VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report
on

Doctor Consultation Application

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

Submitted By

Abhilasha A
1RN18IS002

Under the Guidance of

Mrs. Kusuma R

Assistant Professor
Department of ISE

ESTD:2001
An Institute with a Difference

Department of Information Science and Engineering

RNS Institute of Technology

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

2021-2022

RNS INSTITUTE OF TECHNOLOGY

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

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***Doctor Consultation Application*** has been successfully completed by **Abhilasha A(1RN18IS002)** 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	Professor and HoD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

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

DECLARATION

I, **Abhilasha A[USN: 1RN18IS002]** student of VIII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***Doctor consultation application*** has been carried out by us and submitted in partial fulfilment of the requirements for the *VIII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

Abhilasha A (1RN18IS002)

ABSTRACT

Electronic doctor consultation is an aspect of telemedicine which involves remote communication between patients and clinicians, or between clinicians and specialists.

E-consult was one of the earlier applications which primary care physicians could use to consult specialists. The widespread use of Electronic health records is reducing the need for specialist applications of this kind. There is developing interest in the application of such techniques in mental health.

This application as described above, can lead to error free, secure, reliable and fast management systems. Getting an appointment with care specialists can be difficult. But electronic consultation has stepped in to provide medical diagnosis or evaluation that can be carried out over the phone from the comfort of your home which uses electronic consultation through online platforms or mobile apps and request a virtual visit with doctors, practitioners, and therapists.

Basically, the project describes how to manage and have good performance and better services for the clients. The single application which caters all the details of services along with detailed description that are offered to the clients or end users by the company. The existing application has been enhanced to make the application user friendly with various operations.

.

ACKNOWLEGMENT

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

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

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

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

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

We place our 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 and CEO, Enmaz engineering services**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully

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

Date: _____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

UI: User Interface

FK: Flutter Kick

IoT: Internet of Things

FCL: Flutter Cycle Length

AOT: Ahead of Time

SDK: Software Development Kit

CHAPTER 1
INTRODUCTION

1.1 Introduction to Flutter

Flutter is Google's Mobile SDK to build native iOS and Android, Desktop (Windows, Linux, macOS), Web apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. They are structural elements that ship with a bunch of material design-specific functionalities and new widgets can be composed out of existing ones too. The process of composing widgets together is called composition. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

1.2 History

Flutter launched as a project called Sky which at the beginning worked only on Android. Flutter's goal is enabling developers to compile for every platform using its own graphic layer rendered by the Skia engine. Here's a brief presentation of Flutter's relatively short history.

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, this allows you to create a native mobile application with only one code. It means that you can use one programming language and one codebase to create two different apps (IOS and Android).

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

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

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with new Canvas Kit renderer and web specific widgets, early-access desktop application support for Windows, macOS, and Linux and improved Add-to-App APIs.[9] This release included sound null-safety, which caused many breaking changes and issues with many external packages, but the Flutter team included instructions to mitigate these changes as well.

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

1.3 Framework-Architecture

The major components of Flutter include:

- Dart platform
- Flutter engine
- Foundation library
- Design-specific widgets
- Flutter Development Tools (DevTools)

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

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

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.[10] The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

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.

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.

CHAPTER 2

LITERATURE SURVEY

2.1. Background

Consultation can be defined as the communication process between a patient and a physician on a medical issue (Internet consultation in medicine, 2006). Pawlikowska et al (2016) Explained that the medical consultation is a two-way encounter between a doctor and a patient. This may be initiated by a patient when they are ill or by a doctor when instituting preventive medicine or screening. The medical consultation is being done by the patient booking an appointment with the physician by physical appearance of the patient at the health centre which at time lead to queue as many might also be waiting to be responded to by the physician

Medical appointment can be seen as a means of patient booking and securing a particular time with the doctor either for medical check-up or for medical consultation. Online medical appointment scheduling helps to manage and effectively utilize the time spent at the health centre in a way that the patient is assigned a time which is visible for both the patient and the doctor. Large waiting times at hospital outpatient clinics are a cause of dissatisfaction to patients, cause additional stress to hospital staff, increase the risk of contagion and add complications for patients with medical conditions. Reducing waiting times and surgeon idle time improves the quality of service and efficiency of a hospital (Strahl, 2015).

