# CONTENTS

ABSTRACT

Chapter Name of the Chapter Page No.

1. **INTRODUCTION**
   1. Problem Statement 1
   2. Objectives 1
   3. Motivation 1
2. **LITERATURE SURVEY**

Existing System 2

Proposal System 2

Summary 2

1. **SOFTWARE REQUIREMENTS S SPECIFICATIONS** 
   1. External Interface Requirements 3

Non-Functional Requirements 4

1. **SYSTEM DESIGN** 
   1. System Architecture 5
   2. Use Case Diagram 6
   3. Entity Relationship Diagram 8
   4. Database Entries 9
   5. Android Application Design 12
   6. Activity Diagram 13
   7. Class Diagram 14
2. **IMPLEMENTATION**
   1. Environmental Setup 15
   2. Module Description 18
   3. Software Description 24

Sample Code 25

1. **SYSTEM TESTING**
   1. Tests 35
2. **RESULTSCREEN SHOTS 36**

1. **CONCLUSION 42**
2. **BIBLIOGRAPHY 43**

# LIST OF FIGURES

Figure 1: System architecture……………………………………………….….5

Figure 2: Use Case Diagram for Developer…………………………………...6

Figure 3: Use Case Diagram for User……………………………………….….7

Figure 4: Entity Relationship Diagram………………………………………..8

Figure 5: Database entries of French cuisine…………………………………9

Figure 6: Database entries of Italian cuisine…………………………….…..10

Figure 7: Database entries of Mexican cuisine………………………….……11

Figure 8: Android Activities…………………………………………………….12

Figure 9: Activity Diagram………………………………………………...……13

Figure 10: Class Diagram………………………………………………..………14

Figure 11: User viewing the recipe……………………………………...………17

Figure 12: Viewing the name of the recipe…………………………………...………17

Figure 13: Ingredients of the above recipe…………………………...………..18

Figure 14: User adding the recipe as his/her favourite………………...……18

Figure 15: User can view his /her favourite recipe……………………...……19

Figure 16:CODE FOR BUILD.GRADEL………………………………………....……………21

Figure 17: CODE FOR Widget\_test.dart……………………………………………….….22

Figure 18: CODE FOR gridle android app…………………………………………….……22

Figure 19: CODE FOR lib main.dart……………………………………………………..….23

Figure 20: CODE FOR chefs curated…………………………………………………….....25

Figure 21: CODE FOR indian cusines………………………………………………..………30

.

# LIST OF SCREEN SHOTS

SS 1: App interface (i)..............................................................................*.*21

SS 2: App interface (ii)....................................................................................21

SS 3: App interface (iii)..............................................................................22

SS 4: App interface (iv)……………………………………………………...22

# ABSTRACT

A mobil eszközök használata az elmúlt évtizedben jelentősen megnőtt. Mindezek az eszközök olyan alkalmazásokat használnak, amelyeket számukra hoztak létre. Ezek az alkalmazások számos különböző szolgáltatást nyújthatnak, beleértve a közösségi médiát, a zene streamelést, a videó streamelést, a megosztott utazás lehetőségét, az online vásárlást és a videojátékokat. Ezeknek az alkalmazásoknak egy része folyamatosan csatlakoznia kell az internethez a megfelelő működéshez, míg mások offline is működhetnek.

Ez a dokumentum egy ételreceptekre vonatkozó, platformfüggetlen alkalmazást mutat be, amely segít a felhasználóknak különböző ételreceptek megtalálásában és megtekintésében különböző kategóriák alapján, valamint lehetővé teszi számukra, hogy intelligens keresési szűrők és kategóriák segítségével keressenek recepteket a kényelmük szerint. A felhasználók szűrhetik a receptek listáját a receptben használt összetevők, az előkészítési idő, a főzési idő és az étrend alapján. Az alkalmazás célja, hogy hatékonyan működjön, miközben elegáns felhasználói élményt nyújt az online alkalmazások alapvető funkcióival, ugyanakkor teljesen offline maradjon.

# 1. BEMUTATKOZÁS

Manapság biztonságosan kijelenthetjük, hogy a világ globális faluvá vált, ahol minden elérhető a technológia révén. A mobiltelefon megjelenése sok ember életét átalakította. Nehéz elképzelni egy napot anélkül, hogy ellenőriznénk és újra ellenőriznénk a közösségi média fiókjainkat. Ez csak azt erősíti meg, hogy a mobilalkalmazások már beépültek az életünkbe. A kutatások azt mutatják, hogy a rendszeres, otthon főzött családi étkezések összefüggésben állnak az egészségesebb és boldogabb gyerekekkel, valamint a serdülőkkel, akik kevésbé valószínű, hogy alkoholt, drogokat vagy cigarettát fogyasztanak. Butler szerint: „Az elmúlt években az okostelefonok megjelenése megváltoztatta a mobiltelefonok definícióját. A telefon már nem csak kommunikációs eszköz, hanem az emberek kommunikációjának és mindennapi életének elengedhetetlen része. A különféle alkalmazások korlátlan szórakozást adtak az emberek életéhez. Biztosak lehetünk benne, hogy a hálózat jövője a mobil terminál lesz.”

