigger .

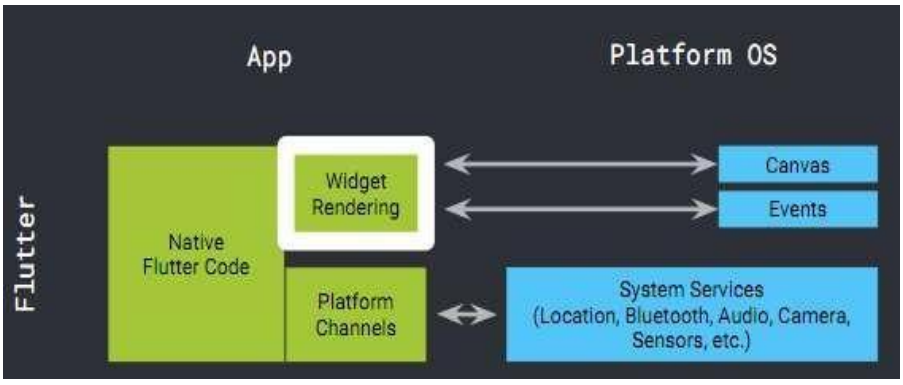


Fig.2 Widget Rendering

### 1.4 Layout

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.

### 1.5 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<br>2. Git for Windows 2.x (Here, Use Git from Windows Command Prompt option). |
| **SDK** | Flutter SDK for Windows |
| **IDE** | Android Studio (Official) |

1.5 Flutter requirement

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

**Step 8:** Again, set up an android emulator or connect an android device to the system for developing an android application.

**Step 9:** 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.

# 1.6 INTRODUCTION TO THE PROJECT

## 1.6.1 OVERVIEW OF THE PROJECT

Gone are the days when Salon and Spa 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 salon 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

Salon booking app is one where the user can book there slot and even they can book the best stylist avaliable at there salon by reviewing the feedback by the previous users.

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. Manifest a healthy relationship with the clients
2. Build and grow the salon revenue
3. Monetary savings
4. Centralized Information system
5. Email and SMS Remainders
6. Valuable Insight into the salon

# Chapter 2

# Literature Review

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 architectues 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 beauty parlour is a service-sector, so the success of beauty parlour depends on thesatisfaction of customer. Definition of customer satisfaction has been widely discussed from theview of many researchers & organizations who increasiningly desire to measure it. The study of Social policy (2007) conceptualize that satisfaction is based on custpmer's experience of bothcontact 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- pecific satisfaction and cumulative satisfaction. Transaction-pecific satisfaction is a customer evoluation of her particular experience and reaction to a particular service encounter (Croning &Tailor, 1992; Boshoff & Gray.2004). Cumulative satisfaction refers to the customer overallevaluation 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

# ANALYSIS

## 3.1 Software requirement specification

### 3.1.1 Introduction

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

#### 3.1.1.1 Flutter

Flutter is an open source framework to create high quality, high performance mobile applications across mobile operating systems - Android and iOS. It provides a simple, powerful, efficient and easy to understand SDK to write mobile applications in Google's own language, *Dart*. Flutter also offers many ready to use widgets (UI) to create a modern application. These widgets are optimized for mobile environments and designing the application using widgets is as simple as designing HTML.Flutter widgets also supports animations and gestures.

#### 3.1.1.2 Visual Studio

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.It is based on the Electron framework which is used to develop Node.js Web applications that run on the Blink layout engine. It allows users to open one or more directories, which can then be saved in workspaces for future reuse.Visual Studio Code can be extended via extensions, available through a central repository. A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol. Visual Studio Code includes multiple extensions for FTP, allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software.

#### 3.1.1.3 Dart

Dart is a client-optimized language for fast applications on any platform.Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. Dart is

an object-oriented, class-based, garbage-collected language with C -style syntax. Dart can compile to either native code or JavaScript.

## 3.2. Requirements

### 3.2.1 Functional Requirements

1. Splash Screen : The loading screen needs to be navigated to the login screen with time gap of few seconds
2. Booking screen: Gives the information about the salon services.
3. Appointment screen: The information about the booking details.

### 3.2.2 Non-Functional Requirements

1. The concept of splash screen or Launch Screen is implemented using the built in method Timer().
2. Floating Action Button and its attribute onSelect() helps to implement the features on increment and decrement of the quantity.
3. Models are designed to display images,cost and to keep count of the number of products the user wants to purchase
4. Navigation has to implemented to traverse across the screens . The class Navigator has a lot of methods where each perform different actions . The route widget has all the pages or screens in the Flutter application

### 3.2.3 Software and Hardware Requirements

Operating System : Windows10
Programming Language : Dart
Processor : Intel Core i5 7th Gen
Speed : 2.33 Ghz
RAM : 8GB
Software Development Kit : Flutter
IDE : Visual Studio or Android Studio(Official)
Tools : Windows Powershell , Git

## 3.3 General Description

### 3.3.1 Assumptions and Dependencies

Flutter supports using shared packages contributed by other developers to the Flutter and Dart ecosystems. This allows quickly building an app without having to develop everything from scratch.