2.2. Related works

Rinder et al., (2012) proposes a Healthcare Scheduling by Data Mining: Literature review and future directions. The article presents a systematic literature review of the application of industrial engineering methods in healthcare scheduling, the research work defines two categories of healthcare scheduling which are; work/patient scheduling and provider resource scheduling, Patient appointments are scheduled based on the number of appointments slots available. The number of appointments available is established based on the type of work, such as regular visits, follow-up visits, tests and procedures, education sessions, and the number of providers available by hour and day of the week. The paper described application of data modelling technique to improve scheduling in healthcare, and the modelling technique had to be a data mining technique. However, none of the methods modelled no-show and walk-in’s

patient behaviour, also the research work should include more variables related to patient and/or environment.

Mardiah & Basri (2013) researched Analysis of Appointment System to Reduce Outpatient Waiting Time at Indonesia's Public Hospital. The research aimed to provide a study of the major causes of patients length of time for medical treatment in an outpatient clinic at one of Indonesian public hospital and also provide recommendation on the best strategy to improve the appointment system so that can maximize the effectiveness and efficiency of resource and capacity, The research made use of interview method (Data Collection) to see the appointment system and the factor that effect patient waiting time, Observations was also employed for field visit to public hospital and to see directly outpatient service and arrival pattern of patient hospital, the collected data was analysed using descriptive analysis. However, the research is a preliminary study that analysed each variable separately. The analysis was performed to confirm that the waiting time target not met the minimum service of standard of hospital. For future, further analysis is still required to design this system such as make a simulation of all variable that affects bottleneck and it supposed to make clinical performance more effective and efficient.

Zhan & Liu (2013) "Design and implementation of a clinic appointment registration system", designed and implemented a desktop-based .NET application for clinic appointment registration with the use of MS Access as the database for keeping medical records. The operational function of the system includes appointment registration, data management (Data addition, deletion and searching) and data backup and recovery. The research is based on the following sections: the introduction into the system, the second section is based on the system requirements analysis where the functional requirement analysis and the technical requirement analysis are specified, the third section was the system design where the system function modular design, and database design, the last section of the research work is the System implementation. Because of the inadequate teaching equipment, the functionality of online credit card payment was not implemented.

Peter et al. (2014) the research work proposed a dependable online appointment booking system for NHIS outpatient in Nigerian teaching hospitals. The research proposes a web based medical appointment booking. The scope of the research work is to design a web-based appointment booking system where patients can register, login to the system, book an appointment with a doctor and view appointments. The system also includes the doctor logging

into the system, cancel an appointment, generate appointment and view appointments. However, the research was limited in that the display of bio-data an X-rays and laboratory results were not included in the system due to technical constrains. And more so, the system was not able to diagnose of prescribe drug for usage.

Choudhari et al (2014) proposes an Android Application for Doctor's Appointment. The proposed system consists of two main panel which include the patient, the patient can register with the system, login into the system. After logging into the system, the patient can see list of available doctors and click on any available doctor to view profile of the doctor and access the doctor schedule, and can send a request for an appointment. The doctor on the other hand will be able to view request from patients and can respond to patient request by either accepting the request or rejecting the request. The system will then notify the patient as to the response from the doctor and get notification 2 hours before the actual appointment which will be very useful in case the patient tends to forget the appointment. However, the research does not integrate medical consultation.

Kyambille and Khamisi (2015) researched Enhancing Patient Appointments Scheduling that Uses Mobile Technology. The research presents a mobile based application scheduling system for managing patient appointments in hospital by allowing patients to register for appointments through mobile phones at their own time wherever they are, and make an appointment on their desired slot of time. The design and development of the mobile appointment scheduling system was done using MYSQL with WAMP server and PHP. The database system is developed with MySQL and the scripting language was done utilizing PHP. However, the limitation of this research work is for the system to be able to direct appointment requests to another hospital where doctors with similar expertise are working.

Jain et al (2016) proposed Android Application of patient Appointment System, the system is a mobile based for medical appointment. The focus and scope of this research work is to provide communication between the patient and the doctor, whereby a patient can schedule appointment with the doctor as per the doctors' availability, patient can also interact with doctor through a message system. Modules in this research work include the patient registering, login, search for available doctor, request for appointment. The server generates a QR code for the patient and accept the appointment, the doctor also login, and scan the QR code generated for the patient and also view patient details. The research work does not take into consideration the online medical consultation.