Problémafelvetés:

Akár szeretik a főzést, akár csak az evést, mindenki rendelkezik olyan ételek és receptek gyűjteményével, amelyeket ki szeretne próbálni. Talán van néhány, amit egy szeretett személytől örököltek. Bármelyik esetben is, biztosan jobb módszerre van szükségük a hosszú távú tároláshoz, mint egy csomó indexkártya egy irattartóban, ami régi és fárasztó. Ezért a főzés a telefonnal sokkal ízletesebb, ha a megfelelő receptek állnak rendelkezésre.

A felnőttek is jelentős előnyöket élveznek a házilag főtt ételek fogyasztásából. A kutatások azt mutatják, hogy azok az emberek, akik rendszeresen házilag főtt ételeket fogyasztanak, általában boldogabbak és egészségesebbek, kevesebb cukrot és feldolgozott élelmiszert fogyasztanak, ami magasabb energiaszinthez és jobb mentális egészséghez vezethet. A heti öt vagy több napon házilag főtt ételek fogyasztása akár hosszabb élettartammal is összefügg.

Célok:

Ennek a projektnek a célja egy "Recetté" mobilalkalmazás kifejlesztése, amelyet a felhasználók konyhai készségeinek fejlesztésére lehet használni, ahelyett, hogy fájlokban indexkártyákat használnának. Más szóval, a Recetté a telefonodat zsebszakácsává teszi.

Motiváció:

A Recetté alkalmazás egy nagyon hasznos app azoknak, akik szeretnek főzni és kipróbálni új recepteket. Lehetővé teszi a felhasználók számára, hogy kereshessenek és menthessenek recepteket egy adatbázisból, valamint törölhessenek olyan recepteket, amelyekre már nincs szükségük. Ez az alkalmazás időt takarít meg, mivel néhány kattintással recepteket biztosít. A felület tiszta és egyszerű. Kihasználja a Flutter konténerek képességét, hogy képi ikonokkal megjelenítsen opciókat a kezdőképernyőn. A felhasználó kereshet recepteket, megtekintheti a hozzáadott kedvenc receptjeinek listáját, mindezt a kezdőképernyőről.

# 2. LITERATURE SURVEY

Existing System:

Surely there are many recipe apps that offer wide varieties of options to choose from, but most of them are online only and those that are offline lack either that multi-tude of options and devoid of proper UX and structure.

The feasibility study, as its name indicates, aims at assessing the practicality of the proposed project. As stated in the Initial Specifications Report, the objective of this project is to develop a “Recetté” mobile application to be used to elevate user’s kitchen skills and streamline grocery shopping, instead of using index cards in a file folder.

Proposal system:

We are developing an application to help people who are having problems with internet, while retaining all the features that makes an online app great and more features than some existing ones.

We are using dart and flutter to develop our native app to both iOS, Android, etc...,. Dart/futter allows devs to developed native apps for multiple platforms from a single code base.

Summary:

The goal of this step is to ensure that the requirements are consistent, precise and complete to ensure that we meet the final outcome expectations. There are two types of requirements: functional and non-functional requirements. The functional requirements are the ones that describes the functions of the app; whereas, the non-functional requirements are the ones that present the app constraints and properties.

# 3. SOFTWARE REQUIRMENTS SPECIFICATIONS

External Interface Requirements:

* Hardware

The mobile app will be operating on Android and iOS.

* Software

The mobile app will be compatible with the mobile and tablet(Android app) last versions.

DeveloperRequirements**:**

* Hardware Requirements:
* OS : MacOS-64bit and Windows 7 (64bit) or later
* HDD :1.23GB if windows or 2.8GB if MacOS for Flutter SDK
* RAM :4GB (minimum)
* Processor : Intel i3 3rdgen or AMD equivalent
* SoftwareRequirements:
* OS : MacOS-64bit and Windows 7 (64bit) or later
* Tools : Flutter SDK
* Android Studio/Visual Studio Code or any other IDE supporting dart
* Git
* Windows Powershell 5.0(only for windows users)
* User Requirements:
* OS : Android Jellybean or newer and iOS 8 or newer
* Hardware : iOS devices (iPhone 4s or newer) and ARM android devices
* Storage Space : ~200MB
* RAM : 2GB

Non-Functional Requirements:

Non-functional analysis contains about the application analysis of the user's interest level. In

this section will explain how the application is tested in such a way and get the user's assessment

of the appropriateness and condition of the application.

