# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

### BELAGAVI – 590 014



#### Internship Project Report On

**“*FOOD DELIVERY APP* ”**

By:

### ENMAZ

*Submitted in partial fulfillment of the requirements for the 7th semester of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by:

#### Varun Kumar Shivam Paliwal

#### 1RN18IS121 1RN18IS099

Under the Guidance of:

**Dr. R Rajkumar**

**Associate Professor**

 **Dept. of ISE**

**Department of Information Science and Engineering**

**RNS Institute of Technology**

**Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098**

**2021-2022**

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

### DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING



**CERTIFICATE**

Certified that the Internship project work entitled **“Food delivery app”** has been successfully carried out by **Varun Kumar** bearing USN **1RN18IS121** and **Shival Paliwal** bearingUSN **1RN18IS099** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the **7th semester Bachelor of Engineering** in **Information Science and Engineering** of **Visvesvaraya Technological University**, Belagavi, during the academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the internship project of 7th semester BE in ISE.

|  |  |  |
| --- | --- | --- |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Mr. R Rajkumar** | **Dr. Suresh L** | **Dr. M.K. Venkatesha** |
| Internship Guide | Professor and HoD | Principal |
| Associate Professor | Department of ISE | RNSIT |
| Department of ISE | RNSIT |  |
|  |  |  |
|  | **External Viva** |  |
| **Name of the Examiners** |  | **Signature with Date** |
| **1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | | **2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |

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, we would like to thank the **Management of RNS Institute of** **Technology** for providing such a healthy environment for the successful completion ofinternship work.

In this regard, we 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 **Mr. Ravi Kumar** 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 always helping.

We thank **Mr. Akshay D R, Enmaz**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

We 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: Varun Kumar 1RN18IS121

Place: Bengaluru Shivam Paliwal 1RN18IS099

#### i

**ABSTRACT**

Food delivery apps are the new favorite of the people and that is the final verdict.

With the option to order their meals from a mobile app, they can easily place an order when they are stuck in traffic or on the way to pick up the kids. Mobile apps provide the freedom to order from any place at any time without pausing everything and making a call to the restaurant.

An Online Food Ordering System is proposed here which simplifies the food ordering process. The proposed system shows an user interface and update the menu with all available options so that it eases the customer work. Customer can choose more than one item to make an order and can view order details before logging off. The order confirmation is sent to the customer. The order is placed in the queue and updated in the database and returned in real time. This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

|  |  |
| --- | --- |
| **CONTENTS**  **Description Acknowledgement Abstract**  **Content**  **List of Figures**  **1. Introduction** | **Page No.**  i ii iii iv |
| * 1. Organization      1. Company Profile | 1  1 |
| * 1. Domain Technology      1. Introduction to Flutter      2. History      3. FrameWork-Architecture | 2  2  3 |
| **2. Literature Survey** |  |
| 2.1 Food delivery app using Flutter | 5 |
| * 1. Functional Requirements   2. Tools/ Languages/ Platform | 6  9 |
| **3. Design and Implementation Definition** |  |
| * 1. Architecture/ Sequence Diagram/ Flowchart   2. Algorithm/Methods/ Pseudo code | 10  11 |
| 1. **Observation And Results**    1. Testing    2. Code    3. Results & Snapshots 2. **Conclusion And Future Enhancement References** | 14  14  20  25  26 |

1

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **Fig. No.** | **Description** | **Page No** |
| Fig. 3.1 | Three Layer Architecture | 10 |
| Fig. 4.1 | Test Case | 14 |
| Fig. 4.2 | Landing Page | 20 |
| Fig. 4.3 | Home Page | 21 |
| Fig. 4.4 | About Page | 22 |
| Fig. 4.5 | Login Page | 22 |
| Fig. 4.6 | Post Page | 23 |

**CHAPTER 1**

**INTRODUCTION**

In today’s world, Smart phones have changed our lives and have become an indispensable part of our lives because of its specialty to simplify our routine work and thereby saving our time.