Mahalakshmi, et al (2016) proposed an Online Appointment Reservation and Scheduling for Healthcare, this research work show the Different types of Appointment Scheduling, also Show the study between the traditional appointment reservation and scheduling system, also the software architecture for online appointment reservation and scheduling system were enlisted which include; Features of online appointment reservation and scheduling (Schedule appointment, Reschedule appointment, Check doctors availability, send reminder message, view patient information cancel appointment), practical flow of an online appointment structure, functionality of the appointment system. However, the research work was not implemented on any platform.

Doctor patient interaction system for Android by Bhuvaneswari (2017) present a mobile based application that allows patient to register to use the system, login to the system, book an appointment with a doctor, and can also make complaints on the system, also the system brings the doctor into the platform whereby they can also login into the system, view complaints and as well pot solutions to complaints, also view the clinic register. The research work is categorized into sections which include: the introduction into the research work, the second section is the system analysis which include the existing system, the proposed system and also the system requirements of the proposed system where the hardware and the software specification are mentioned. In the third section, the UML model is being shown showing the use case of each entity concerned in the system, also showing the Data Flow Diagram, Sequence diagram. While the last section includes the System development and implementation.

Android Application for Healthcare Appointment booking System by Chaudhari et al (2017). The system is a mobile based application which was implemented on android operating system, the system proposes two main panels which include the Doctor and the patient, the scope of the research is for the patient to request an appointment with a doctor after seeing various doctor specialization, the doctors' profile and view the doctors schedule so as to know when to fix appointment with the particular doctor. The doctor can either accept or reject the appointment. The proposed system is Doctor's Appointment Booking System for Nagpur City only, the research can enhance the system by expanding the application and including more cities in the application.

CHAPTER 3

SYSTEM DESIGN

3.1 WIDGET TREE

Widget tree is a structure that represents how our widgets are organized in our application.

3.1.1 On boarding screen

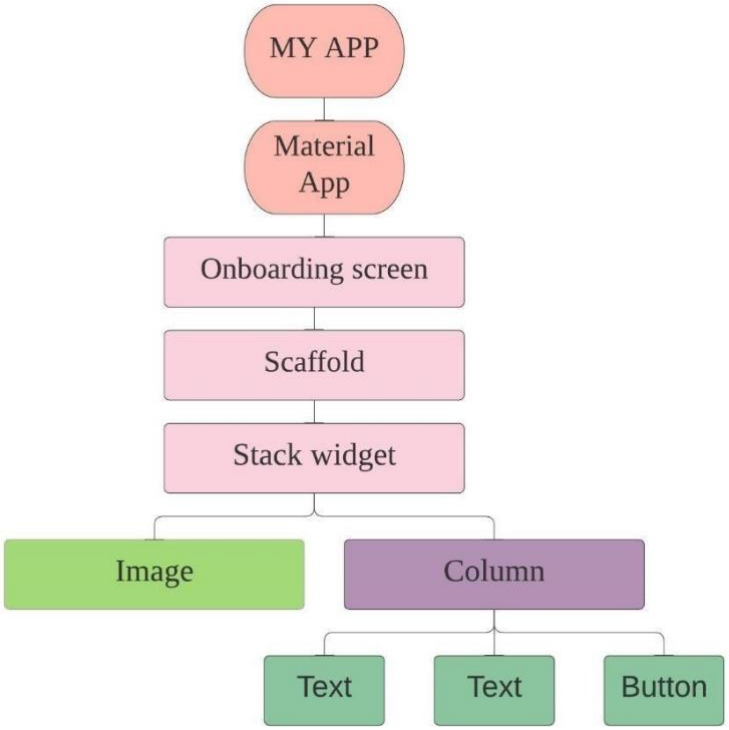

Fig. 3.1.1

The above fig. 3.1.1 represents the widget tree for onboarding screens, all the widgets and components can be accessed using the predefined class called as MaterialApp**.**