* Security:

The data within the application must be protected from tampering.

* Availability:

The application will must available to the user without internet connection at all times.

* Usability:

It has good UX/UI with elegant design and enough information to allow the user to fully use the application.

* Performance:

The application must be responsive and should match the refresh-rate of the screen.

* Product Requirements:
* Usability Requirements:
  + - The application shall be easy to use and intuitive.
    - The application shall have a user-friendly interface.
    - GUI shall be simple and clear.
* Performance Requirements:
  + - The application shall be fast and robust when loading.
    - The program shall not allow more than 10 min/year of failure.
* Space Requirements:
* The application shall have enough memory space in order to store high number of data.
* Reliability Requirements:
* The application shall not produce an incorrect output.
* Portability Requirements:
* The software shall work in all different platforms.
* Organizational Requirements:
* Implementation Requirements:

-The application shall be implemented using Android Studio.

* External Requirements:
* Inter-operability Requirements.

- The application shall allow access to the different department of the application without altering its efficiency and consistency.

* Ethical Requirements:

- The application shall be license free.

# 4.SYSTEM DESIGN

System architecture:

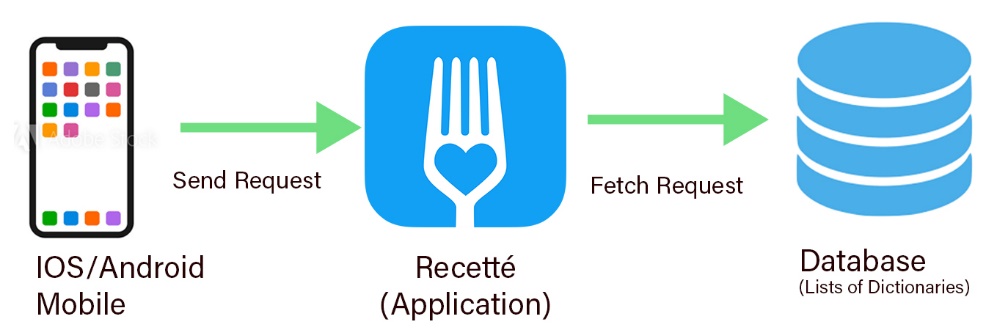


Figure1: System architecture

System architecture describes the organization of the system in terms of structure and behaviour by representing the different components and the relationship between them. The architecture of this system is composed of a mobile device (iOS/Android) that allows the users to make use of the application and its different functionalities of the system.

The Database is part of the application is composed of various lists of dictionaries.

Whenever the user sends a request, the application sends the appropriate parameters based on the request to List\_builder that generates appropriate response by making use of the database.

Use Case Diagrams:

1. Use case diagram for developer:

Graphical user interface, application

Description automatically generated

*Figure 2: Use Case Diagram for Developer*

1. Use case diagram for User:

A picture containing text, electronics

Description automatically generated

Figure 3: Use Case Diagram for User

Entity Relationship Diagram:

This flowchart illustrates how “entities” such as Users, objects or concepts relate to each other within a system. They mirror grammatical structure, with entities as nouns and relationships as verbs.