Currently most of the Examination like organizational recruitment, University class test are paper based, which costs time and resources. Questionnaire is developed, printed, and then collect data, entry, editing, cleaning, which time consuming and costly.

Proposed application is the starting for avoid those circumstances which are been currently faced by any organization.

## ORGANIZATION

### Company Profile

**Enmaz** has a simple yet robust solution that helps any Industry / Factory digitise their workfloor in no time. The products offered will help in remote monitoring, controlling and also analysing any machinene parameter or process.

* 1. **Domain Technology**

**1.2.1 Introduction to Flutter**

[Flutter](https://www.geeksforgeeks.org/flutter-an-introduction-to-the-open-source-sdk-by-google/) 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.

### 1.2.2 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](https://en.wikipedia.org/wiki/Android_(operating_system)) operating system. It was unveiled at the 2015 [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)) developer summit[[6]](https://en.wikipedia.org/wiki/Flutter_(software)#cite_note-6) with the stated intent of being able to [render](https://en.wikipedia.org/wiki/Rendering_(computer_graphics)) consistently at 120 [frames per second](https://en.wikipedia.org/wiki/Frame_rate).[[7]](https://en.wikipedia.org/wiki/Flutter_(software)#cite_note-7) During the keynote of [Google Developer Days](https://en.wikipedia.org/wiki/Google_Developer_Day) 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.[[8]](https://en.wikipedia.org/wiki/Flutter_(software)#cite_note-8)

On May 6, 2020, the Dart software development kit ([SDK](https://en.wikipedia.org/wiki/Software_development_kit)) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the [Metal](https://en.wikipedia.org/wiki/Metal_(API)) 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 CanvasKit renderer and web specific widgets, early-access desktop application support for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), and [Linux](https://en.wikipedia.org/wiki/Linux) and improved Add-to-App [APIs](https://en.wikipedia.org/wiki/API).[[9]](https://en.wikipedia.org/wiki/Flutter_(software)#cite_note-9) This release included sound [null-safety](https://en.wikipedia.org/wiki/Void_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](https://en.wikipedia.org/wiki/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.

### 1.2.3 Framework-Architecture

The major components of Flutter include:

* [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)) platform
* Flutter engine
* Foundation library
* Design-specific widgets
* Flutter Development Tools (DevTools)

**Dart platform**

Flutter apps are written in the [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)) language and make use of many of the language's more advanced features.

On [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [macOS](https://en.wikipedia.org/wiki/MacOS), and [Linux](https://en.wikipedia.org/wiki/Linux)[[11]](https://en.wikipedia.org/wiki/Flutter_(software)#cite_note-11) Flutter runs in the Dart virtual machine, which features a [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) 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](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/State_(computer_science)).

For better performance, release versions of Flutter apps targeting Android and iOS are compiled with [ahead-of-time (AOT) compilation](https://en.wikipedia.org/wiki/Ahead-of-time_compilation).

**Flutter engine**

Flutter's engine, written primarily in [C++](https://en.wikipedia.org/wiki/C%2B%2B), provides low-level [rendering](https://en.wikipedia.org/wiki/Rendering_(computer_graphics)) support using Google's [Skia](https://en.wikipedia.org/wiki/Skia_Graphics_Engine) graphics library. Additionally, it interfaces with [platform-specific](https://en.wikipedia.org/wiki/Platform-specific_model) [SDKs](https://en.wikipedia.org/wiki/Software_development_kit) such as those provided by [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS).[[10]](https://en.wikipedia.org/wiki/Flutter_(software)#cite_note-:0-10) 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](https://en.wikipedia.org/wiki/Dart_(programming_language)), provides basic classes and functions that are used to construct applications using Flutter, such as [APIs](https://en.wikipedia.org/wiki/Application_programming_interface) to communicate with the engine.