3.1.2 Home screen

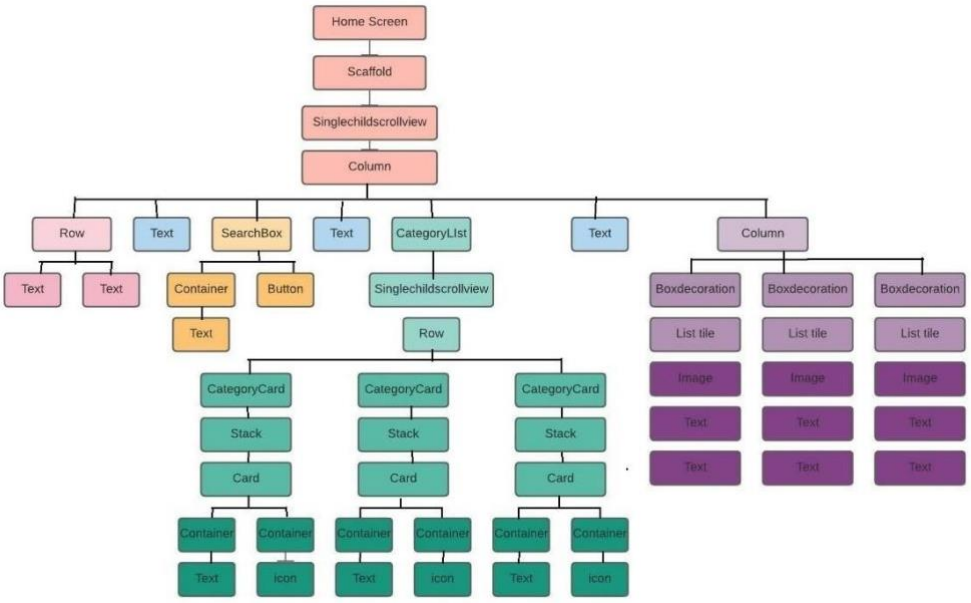

Fig. 3.1.2

The above fig.3.1.2 is the screen that represents the widget tree for home screen which contains row, container (containing row, column, text, icon, row widgets), text, column widgets are defined inside a column widget.

3.1.3 Detail screen

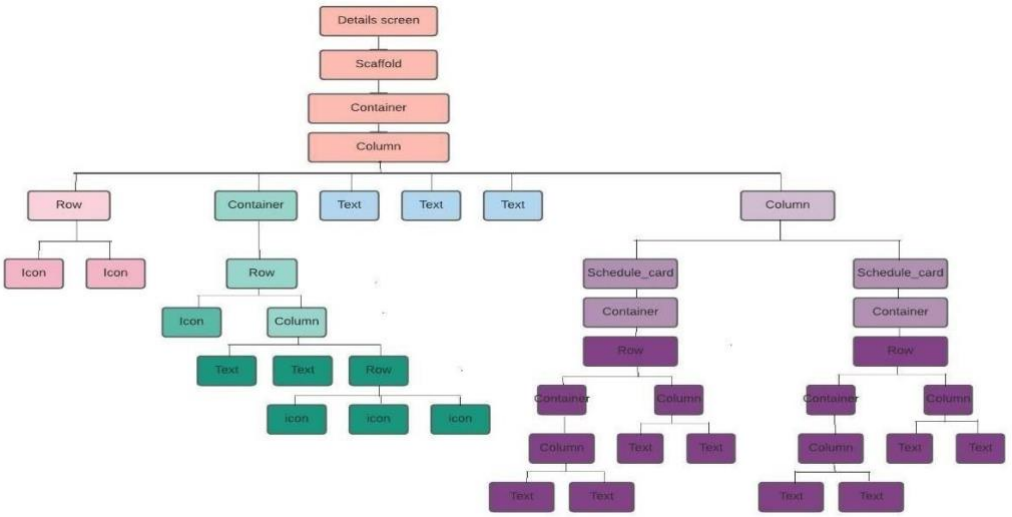

Fig.3.1.3

The above fig. 3.1.3 screen represents the widget tree for detail screen which contains row, container (containing row, column, text, icon, row widgets), text, column widgets are defined inside a column widget.

CHAPTER 4

ANALYSIS

4.1 Hardware and Software Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor: Pentium 4 Processor
Processor Speed: 2.4 GHz
RAM: 2 GB
Storage Space: 40 GB

The software requirements are very minimal and the program can be run on the machines with these requirements satisfied:

Editor: Visual Studio Code
Operating System: Windows/Mac OS
IDE: VS Code

4.2 Tools/ Languages/ Platform

Various tool used in making this project is given below:
Editor/IDE: Visual Studio Code
Operating System: Windows/Mac OS
Languages: Dart

4.3 Functional Requirements

Flutter

Flutter is Google's Mobile SDK to build native iOS and Android apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

Compared to its contemporary technologies like React Native, Kotlin, and Java, Flutter is much better in regard to having a Single Codebase for Android and iOS, Reusable UI and Business Logic, high compatibility, performance, and productivity.