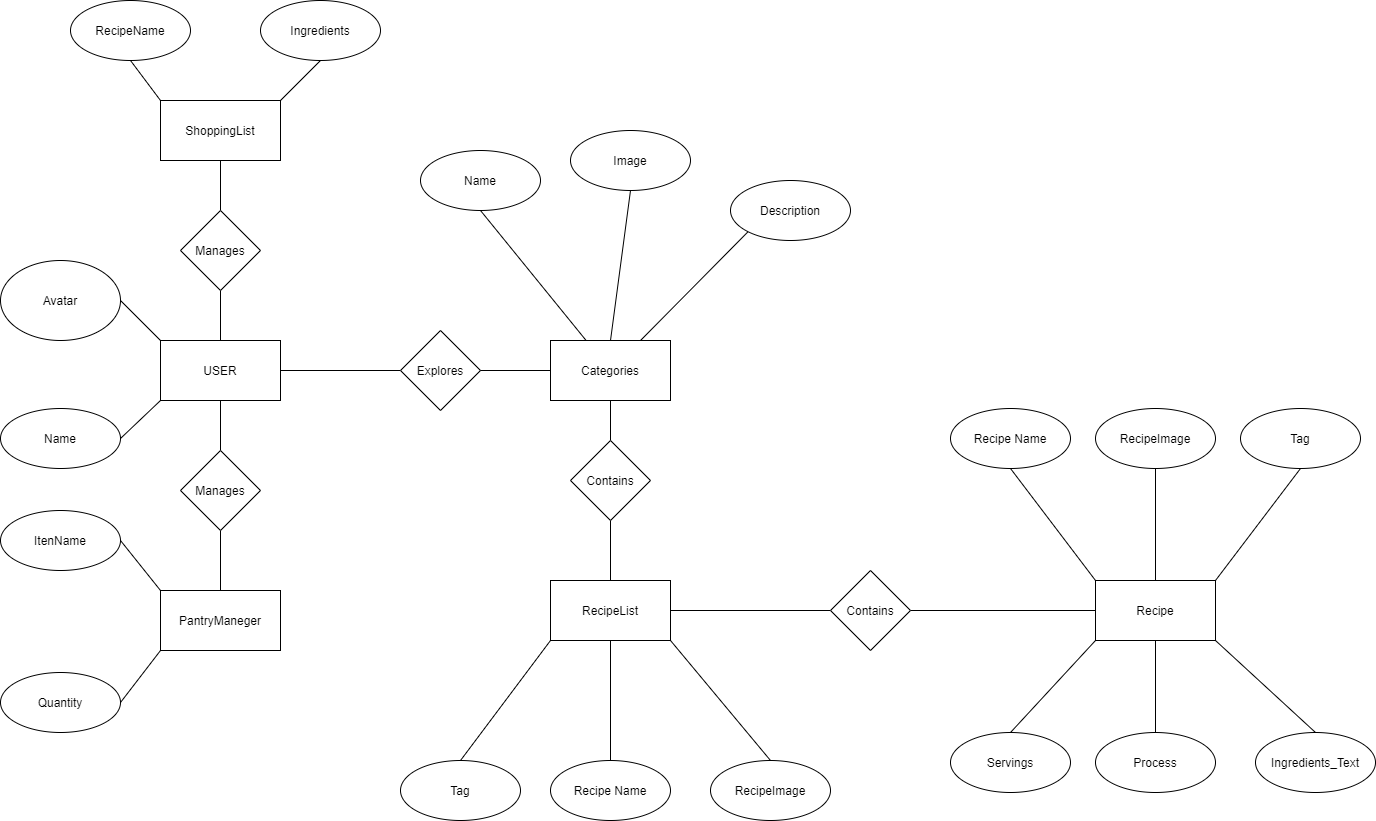


Figure 4: Entity Relationship Diagram

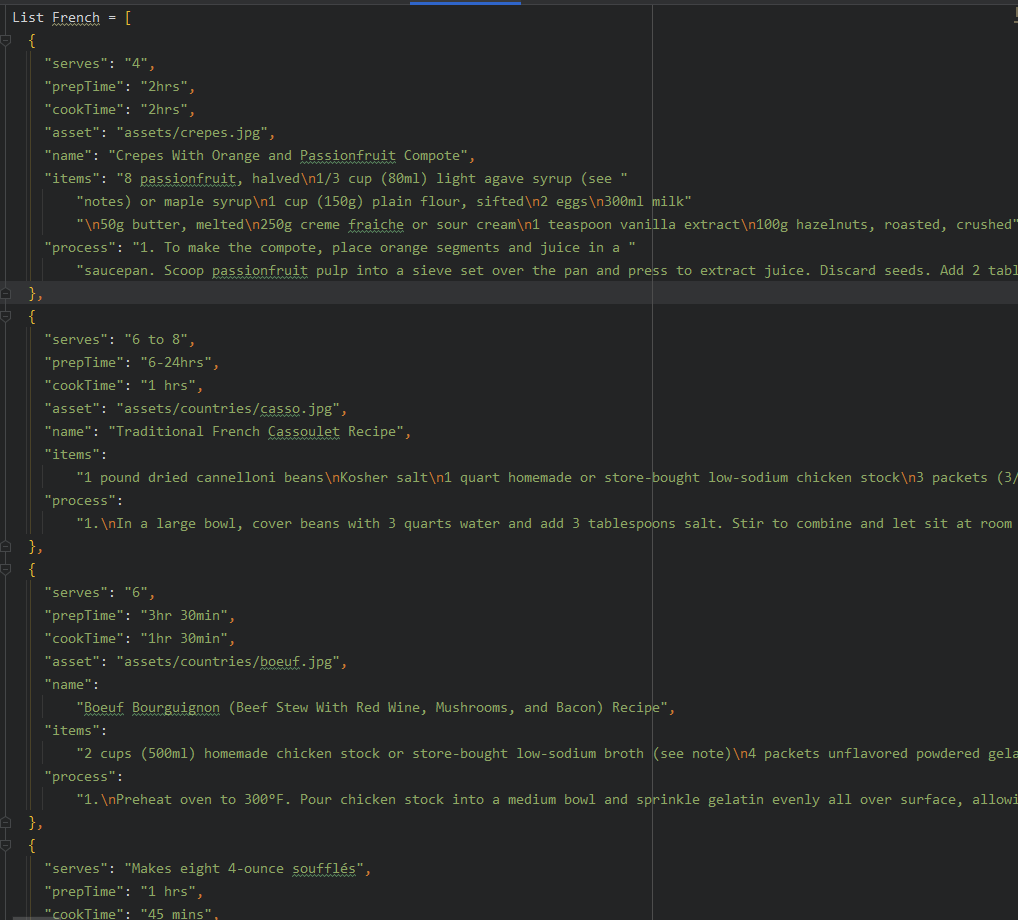
Database Entries:******

Figure 5: Database entries of French cuisine

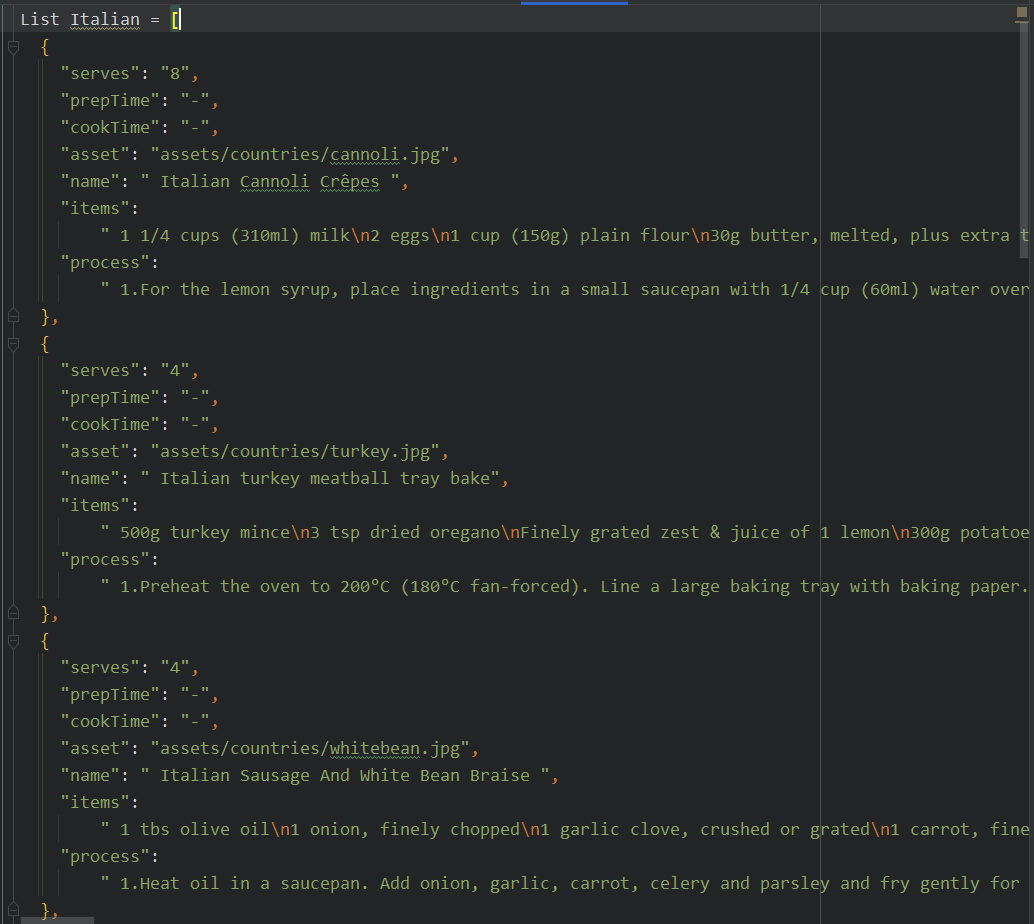
******

Figure 6: Database entries of Italian cuisine

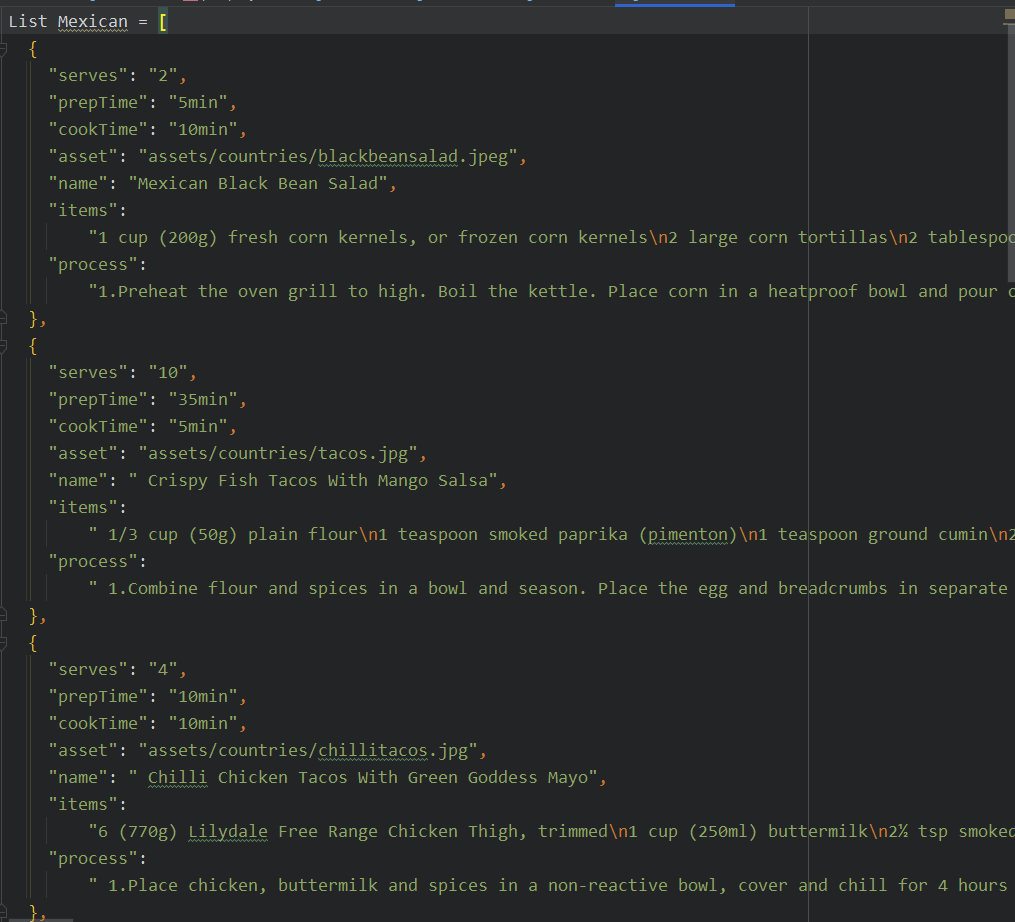