A Flutter app can depend on a plugin via a file system path: dependency. The path can be either relative or absolute. Relative paths are evaluated relative to the directory containing pubspec.yaml.

**Scoped Model:**

A set of utilities that allow you to easily pass a data Model from a parent Widget down to its descendants. In addition, it also rebuilds all of the children that use the model when the model is updated. This library was originally extracted from the Fuchsia codebase.

**Path_provider:**

A Flutter plugin for finding commonly used locations on the filesystem. Supports Android, iOS, Linux, macOS and Windows. Not all methods are supported on all platforms.

# Chapter 4

# SYSTEM DESIGN

## 4.1 Introduction

The system design can be represented as a widget tree. Flutter widgets are built using a modern framework that takes inspiration from React. The central idea is that can build one UI out of widgets. Widgets describe what their view should look like given their current configuration and state. When a widget's state changes, the widget rebuilds its description, which the framework diffs against the previous description in order to determine the minimal changes needed in the underlying render tree to transition from one state to the next.

## 4.2 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.

### 4.2.1 Scaffold

Scaffold is a class in flutter which provides many widgets or we can say APIs like Drawer, SnackBar, 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.

### 4.2.2 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 MediaQuery, *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. Or we can create our own custom app bar from scratch.

### 4.2.3 Text

The RichText widget is used to display text that uses various different styles. The displayed text is described using a tree of TextSpan 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.

### 4.2.4 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.

### 4.2.5 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, StatelessWidget, StatefulWidget, IconButton 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 an attractive app.

### 4.3 System Architecture

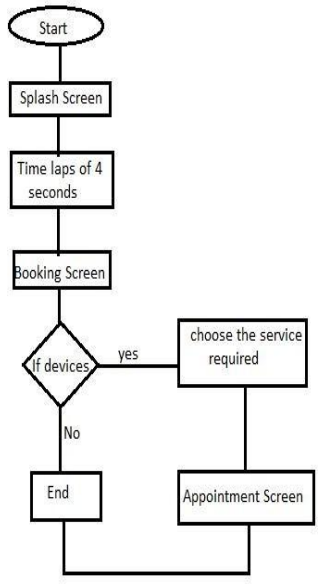


Fig.4.3 System Architecture

The application starts with the splash screen. It contains the company logo which is displayed for about 4 seconds.

This is followed by the booking screen .It contains the details about the services that are available.If the user does not like any services then the user can exit. If the user like the services then the user can choose the required services and then take to appointment screen .

# Chapter 5

# DETAILED DESIGN

## 5.1 Widget tree for Splash Screen

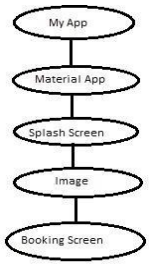


Figure 4.1 Widget tree for Splash Screen

Myapp is the top most part in the flutter inside the myapp we have materialapp which the pre defined function in flutter. It contains the splashscreen which is an image and it is directed to booking screen

## 5.2 Widget tree for Booking Screen

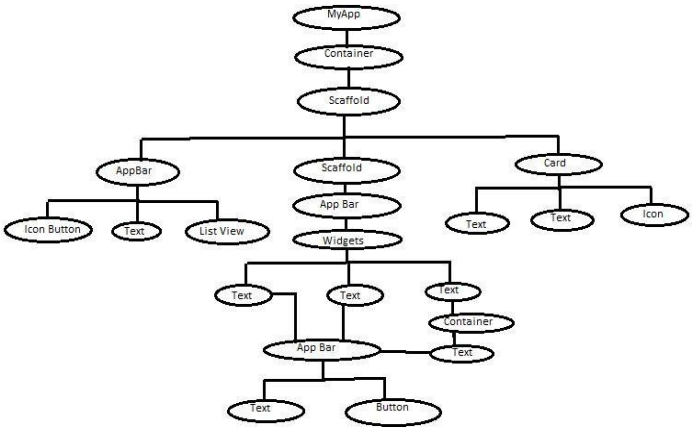


Figure 4.2 Widget tree for Booking Screen

Myapp class contains container class used to store more then one widgets and it contains the scaffold which many other widgets such as appbar, text, listview, row, coloum, icon, card etc.