**Design-specific widgets**

The Flutter framework contains two sets of [widgets](https://en.wikipedia.org/wiki/Flutter_(software)#Widgets) that conform to specific design languages: [Material Design](https://en.wikipedia.org/wiki/Material_Design) widgets implement Google's [design language](https://en.wikipedia.org/wiki/Design_language) of the same name, and Cupertino widgets implement Apple's [iOS](https://en.wikipedia.org/wiki/IOS) [Human interface guidelines](https://en.wikipedia.org/wiki/Human_interface_guidelines)

**Chapter 2**

## LITERATURE SURVEY

### 2.1 Food delivery app using Flutter

An Online Food Ordering System is proposed here which simplifies the food ordering process. The proposed system shows an user interface and update the menu with all available options so that it eases the customer work. Customer can choose more than one item to make an order and can view order details before logging off. The order confirmation is sent to the customer. The order is placed in the queue and updated in the database and returned in real time. This system assists the staff to go through the orders in real time and process it efficiently with minimal errors.

**2.1.1 Introduction**

The labour rates are increasing steadily year on year thus making it difficult to find employees. The food industry is highly labour intensive and the biggest expense in the food industry is the cost of employing the right kind of people to do the work. One of the ways to reduce this expense is to use modern technology to replace some of the jobs done by human beings and make machines do the work. Here we propose an “Online Food Ordering System” that has been designed for Fast Food restaurant, Take-Out or College Cafeterias. The system can also be used in any food delivery industry. This simplifies the process of food ordering for both the customer and the restaurant, as the entire process of taking orders is automated. Android is very popular nowadays among students and students are now choosing Android for his or her projects. It’s greatly important for a beginner to create baby Android apps to learn Android. Android Mobile hardware and software platforms allow the running of faster and richer applications. This paper presents the most steps within the development of a quiz application for Android using Android Studio. Android could be a software package and Linux based operating system for mobile devices/ equipments such as tablet computers and smartphones. It's developed by Google and later the OHA (Open Handset Alliance). Java language is principally wont to write the android code while other languages are used. The goal of the android project is to form a successful real-world product that improves the mobile experience for end-users.

**2.1.2 Literature Survey**

Android is rapidly getting famous, and therefore the number of its users are increasing day by day, because it's easy to access the required Android-based apps on tablets and smartphones . Therefore, we found the use of the Android App is less complicated and longer efficient to facilitate the users during this way with none difficulty. Fig 1 Applications and widgets Application Programming Interface Libraries Android runtime Android Architecture reduces the time of development and can reduce the quantity of memory that is required by the appliance. In Future we are able to include redesigned an friendlier user interface because the application targets various levels of users starting from sophisticated programmer to naive users Android is most used Mobile OS worldwide. You simply must have minor knowledge about android structure and tiny java programming. Android is an open source and Linuxbased package for mobile devices like smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies. Android offers a unified approach to application development for mobile devices which suggests developers need only develop for Android, and their applications should be ready to run on different devices powered by Android. The very first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 while as because the first working commercial version, Android 1.0, was released in September 2008, and later it kept on changing/updating its rules. On June 27, 2012, at the Google I/O conference, in that meeting, Google announced the subsequent Android version, 4.1. Jelly Bean is an incremental update, with the first aim of improving the program, both in terms of functionality/ activities and performance. The text file i.e source code for Android is out there under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and therefore the rest, Linux kernel changes, under the GNU General Public License version.

## CHAPTER 3

**REQUIREMENT ANALYSIS, TOOLS & TECHNOLOGIES**

* 1. **Hardware & 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 | : | 4 GB |
| Storage Space | : | 40 GB |
| Monitor Resolution | : | 1024\*768 or 1336\*768 or 1280\*1024 |

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

|  |  |  |
| --- | --- | --- |
| Editor | : | Visual Studio Code |
| Operating System | : | Windows Operating System |
| IDE | : | VS Code |
| Backend Tool | : | MongoDB |

**3.2 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,

Backend Tool : Open Trivia Api

**3.3 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](https://www.geeksforgeeks.org/kotlin-programming-language/), and [Java](https://www.geeksforgeeks.org/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](https://www.geeksforgeeks.org/flutter-an-introduction-to-the-open-source-sdk-by-google/).