******

Figure 7: Database entries of Mexican cuisine

Android Application Design:

The Android’s user interface is built around Activities which are single focused thing that a user can do. They are directly related to the functional requirement defined for the application. Following are the derived, from the user requirements, activities that should be implemented to fulfil the functional requirements.

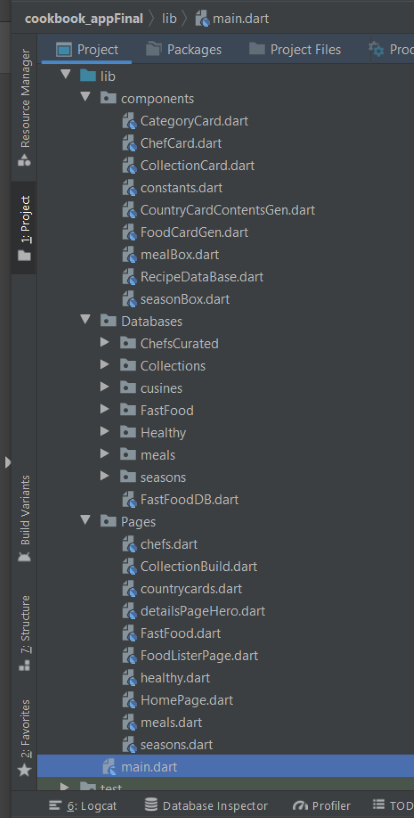


Figure 8: Android Activities

Activity Diagram:

This is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. It is essentially an advanced version of flow chart that modelling the flow from one activity to another activity.