### 5.3 Widget tree for Appointment Screen

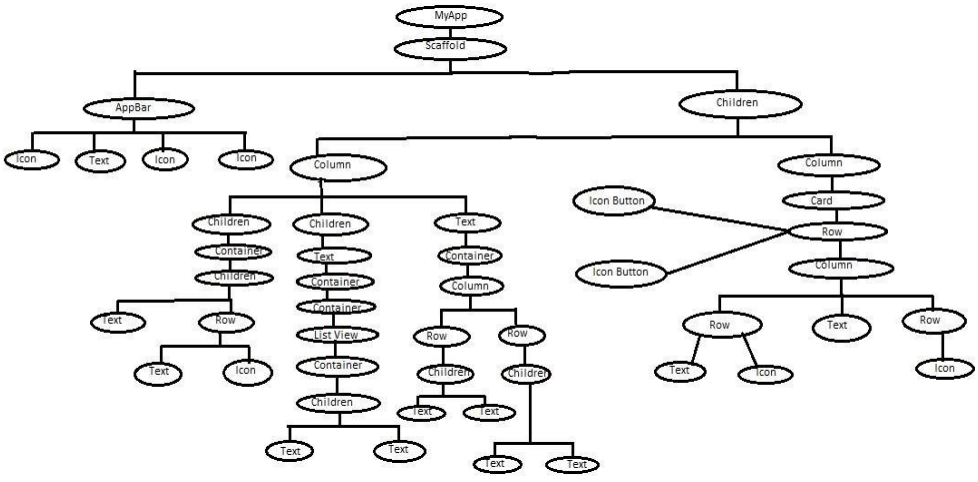


Figure 4.3 Widget tree for Appointment Screen

Myapp class contains the scaffold which many other widgets such as appbar, text, listview, row, coloum, icon, card ,childern etc.

# Chapter 6

## IMPLEMENTATION

## 6.1 Introduction

### 6.1.1 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 central idea behind Flutter is the use of widgets. It's by combining different widgets that developers can build the entire UI. Each of these widgets defines a structural element (like a button or menu), a stylistic element (a font or color scheme), a layout aspect (like padding), and many others. Flutter also provides developers with reactive-style views. To avoid performance issues deriving from using a compiled programming language to serve as the JavaScript bridge, Flutter uses Dart. It compiles Dart ahead of time (AOT) into the native code for multiple platforms. That way, Flutter can easily communicate with the platform without needing a JavaScript bridge that involves a context switch between the JavaScript realm and the native real.

### 6.1.2 Visual Studio Code

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.It is based on the Electron framework which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services). Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language. It supports a number of programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette. Visual Studio Code can be extended via extensions, available through a central repository. This includes additions to the editor[23] and language support.[21] A notable feature is the ability to create extensions that add support for new languages, themes, and debuggers, perform static code analysis, and add code linters using the Language Server Protocol. Visual Studio Code includes multiple extensions for FTP,

allowing the software to be used as a free alternative for web development. Code can be synced between the editor and the server, without downloading any extra software. Visual Studio Code allows users to set the code page in which the active document is saved, the newline character, and the programming language of the active document. This allows it to be used on any platform, in any locale, and for any given programming language.

## 6.2 Implementation Details

### 6.2.1 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

## 6.3 Pseudo Codes

### 6.3.1 Navigation Code

```
Widget build(BuildContext context)
  {return MultiProvider(
   providers:
    [ ChangeNotifierProvider<MapDataProvider>(
        create: (_) => MapDataProvider()),
   ],
   child:
    MaterialApp( title:
    "Barber App",
    theme: ThemeData(
      // Define the default brightness and colors.
      brightness:  Brightness.dark,
      primaryColor: Colors.blueGrey[900],
      accentColor: Colors.cyan[600], textTheme:
      TextTheme(
       headline1:
          TextStyle(fontSize:
          25.0,
          fontWeight: FontWeight.bold,
```

```
        color: Colors.white),
    headline2: TextStyle(fontSize: 20.0, fontWeight: FontWeight.normal),
```

```
          headline6: TextStyle(fontSize: 20.0, fontStyle: FontStyle.normal),
          bodyText2: TextStyle(
            fontSize: 14.0,
          ),
        ),
      ),
      routes: {
        '/bookingScreen': (context) => BookingScreen(),
        '/mappage': (context) => Mappage(),
        '/appointmentScreen': (context) => AppointmentScreen(),
      },
      home: SplashScreen(),
      debugShowCheckedModeBanner: false,

class SplashScreen extends StatefulWidget
  {@override
  _SplashScreen createState() => _SplashScreen();
}

class _SplashScreen extends State<SplashScreen>
  {Widget defaultPage = Container();
  void checkSharedPrefs() async {
    var sharedPrefs = await SharedPreferences.getInstance();
    if (sharedPrefs.containsKey("firstTime")) {
      defaultPage = Container();
    }
  }
  void  initState()
    { super.initState();
    checkSharedPrefs();
    Timer(Duration(seconds: 4), () {
      Navigator.of(context).pushReplacement(MaterialPageR
        oute(
          //builder: (BuildContext context) => AppointmentScreen(),
          builder: (BuildContext context) => BookingScreen(),
        ),

      );
    });
```

```
 void _navigateToNextScreen(BuildContext context) {

 Navigator.of(context).push(MaterialPageRoute(builder: (context) => Mappage()));
}
}
Widget build(BuildContext context)
 {return Scaffold(
     backgroundColor: Colors.blueGrey[900],
  body: Column(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    crossAxisAlignment: CrossAxisAlignment.center,
    children:
    <Widget>[ Image.asset("assets/splash_screen.gif",

      height: (60 / 100) * MediaQuery.of(context).size.height,
      width: MediaQuery.of(context).size.width,
     ),
     CircularProgressIndicator(strokeWidt
      h: 4,
      backgroundColor: Colors.amberAccent[400],

     )
          ],
        ),
      );
     }
    }
```

This is the navigation code where the screen changes from one screen to other screen. Here initial screen is splash screen where the screen appears according to the time we mentioned in the code the screen navigate from splash screen to booking page and from booking page to appointment page.