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 4

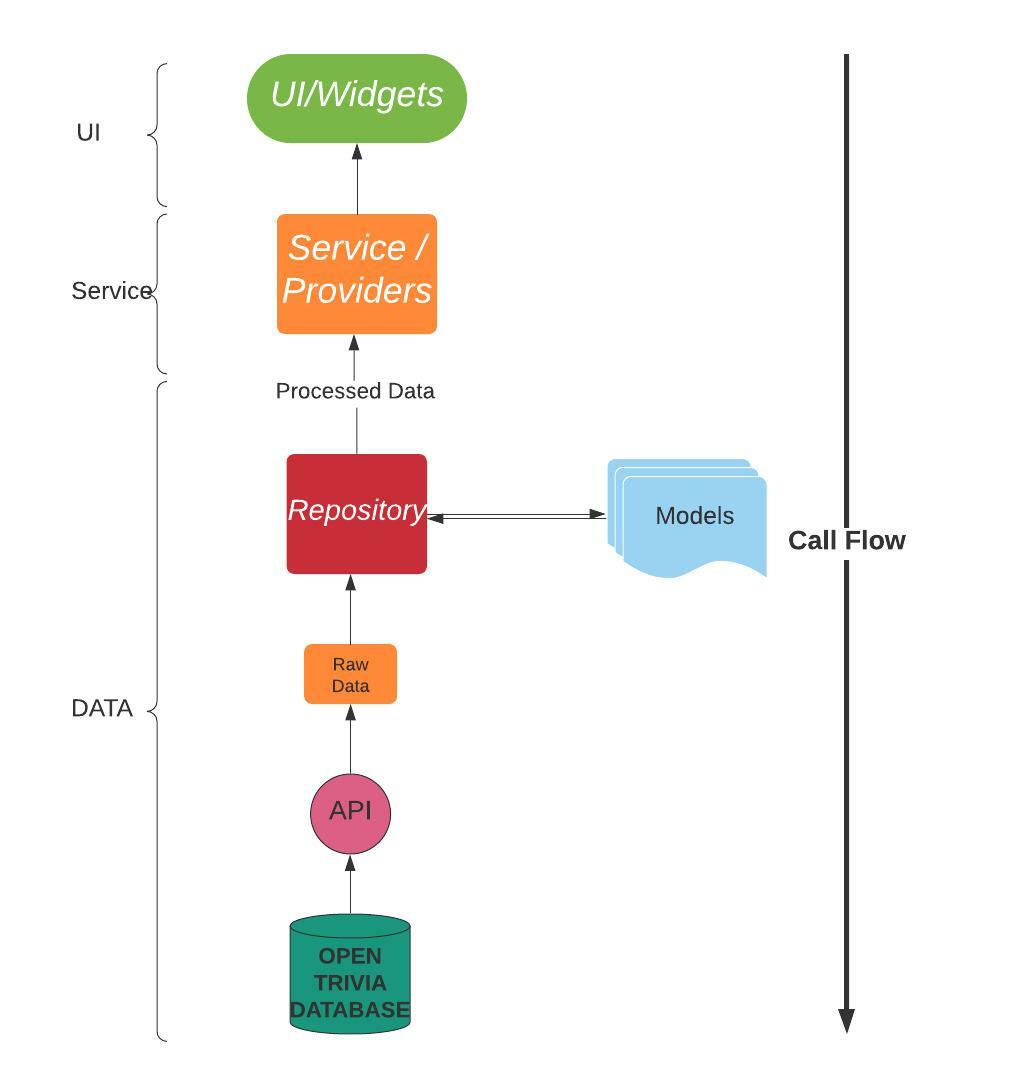
**DESIGN AND IMPLIMENTATION**

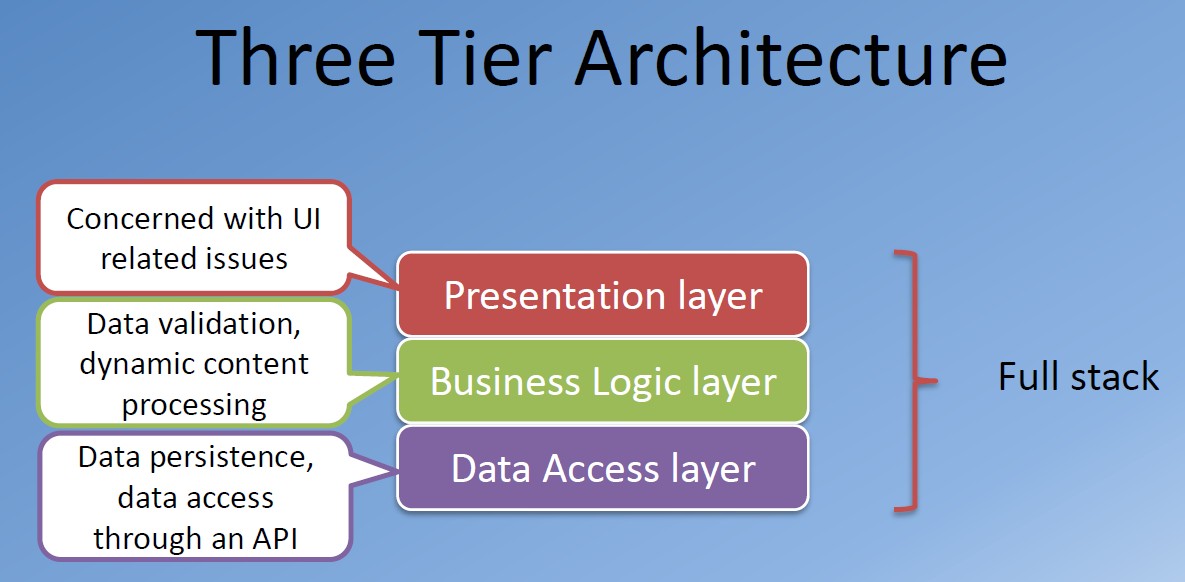
* 1. **Architecture/Sequence diagram/Class diagrams /Flowchart**

Humans face many choices on a daily basis and sentiment analysis can systematize the process of coming to a decision based on past outcomes of that decision. For example, if one has to vote during elections, then rather than manually going through all the opinions and propaganda, a sentiment classifier can predict the overall sentiment of the popular vote and the favoured party during the general election.

**3.2 LIMITATIONS OF EXISTING SYSTEM**

There are few websites like LibraryThings, GoodReads having good interface but the a

****



*Fig. 3.1 Three Layer Architecture*

## Chapter 5

**IMPLEMENTATION DETAILS**

**5.1 main.dart**

import 'package:flutter/material.dart';

import 'package:foodorderapp/Style.dart';

import 'package:foodorderapp/HotelPage.dart';

**void** main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

*// This widget is the root of your application.*

**@override**

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      theme: ThemeData(

        fontFamily: 'mont'

      ),

      home: MyHomePage(),

    );

  }

}

class MyHomePage extends StatefulWidget {

**@override**

  \_MyHomePageState createState() => \_MyHomePageState();

}

**5.2 cartpage.dart**

import 'dart:ffi';

import 'package:flutter/material.dart';

import 'package:foodorderapp/Style.dart';

import 'package:foodorderapp/SuccessPage.dart';

class CartPage extends StatelessWidget {

**@override**

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      home: cartPage(),

    );

  }

}

class cartPage extends StatefulWidget {

**@override**

  \_cartPageState createState() => \_cartPageState();

}

Row placesWidget(String img, String name)

  {

    return Row(

      children: [

        Container(

          height: 100,

          width: 100,

          decoration: BoxDecoration(

              image: DecorationImage(

                  image: AssetImage("asset/images/$img.png")

              )

          ),

        ),

        Expanded(

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              Text("$name", style: TextStyle(

                  fontSize: 16,

                  fontWeight: FontWeight.w600

              ),),

              Row(

                children: [

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                ],

              ),

              Text("Lorem ipsum sits dolar amet is for publishing", style: TextStyle(

                  fontSize: 12

              ),)