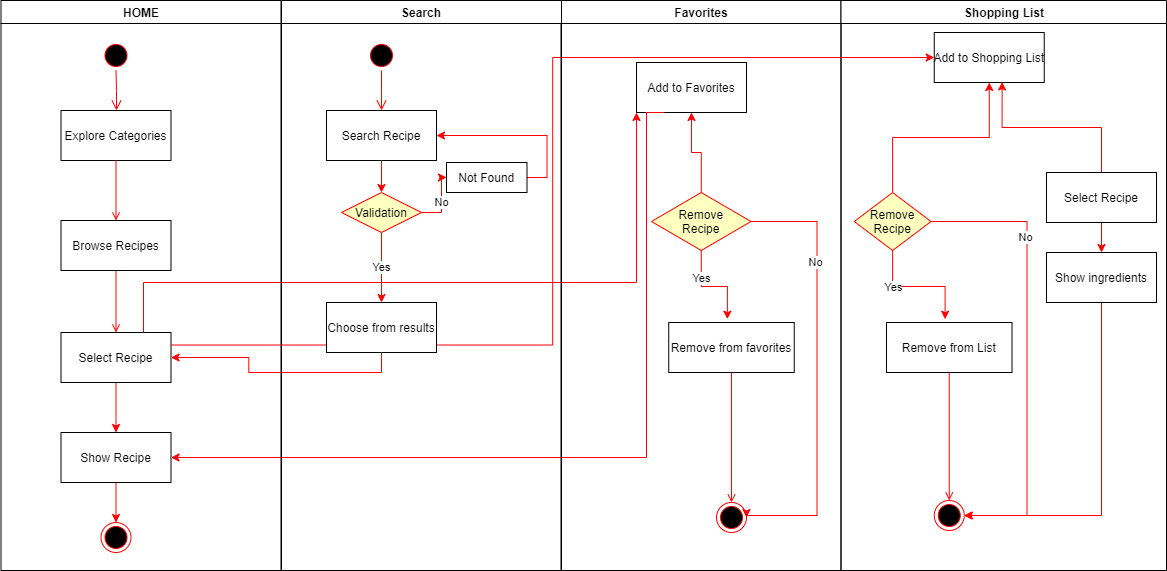


Figure 9: Activity Diagram

Class Diagram:

Class diagrams are the main building block in object-oriented modelling. They are used to show the different objects in a system, their attributes, their operations and the relationships among them. In our example, “recipes” are depicted.