Dart

Dart is an open-source general-purpose programming language developed by Google. It supports application development in both client and server-side. But it is widely used for the development of android apps, iOS apps, IoT(Internet of Things), and web applications using the Flutter Framework.

Syntactically, Dart bears a strong resemblance to Java, C, and JavaScript. It is a dynamic object-oriented language with closure and lexical scope. The Dart language was released in 2011 but came into popularity after 2015 with Dart 2.0.

CHAPTER 5

IMPLEMENTATION

5.1 Code Snippets

5.1.1 main.dart

```
import 'package:doctor_consultation_app/screens/onboarding_screen.dart';
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        textTheme:
            GoogleFonts.varelaRoundTextTheme(Theme.of(context).textTheme),
      ),
      home: OnboardingScreen(),
    );
  }
}
```

Fig. 5.1 code snippet for main. dart

The above fig.5.1 represents main. dart which is the entry point to the program

5.1.2 onboarding screen

```
import 'package:doctor_consultation_app/constant.dart';
import 'package:doctor_consultation_app/screens/home_screen.dart';
import 'package:flutter/material.dart';

class OnboardingScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: kBackgroundColor,
      body: SafeArea(
        bottom: false,
```

```
        child: Stack(
          children: <Widget>[
            Align(
              alignment: Alignment.bottomCenter,
              child: Image.asset(
                'assets/images/onboarding_illustration.png',
                width: MediaQuery.of(context).size.width,
                fit: BoxFit.fitWidth,
              ),
            ),
            Positioned(
              top: MediaQuery.of(context).size.height / 6,
              child: Padding(
                padding: EdgeInsets.symmetric(
                  horizontal: MediaQuery.of(context).size.width / 8,
                ),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: <Widget>[
                    Text(
                      'Welcome',
                      style: TextStyle(
                        fontWeight: FontWeight.bold,
                        fontSize: 32,
                        color: kTitleTextColor,
                      ),
                    ),
                    SizedBox(
                      height: 20,
                    ),
                    Text(
                      'Choose the doctor you want!',
                      style: TextStyle(
                        fontSize: 20,
                        color: kTitleTextColor.withOpacity(0.7),
                      ),
                    ),
                    SizedBox(
                      height: 20,
                    ),
                    MaterialButton(
                      onPressed: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => HomeScreen(),
                          ),
                        );
                      },
```

```
                    color: kOrangeColor,
                    padding: EdgeInsets.symmetric(
                      horizontal: 30,
                    ),
                    shape: RoundedRectangleBorder(
                      borderRadius: BorderRadius.circular(20),
                    ),
                    child: Text(
                      'Get Started',
                      style: TextStyle(
                        color: kWhiteColor,
                        fontSize: 16,
                      ),
                    ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
```

Fig. 5.2 code snippet for onboarding screen

The above fig.5.2 represents onboarding screen code snippet which is the entry point to the program. All the widgets and components can be accessed using the predefined class called as MaterialApp