            ],

          ),

        ),

        SizedBox(width: 10,),

        Row(

          children: [

            Text("Quantity ", style: TextStyle(

              fontSize: 14,

              color: black

            ),),

            Container(

              padding: EdgeInsets.symmetric(horizontal: 10, vertical: 10),

              decoration: BoxDecoration(

                  border: Border.all(color: black),

              ),

              child: Text("1", style: TextStyle(

                  fontSize: 13,

                  fontWeight: FontWeight.w700

              ),),

            ),

          ],

        )

      ],

    );

  }

**void** openSuccessPage()

  {

    Navigator.push(context, MaterialPageRoute(builder: (context)=>SuccessPage()));

  }

}

**5.3 hotelpage.dart**

import 'package:flutter/material.dart';

import 'package:foodorderapp/CartPage.dart';

import 'package:foodorderapp/Style.dart';

class HotelPage extends StatelessWidget {

**@override**

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      theme: ThemeData(

        fontFamily: "mont"

      ),

      home: hotelPage(),

    );

  }

}

class hotelPage extends StatefulWidget {

**@override**

  \_hotelPageState createState() => \_hotelPageState();

}

Row placesWidget(String img, String name)

  {

    return Row(

      children: [

        Container(

          height: 100,

          width: 100,

          decoration: BoxDecoration(

              image: DecorationImage(

                  image: AssetImage("asset/images/$img.png")

              )

          ),

        ),

        Expanded(

          child: Column(

            crossAxisAlignment: CrossAxisAlignment.start,

            children: [

              Text("$name", style: TextStyle(

                  fontSize: 16,

                  fontWeight: FontWeight.w600

              ),),

              Row(

                children: [

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                  Icon(Icons.star, size: 20, color: Colors.orange,),

                ],

              ),

              Text("Lorem ipsum sits dolar amet is for publishing", style: TextStyle(

                  fontSize: 12

              ),)

            ],

          ),

        ),

        InkWell(

          onTap: (){},

          child: Container(

            padding: EdgeInsets.symmetric(horizontal: 20, vertical: 10),

            decoration: BoxDecoration(

                borderRadius: BorderRadius.all(Radius.circular(40)),

                color: greenBtn

            ),

            child: Text("Order Now", style: TextStyle(

                color: Colors.white,

                fontSize: 12,

                fontWeight: FontWeight.w700

            ),),

          ),

        )

      ],

    );

  }

**void** openCartPage()

  {

    Navigator.push(context, MaterialPageRoute(builder: (context)=>CartPage()));

  }

}

**5.4 successpage.dart**

import 'package:flutter/material.dart';

import 'package:foodorderapp/Style.dart';

class SuccessPage extends StatelessWidget {

**@override**

  Widget build(BuildContext context) {

    return MaterialApp(

      debugShowCheckedModeBanner: false,

      theme: ThemeData(

        fontFamily: "mont"

      ),

      home: successPage(),

    );

  }

}

class successPage extends StatefulWidget {

**@override**

  \_successPageState createState() => \_successPageState();

}

class \_successPageState extends State<successPage> {

**@override**

  Widget build(BuildContext context) {

    return Scaffold(

      body: Container(

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.start,

          children: [

            Container(

              padding: EdgeInsets.all(20),

              decoration: BoxDecoration(

                  color: blue,

                  image: DecorationImage(

                      image: AssetImage("asset/images/hotelBig.png"),

                      fit: BoxFit.cover

                  ),

                  borderRadius: BorderRadius.only(bottomLeft: Radius.circular(40), bottomRight: Radius.circular(40))

              ),

              child: Column(

                crossAxisAlignment: CrossAxisAlignment.start,

                children: [

                  SizedBox(height: 20,),

                  Row(

                    mainAxisAlignment: MainAxisAlignment.spaceBetween,

                    children: [

                      IconButton(

                          icon: Icon(Icons.arrow\_back, color: Colors.white,)

                      ),

                      IconButton(

                          icon: Icon(Icons.search, color: Colors.white,)