### 6.3.2 Booking page

```
Widget build(BuildContext context)
   {return Container(
    child: Stack(
     fit: StackFit.expand,
```

```
children: <Widget>[


 Image.network(
  'https://thumbs.dreamstime.com/b/blue-sky-clouds-abstract-art-background-watercolor-digital-artwork-136551201.jpg',
  fit: BoxFit.fitHeight,
 ),
 Scaffold(
  endDrawer: Drawer(),
  appBar:
  AppBar( title: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: <Widget>[
     SizedBox(width: 50,),
     Text(
      "Booking",
      style: Theme.of(context).textTheme.headline6,
     ),
     IconButton(
      icon: Icon(Icons.notifications),
      onPressed: () {},
     ),
    ],
   ),
   leading: IconButton(
    icon: Icon(Icons.arrow_back),
    onPressed: () {},
   ),
  ),
  backgroundColor: Colors.transparent,
  body: ListView(
   children:
    <Widget>[Card(
     margin: EdgeInsets.fromLTRB(40, 20, 40, 30),
```

```
shape:
  RoundedRectangleBorder( borderRadius:
  BorderRadius.circular(50.0)),
color: Theme.of(context).primaryColor,
elevation: 10,
child: Padding(
 padding: const EdgeInsets.fromLTRB(0, 50, 0, 10),
 child: Column(
  children:
   <Widget>[CircleA
   vatar(
    backgroundColor: Colors.white,
    radius: 48.0,
    child:
     CircleAvatar(radi
     us: 40,
     backgroundImage:
NetworkImage( 'https://images.unsplash.com/photo1506794778202cad84cf
45f1d?ixlib=rb1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=1868&q=
80') ),   ),
   SizedBox(
    height: 7,
   ),
   Text(
    'John Doe',
    style: Theme.of(context).textTheme.headline1,
   ),
   SizedBox(
    height: 7,
   ),
   Text('Thane',
    style: Theme.of(context).textTheme.headline2,
   ),
  ],child: Scaffold(
 backgroundColor: Colors.transparent,
```

```
appBar: AppBar(

shape: RoundedRectangleBorder(
```

```
        borderRadius:
            BorderRadius.only(     topRight:
            Radius.circular(40.0),    topLeft:
            Radius.circular(40.0))),
      title: TabBar(
       indicatorSize: TabBarIndicatorSize.label,
       indicatorWeight: 3,
       labelColor: Colors.white,
       unselectedLabelColor: Colors.white38,
       controller: _tabController,
       indicatorColor: Colors.white,
       tabs:
        <Widget>[Tab
        (
           child:
         Text("Servic
         es",
         style: TextStyle(fontSize: 20),
        )),
        Tab(
           child:
         Text("Galler
         y",
         style: TextStyle(fontSize: 20),
        )),
        Tab(
           child:
         Text("Revie
         ws",
         style: TextStyle(fontSize: 20),
        ))
       ],
      ),
```

```
),
body:
  TabBarView( controller:
  _tabController,children:
  <Widget>[ ServiceBody(),
```

```
GalleryBody(), Container(child:
Text("hello")),
```

This the code for booking page which contains services that are available in the salon ,the gallery which contains the photos and the feedback page. It has the arrow button which navigates back to the previous page it also contains the notification button when the user press the icon it notifes the user about the slots that are available in salon. It also contains the list of services that are available. Once the user opt the services the booking button get activated and the booking button take us to the appointment page.