5.1.3 Home screen

```
import 'package:doctor_consultation_app/components/category_card.dart';
import 'package:doctor_consultation_app/components/doctor_card.dart';
import 'package:doctor_consultation_app/components/search_bar.dart';
import 'package:doctor_consultation_app/constant.dart';
import 'package:flutter/material.dart';
import 'package:flutter_svg/svg.dart';

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: kBackgroundColor,
      body: SafeArea(
```

```
        bottom: false,
        child: SingleChildScrollView(
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: <Widget>[
              Padding(
                padding: EdgeInsets.symmetric(horizontal: 30),
                child: Row(
                  mainAxisAlignment: MainAxisAlignment.spaceBetween,
                  children: <Widget>[
                    SvgPicture.asset('assets/icons/menu.svg'),
                    SvgPicture.asset('assets/icons/profile.svg'),
                  ],
                ),
              ),
              SizedBox(
                height: 50,
              ),
              Padding(
                padding: EdgeInsets.symmetric(horizontal: 30),
                child: Text(
                  'Find Your Desired\nDoctor',
                  style: TextStyle(
                    fontWeight: FontWeight.bold,
                    fontSize: 32,
                    color: kTitleTextColor,
                  ),
                ),
              ),
              SizedBox(
                height: 30,
              ),
              Padding(
                padding: EdgeInsets.symmetric(horizontal: 30),
                child: SearchBar(),
              ),
              SizedBox(
                height: 20,
              ),
              Padding(
                padding: EdgeInsets.symmetric(horizontal: 30),
                child: Text(
                  'Categories',
                  style: TextStyle(
                    fontWeight: FontWeight.bold,
                    color: kTitleTextColor,
                    fontSize: 18,
                  ),
                ),
```

```
            ),
            SizedBox(
              height: 20,
            ),
            buildCategoryList(),
            SizedBox(
              height: 20,
            ),
            Padding(
              padding: EdgeInsets.symmetric(horizontal: 30),
              child: Text(
                'Top Doctors',
                style: TextStyle(
                  fontWeight: FontWeight.bold,
                  color: kTitleTextColor,
                  fontSize: 18,
                ),
              ),
            ),
            SizedBox(
              height: 20,
            ),
            buildDoctorList(),
          ],
        ),
      ),
    ),
  );
}

buildCategoryList() {
  return SingleChildScrollView(
    scrollDirection: Axis.horizontal,
    child: Row(
      children: <Widget>[
        SizedBox(
          width: 30,
        ),
        CategoryCard(
          'Dental\nSurgeon',
          'assets/icons/dental_surgeon.png',
          kBlueColor,
        ),
        SizedBox(
          width: 10,
        ),
        CategoryCard(
          'Heart\nSurgeon',
          'assets/icons/heart_surgeon.png',
```

```
            kYellowColor,
          ),
          SizedBox(
            width: 10,
          ),
          CategoryCard(
            'Eye\nSpecialist',
            'assets/icons/eye_specialist.png',
            kOrangeColor,
          ),
          SizedBox(
            width: 30,
          ),
        ],
      ),
    );
  }

  buildDoctorList() {
    return Padding(
      padding: EdgeInsets.symmetric(
        horizontal: 30,
      ),
      child: Column(
        children: <Widget>[
          DoctorCard(
            'Dr. Stella Kane',
            'Heart Surgeon - Flower Hospitals',
            'assets/images/doctor1.png',
            kBlueColor,
          ),
          SizedBox(
            height: 20,
          ),
          DoctorCard(
            'Dr. Joseph Cart',
            'Dental Surgeon - Flower Hospitals',
            'assets/images/doctor2.png',
            kYellowColor,
          ),
          SizedBox(
            height: 20,
          ),
          DoctorCard(
            'Dr. Stephanie',
            'Eye Specialist - Flower Hospitals',
            'assets/images/doctor3.png',
            kOrangeColor,
          ),
```

```
          SizedBox(
            height: 20,
          ),
        ],
      ),
    );
  }
}
```

Fig. 5.3 code snippet for home screen

The above fig.5.3 is the home screen code snippet which contains row, container (containing row, column, text, icon, row widgets), text, column widgets are defined inside a column widget, it also contains category_card and doctor_card which are user defined name of the card widgets to which different widgets are passed.