                      ),

                    ],

                  ),

                  SizedBox(height: 100,),

                  Row(

                    mainAxisAlignment: MainAxisAlignment.spaceBetween,

                    children: [

                      Column(

                        crossAxisAlignment: CrossAxisAlignment.start,

                        children: [

                          SizedBox(

                            height: 20,

                          ),

                          Text("Yoshimasa Sushi", style: TextStyle(

                              color: Colors.white,

                              fontWeight: FontWeight.w700,

                              fontSize: 20

                          ),),

                          SizedBox(height: 10,),

                          Row(

                            crossAxisAlignment: CrossAxisAlignment.end,

                            children: [

                              Icon(Icons.star, color: Colors.white,),

                              Icon(Icons.star, color: Colors.white,),

                              Icon(Icons.star, color: Colors.white,),

                              Icon(Icons.star, color: Colors.white,),

                              Icon(Icons.star, color: Colors.white,),

                              Text(" 250 Reviews", style: TextStyle(

                                  color: Colors.white,

                                  fontSize: 13

                              ),)

                            ],

                          )

                        ],

                      ),

                      Container(

                        height: 70,

                        width: 70,

                        decoration: BoxDecoration(

                            shape: BoxShape.circle,

                            color: Colors.white

                        ),

                        child: Center(

                          child: Icon(Icons.favorite,color: Colors.redAccent, size: 35,),

                        ),

                      )

                    ],

                  ),

                ],

              ),

            ),

            Expanded(

              child: Column(

                mainAxisAlignment: MainAxisAlignment.center,

                crossAxisAlignment: CrossAxisAlignment.center,

                children: [

                  Icon(Icons.check\_circle,

                    color: greenBtn,

                    size: 200,),

                  SizedBox(height: 20,),

                  Row(

                    mainAxisAlignment: MainAxisAlignment.center,

                    children: [

                      Text("Done!, Tasty food, Your Way!", style: TextStyle(

                        color: greenBtn,

                        fontSize: 20,

                        fontWeight: FontWeight.w600

                      ),)

                    ],

                  )

                ],

              ),

            )

          ],

        ),

      ),

    );

  }

}

**5.5 style.dart**

import 'package:flutter/material.dart';

Color blue = new Color(0xff50c9ff);

Color green = new Color(0xff00e582);

Color black = new Color(0xff4d4d4d);

Color greenBtn = new Color(0xff00d672);

## 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.

A 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 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.

**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.

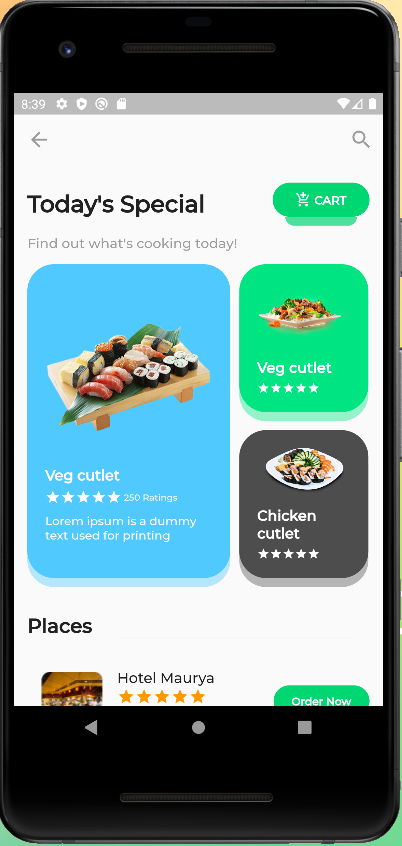
User Acceptance Testing is done in the final phase of testing.