### 6.3.3 Appointment page

```
Widget build(BuildContext context)
{return Scaffold(
 backgroundColor: Colors.white,
 appBar: AppBar(
  elevation: 0,
  backgroundColor: Colors.orangeAccent[100],
  leading: IconButton(
   icon:
    Icon( Icons.arrow
    _back,
   ),
   onPressed: () => Navigator.of(context).pop(),
  ),
  title: Text("Appointment"),
  centerTitle: true,
  actions:
   <Widget>[IconB
   utton(
    icon:
     Icon( Icons.notific
     ations,
    ),
    onPressed: () {},
   ),
   IconButton(
    icon:
    Icon( Icons.
    sort,
    ),
    onPressed: () => {},
   ),
  ],
 ),
body: Stack(
  children:
   <Widget>[ SingleChildSc
```

```
rollView(child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: <Widget>[
      Card(
```

```
            elevation: 10,
            color: Colors.orangeAccent[100],
            margin: EdgeInsets.all(0),
            child:
              Column( children:
              <Widget>[SizedBo
              x(
                 height: 15,
               ),
               Row(
                 mainAxisAlignment: MainAxisAlignment.end,
children: <Widget>[
                   Column(
                    mainAxisSize: MainAxisSize.min,
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: <Widget>[
                     Row(
                      children:
                       <Widget>[Text(
                        "John Doe ",
                        style: TextStyle(
                            fontSize: 20,
                            fontWeight: FontWeight.bold),
                       ),
                       Icon( Icons.
                        done,size:
                        15,
                       )
                     ],
                    ),
                    SizedBox(
                     height: 5,
                    ),
                    Text(
                     "Thane, India",
                            style:
                      TextStyle(fontS
                            ize: 14,
                     ),
                    ),
                    SizedBox(
                     height: 5,
                    ),
                    Row(
                     children: <Widget>[
                      for (int i = 0; i < 5; i++)
                       Icon(
                        Icons.star,
                        size: 17,
                       )
                     ],
                    ),
```

```
SizedBox(
  height: 7,
```

```
      ),
    ],
  ),
  SizedBox(
    width: 50,
  ),  IconButton(
    icon: Icon(Icons.share),
    onPressed: () {},
  ),
  IconButton(
    icon: Icon(Icons.favorite),
    onPressed: () {},
```

This the appointment page code where it contains the details of the user who booked the appointment ,timings , the services user had booked for himself and the total amount he required to pay for the services. This page also contains the user details includes name of the user and the place. The user can also add the page to the favourite and also he could share the particular page to other users.

### 6.3.4 gallarybody

```
class CategoryModel
{ List<Widget> models;
List<String> url = [
   'https://images.pexels.com/photos/792381/pexels-photo-
792381.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/227675/pexels-photo-
227675.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/1193743/pexels-photo-
1193743.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/1149831/pexels-photo-
1149831.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/1624487/pexels-photo-
1624487.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/556667/pexels-photo-
556667.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/1413412/pexels-photo-
1413412.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
   'https://images.pexels.com/photos/459225/pexels-photo-
459225.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
```

```
  'https://images.pexels.com/photos/853168/pexels-photo-
853168.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
  'https://images.pexels.com/photos/220201/pexels-photo-
220201.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
  'https://images.pexels.com/photos/1939485/pexels-photo-
1939485.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
  'https://images.pexels.com/photos/1008155/pexels-photo-
1008155.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
  'https://images.pexels.com/photos/853199/pexels-photo-
853199.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800',
  'https://images.pexels.com/photos/278887/pexels-photo-
278887.jpeg?auto=compress&cs=tinysrgb&fit=crop&h=1200&w=800'
 ];
 Widget model(int i) {
  //return Stack(
   //children:
    <Widget>[return
    ClipRRect(
     borderRadius: BorderRadius.circular(8),
     child: CachedNetworkImage(
      imageUrl: url[i],
      height: 85,
      width: 85,
      fit: BoxFit.cover,
     ),
    );
 }
}
class GalleryBody extends StatefulWidget
 {@override
 _GalleryBodyState createState() => _GalleryBodyState();
}
class _GalleryBodyState extends State<GalleryBody>
 {@override
 Widget build(BuildContext context)
```

```
{return Container(
```

```
      color: Colors.black12,
      child:
      GridView.count(child
      AspectRatio:1.1,
      crossAxisCount: 3,
        padding: EdgeInsets.fromLTRB(15, 20, 15, 20),
        children: List.generate(14, (index) {
          return Center(child: CategoryModel().model(index));
        }),
      ),
    );
  }
}
```

In some cases, it's handy to cache images as they're downloaded from the web, so they can be used offline. For this purpose, use the cached_network_image package.In addition to caching, the cached_network_image package also supports placeholders and fading images in as they're loaded.

# Chapter 7

# TESTING

## 7.1 Introduction

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

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.

## 7.2 Levels of Testing

### 7.2.1 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 is taking the inputs in the correct format or not.

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

### 7.2.2 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where parts of the program is working fine or not. This testing can be done by choosing the optionsin the program and by giving suitable inputs.

### 7.2.3 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code isnot a pre-requisite and is done by the testing team. System testing is done after integrationtesting is complete. System testing should test functional and non-functional requirementsof the software.