5.1.4 Detail screen

```
import 'package:doctor_consultation_app/components/schedule_card.dart';

import 'package:doctor_consultation_app/constant.dart';
import 'package:flutter/material.dart';
import 'package:flutter_svg/svg.dart';

class DetailScreen extends StatelessWidget {
  var _name;
  var _description;
  var _imageUrl;

  DetailScreen(this._name, this._description, this._imageUrl);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Container(
          width: double.infinity,
          decoration: BoxDecoration(
            image: DecorationImage(
              image: AssetImage('assets/images/detail_illustration.png'),
              alignment: Alignment.topCenter,
              fit: BoxFit.fitWidth,
            ),
          ),
          child: Column(
            children: <Widget>[
              SizedBox(
                height: 50,
```

```
            ),
            Padding(
              padding: EdgeInsets.symmetric(
                horizontal: 30,
              ),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: <Widget>[
                  InkWell(
                    onTap: () {
                      Navigator.pop(context);
                    },
                    child: SvgPicture.asset(
                      'assets/icons/back.svg',
                      height: 18,
                    ),
                  ),
                  SvgPicture.asset(
                    'assets/icons/3dots.svg',
                    height: 18,
                  ),
                ],
              ),
            ),
            SizedBox(
              height: MediaQuery.of(context).size.height * 0.24,
            ),
            Container(
              width: double.infinity,
              decoration: BoxDecoration(
                color: kBackgroundColor,
                borderRadius: BorderRadius.vertical(
                  top: Radius.circular(50),
                ),
              ),
              child: Padding(
                padding: EdgeInsets.all(30),
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: <Widget>[
                    Row(
                      children: <Widget>[
                        Image.asset(
                          _imageUrl,
                          height: 120,
                        ),
                        SizedBox(
                          width: 20,
                        ),
```

```
                        Column(
                          crossAxisAlignment: CrossAxisAlignment.start,
                          children: <Widget>[
                            Text(
                              _name,
                              style: TextStyle(
                                fontWeight: FontWeight.bold,
                                fontSize: 20,
                                color: kTitleTextColor,
                              ),
                            ),
                            SizedBox(
                              height: 10,
                            ),
                            Text(
                              _description,
                              style: TextStyle(
                                color: kTitleTextColor.withOpacity(0.7),
                              ),
                            ),
                            SizedBox(
                              height: 10,
                            ),
                            Row(
                              children: <Widget>[
                                Container(
                                  padding: EdgeInsets.all(10),
                                  decoration: BoxDecoration(
                                    color: kBlueColor.withOpacity(0.1),
                                    borderRadius:
BorderRadius.circular(10),
                                  ),
                                  child: SvgPicture.asset(
                                    'assets/icons/phone.svg',
                                  ),
                                ),
                                SizedBox(
                                  width: 16,
                                ),
                                Container(
                                  padding: EdgeInsets.all(10),
                                  decoration: BoxDecoration(
                                    color: kYellowColor.withOpacity(0.1),
                                    borderRadius:
BorderRadius.circular(10),
                                  ),
                                  child: SvgPicture.asset(
                                    'assets/icons/chat.svg',
                                  ),
```

```
                                ),
                                SizedBox(
                                  width: 16,
                                ),
                                Container(
                                  padding: EdgeInsets.all(10),
                                  decoration: BoxDecoration(
                                    color: kOrangeColor.withOpacity(0.1),
                                    borderRadius:
BorderRadius.circular(10),
                                  ),
                                  child: SvgPicture.asset(
                                    'assets/icons/video.svg',
                                  ),
                                ),
                              ],
                            ),
                          ],
                        )
                      ],
                    ),
                    SizedBox(
                      height: 50,
                    ),
                    Text(
                      'About Doctor',
                      style: TextStyle(
                        fontWeight: FontWeight.bold,
                        fontSize: 18,
                        color: kTitleTextColor,
                      ),
                    ),
                    SizedBox(
                      height: 10,
                    ),
                    Text(
                      'Dr. Stella is the top most heart surgeon in
Flower\nHospital. She has done over 100 successful sugeries\nwithin past 3
years. She has achieved several\nawards for her wonderful contribution in
her own\nfield. She’s available for private consultation for\ngiven
schedules.',
                      style: TextStyle(
                        height: 1.6,
                        color: kTitleTextColor.withOpacity(0.7),
                      ),
                    ),
                    SizedBox(
                      height: 20,
                    ),
```

```
                    Text(
                      'Upcoming Schedules',
                      style: TextStyle(
                        fontWeight: FontWeight.bold,
                        fontSize: 18,
                        color: kTitleTextColor,
                      ),
                    ),
                    SizedBox(
                      height: 20,
                    ),
                    ScheduleCard(
                      'Consultation',
                      'Sunday . 9am - 11am',
                      '12',
                      'Jan',
                      kBlueColor,
                    ),
                    SizedBox(
                      height: 10,
                    ),
                    ScheduleCard(
                      'Consultation',
                      'Sunday . 9am - 11am',
                      '13',
                      'Jan',
                      kYellowColor,
                    ),
                    SizedBox(
                      height: 10,
                    ),
                    ScheduleCard(
                      'Consultation',
                      'Sunday . 9am - 11am',
                      '14',
                      'Jan',
                      kOrangeColor,
                    ),
                    SizedBox(
                      height: 20,
                    ),
                  ],
                ),
              ),
            )
          ],
        ),
      ),
    ),
```

```
    );
  }
}
```

Fig. 5.4 code snippet for detail screen

The above fig.5.3 is the code snippet for Detail screen which contains row, container (containing row, column, text, icon, row widgets), text, column widgets are defined inside a column widget, it also contains schedule_card which are user defined name of the card widgets to which different widgets are passed.

CHAPTER 6

TESTING

6.1 Introduction

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

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

6.2 Levels of Testing

6.2.1 Unit Testing

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

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

6.2.2 Integration Testing

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

6.2.3 System Testing

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

6.2.4 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)

6.2.5 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.