## Chapter 7

**DISCUSSION OF RESULTS**

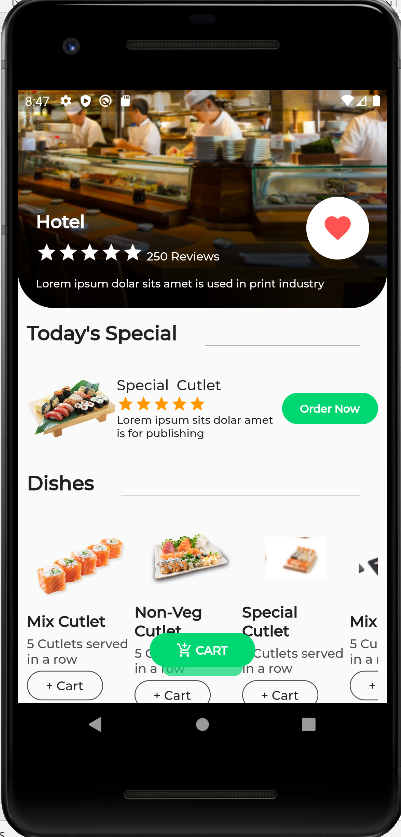
**7.1 Home page**

This is the landing page of the application where we can select all the hotels and top dishes in the application.



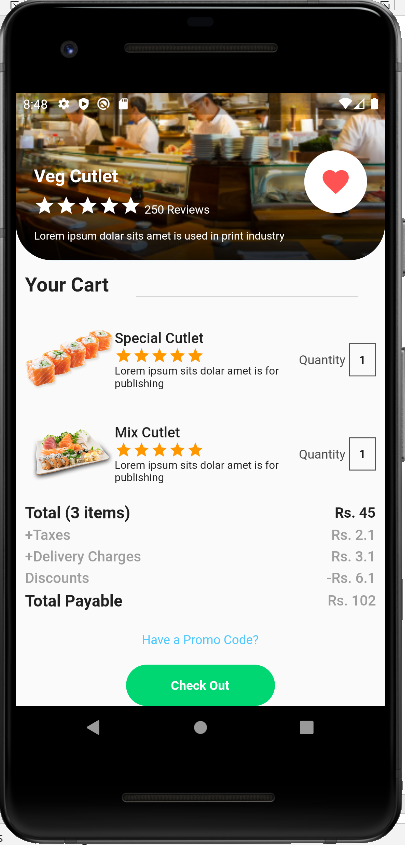
**7.2 Hotel page**

In this page, the items available in the hotel will be listed. An special dish is also included.



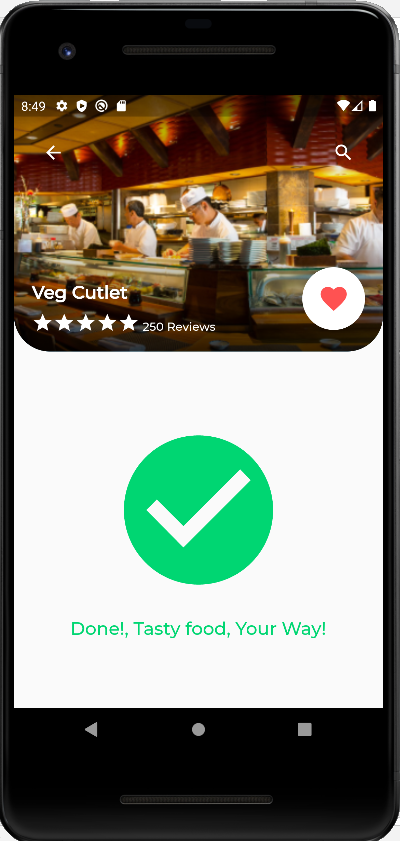
**7.3 Cart page**

It shows the items selected for the order.



**7.4 Final Page**

It shows that order is placed and out for delivery.



## Chapter 8

**CONCLUSION AND FUTURE WORK**

**8.1 Conclusion and Future work**

* Connecting app to internet for real world usage.
* Providing counter and cancel options.
* Online payment gateway.
* Tracking the order.
* Calling the driver.
* Anxiety Bot.
* Having some game for waiting time.
* Getting suggestion according to time and place.
* Adding customer reviews to the food.
* Placing party orders.
* Allowing discount coupons.
* Booking in restaurants for dine out.

## Chapter 9

**REFERENCES**

* <https://flutter.dev/>
* <https://developers.google.com/learn/pathways/intro-to-flutter>
* Beginning Flutter: A Hands On Guide to App Development by Marco L Napoli
* <https://stackoverflow.com/>
* https://www.geeksforgeeks.org