### 7.2.4 Validation Testing

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

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

### 7.2.5 Output Testing

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

### 7.2.6 User Acceptance Testing

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

UAT is done in the final phase of testing.

| | 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 | Pass |
| 3 | Services | Selecting more than 1 services | Book now button get activated | Book now | Pass |
| 4 | Services | No selection | Deactivate the Book now button | Book Now button does not work | Pass |

Table 7.1 Test Cases

# Chapter 8

# RESULTS

## 8.1 Splash screen

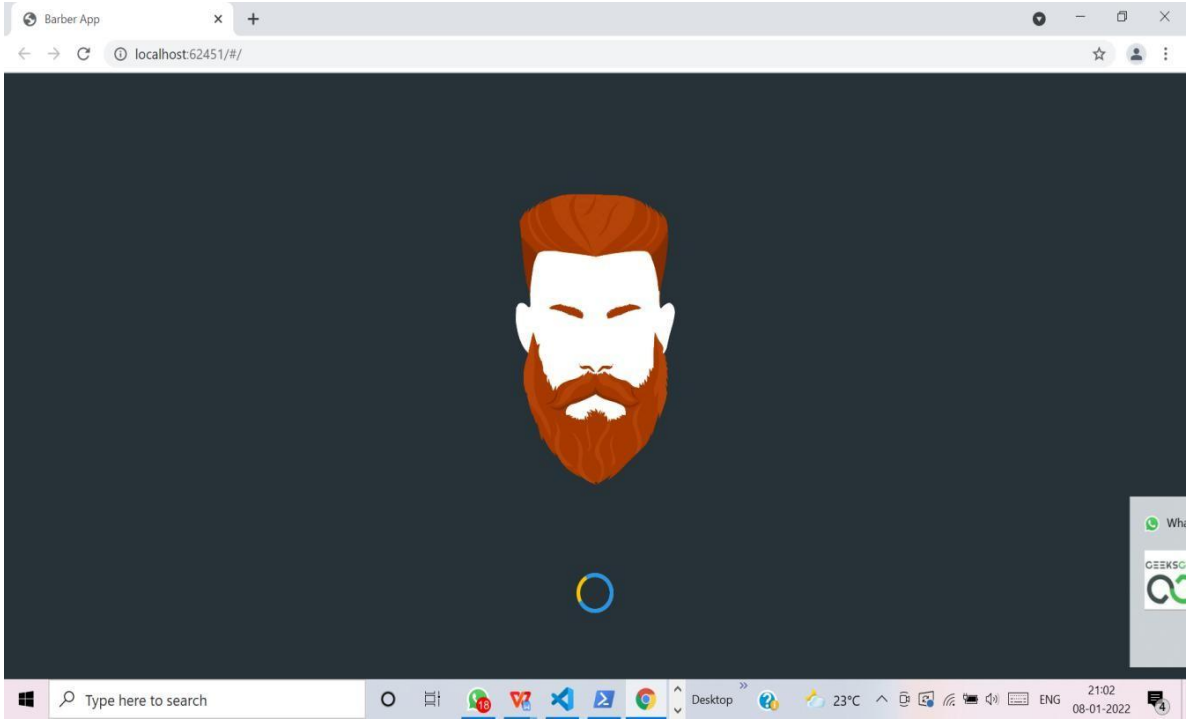


Figure 8.1 Splash screen

The Figure 7.1 shows the Splash screen of the application. This screen is visible for 4 seconds and later the user is moved to booking page.

## 8.2 Booking screen

The figure 7.2 shows the booking screen that contains services that are available, gallery and reviews. This screen shows the services that are available.