6.2.6 User Acceptance Testing

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

CHAPTER 7

RESULTS

7.1 On boarding screen

Fig.7.1 On boarding screen

The above fig.7.1 represents the onboarding screen containing the text and button which on clicked directs the user to the home screen.

7.2 Home screen

Fig 7.2 Home screen

The above fig.7.2 represents the home screen containing the texts, search bar, category list and the list of the doctors.

7.3 Detail screen

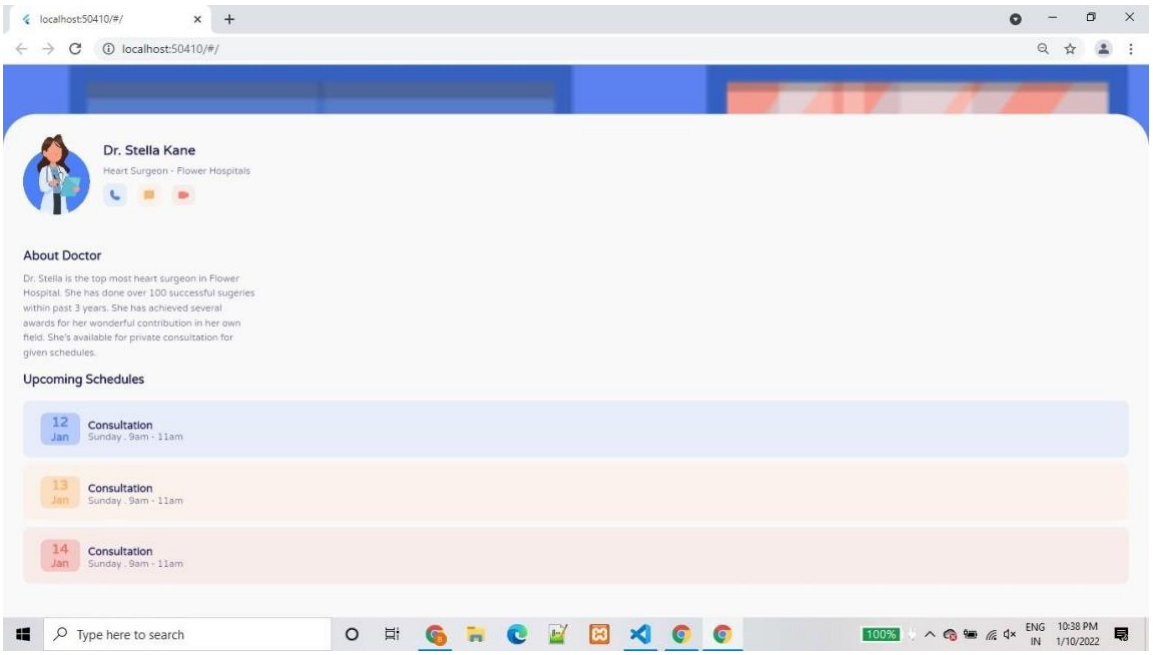

Fig.7.3 Detail screen

The fig.7.3 represents the detail screen containing the information regarding the doctor, icons to call, message and video call them along with their schedule.

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENTS

8.1 Conclusion

After successful completion of the project, the work has been able to achieve the main aim of the work which is the implementation of medical appointment and consultation system. The application has been able to achieve scheduling of patients with number of available doctors. The system will help reduce the stress or fatigue patients go through waiting on the queue to meet a doctor and the application helps in bringing the doctors and patients closer anytime and anywhere with the successful integration of the live consultation module. This single application which caters all the details of services along with detailed description that are offered to the end users by the company and is user friendly with attractive UI thus easily used by all age groups**.**

8.2 Future Enhancements

A separate set of screens can be developed for user so that they can easily keep track of their previous visits to the doctor. Allowing the transparency to distance or places wherever the user/patient is located, using the software connecting to a hospital operated server and using communications GPS and GSM system. Including the integrated database services to store the patient, doctors and schedule records.

REFERENCES

[1] https://flutter.dev/

[2] https://developers.google.com/learn/pathways/intro-to-flutter

[3] Beginning Flutter: A Hands on Guide to App Development by Marco L Napoli

[4] https://flutterawesome.com/

[5]https://stackoverflow.com/

[6] https://www.geeksforgeeks.org/