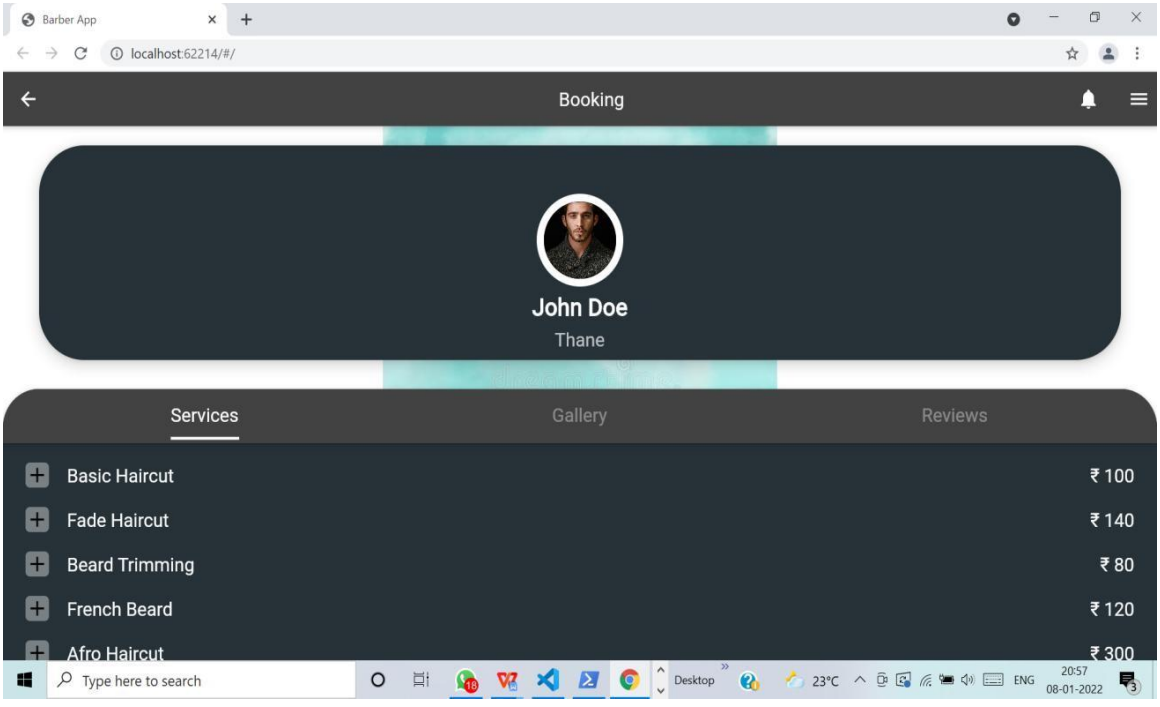


Figure 8.2 Booking screen with services

## 8.2.1 Booking screen with gallery

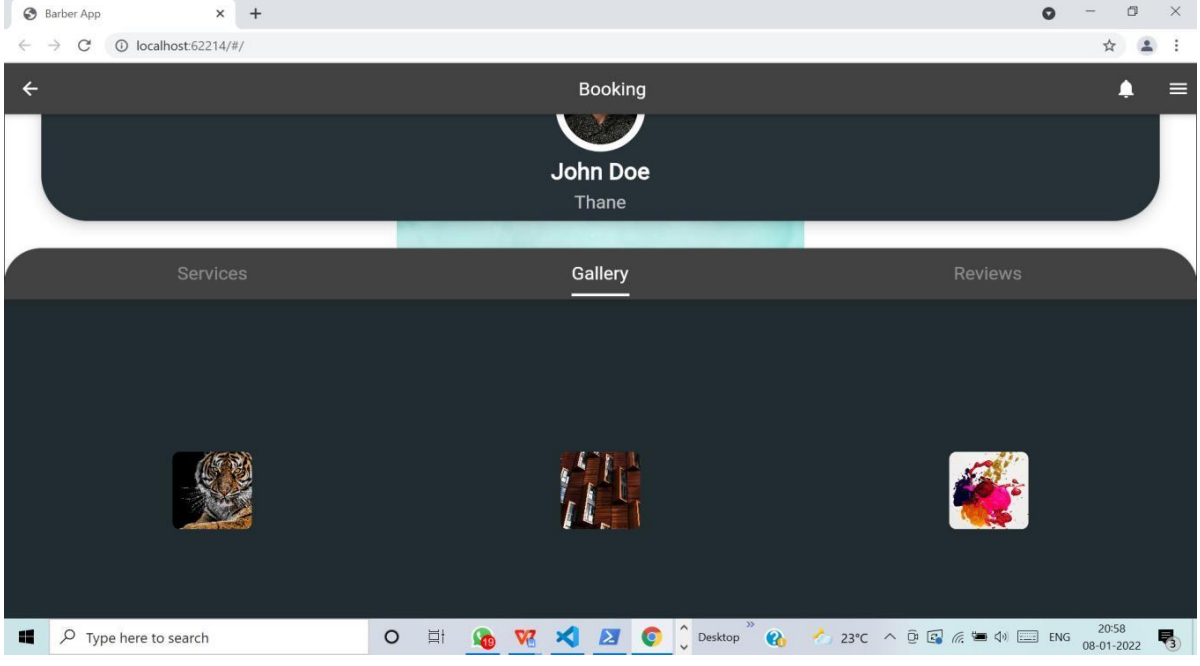


Figure 8.2.1 Booking screen with gallery

### 8.2.2 Booking screen with Feedback

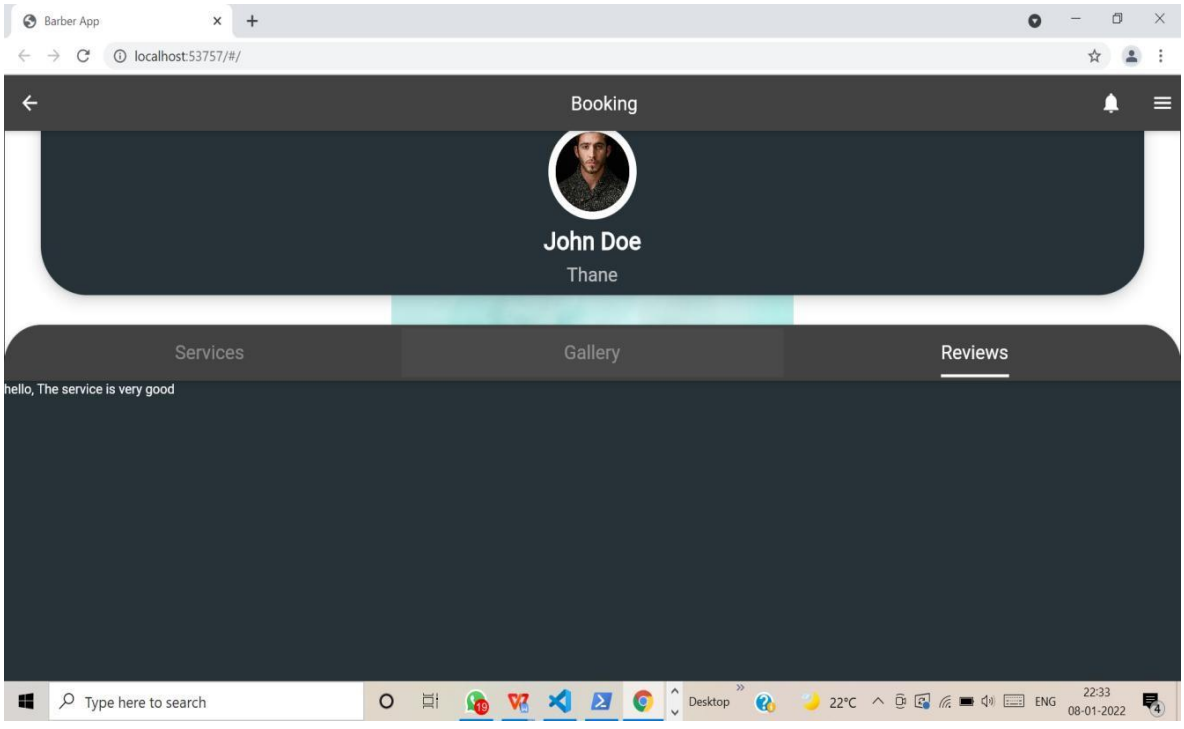


Figure 8.2.2 Booking screen with Feedback

## 8.3 Appointment Screen

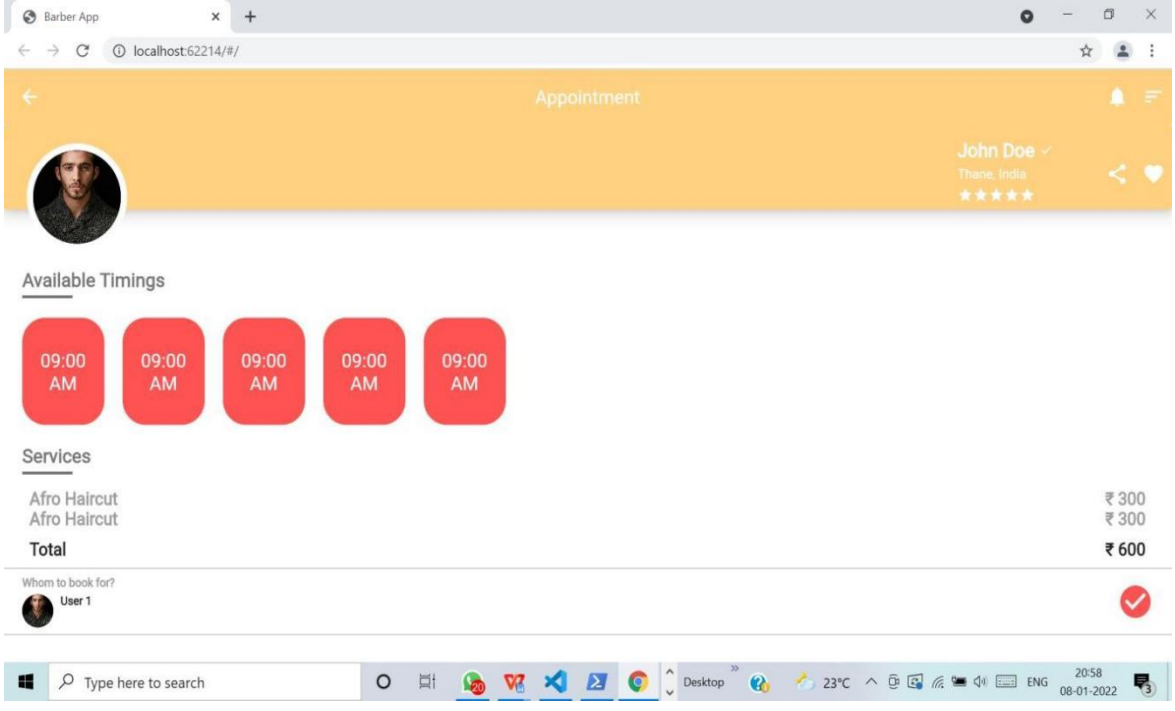


Figure 8.3Appointment screen

# Chapter 9

## CONCLUSION AND FUTURE ENHANCEMENTS

Online booking is a faster, easier way for clients as well as salon businesses 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.

## Future Enhancements

- Login and sign-up page can be implemented in order to secure the application.
- 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 salon shop.

# REFERENCES


[1] https://flutter.dev/

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

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

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