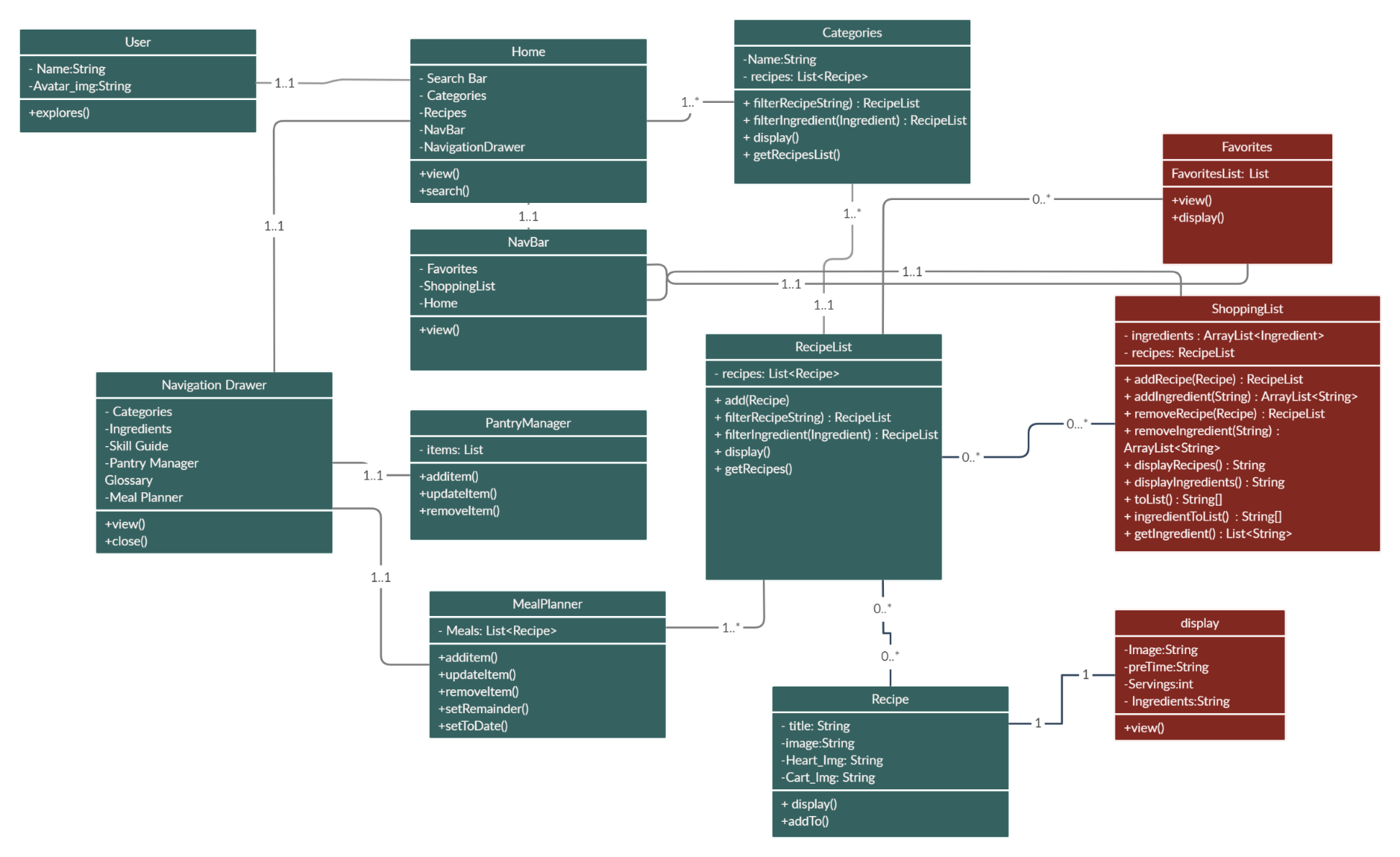


Figure 10: Class Diagram

# 5.IMPLEMENTATION

Environmental Setup:

1. Setting up ‘Environment’ for application
2. Flutter Setup

System requirements:

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

* **Operating Systems**: Windows 7 SP1 or later (64-bit), x86-64 based
* **Disk Space**: 1.32 GB (does not include disk space for IDE/tools).
* **Tools**: Flutter depends on these tools being available in your environment.
  + [Windows PowerShell 5.0](https://docs.microsoft.com/en-us/powershell/scripting/install/installing-windows-powershell) or newer (this is pre-installed with Windows 10)
  + [Git for Windows](https://git-scm.com/download/win) 2.x, with the **Use Git from the Windows Command Prompt** option.

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

Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter\_windows\_1.22.5-stable.zip](https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_1.22.5-stable.zip)

For other release channels, and older builds, see the [SDK releases](https://flutter.dev/docs/development/tools/sdk/releases) page.

1. Extract the zip file and place the contained flutter in the desired installation location for the Flutter SDK (for example, C:\src\flutter).

**Warning:** Do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges.

If you don’t want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the [Flutter repo](https://github.com/flutter/flutter) on GitHub, and change branches or tags as needed. For example:

content*\_* copy

C:\src>git clone https://github.com/flutter/flutter.git -b stable

You are now ready to run Flutter commands in the Flutter Console.

Update your path

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

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

You have to close and reopen any existing console windows for these changes to take effect.

**Note:** As of Flutter’s 1.19.0 dev release, the Flutter SDK contains the dart command alongside the flutter command so that you can more easily run Dart command-line programs. Downloading the Flutter SDK also downloads the compatible version of Dart, but if you’ve downloaded the Dart SDK separately, make sure that the Flutter version of dart is first in your path, as the two versions might not be compatible. The following command (on macOS, linux, and chrome OS), tells you whether the flutter and dart commands originate from the same bin directory and are therefore compatible. (Some versions of Windows support a similar where command.)

*content\_copy*

$ which flutter dart

/path-to-flutter-sdk/bin/flutter

/usr/local/bin/dart

As shown above, the two commands don’t come from the same bin directory. Update your path to use commands from /path-to-flutter-sdk/bin before commands from /usr/local/bin (in this case). After updating your shell for the change to take effect, running the which or where command again should show that the flutter and dart commands now come from the same directory.

*content\_copy*

$ which flutter dart

/path-to-flutter-sdk/bin/flutter

/path-to-flutter-sdk/bin/dart

To learn more about the dart command, run dart -h from the command line, or see the [dart tool](https://dart.dev/tools/dart-vm) page.

Run flutter doctor

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

content*\_* copy

C:\src\flutter>flutter doctor

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

For example:

content*\_* copy

[-] Android toolchain - develop for Android devices

• Android SDK at D:\Android\sdk

**✗ Android SDK is missing command line tools; download from https://goo.gl/XxQghQ**

• Try re-installing or updating your Android SDK,

visit https://flutter.dev/setup/#android-setup for detailed instructions.

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

**Note:** If flutter doctor returns that either the Flutter plugin or Dart plugin of Android Studio are not installed, move on to [Set up an editor](https://flutter.dev/docs/get-started/editor?tab=androidstudio) to resolve this issue.

**Warning:** The flutter tool uses Google Analytics to anonymously report feature usage statistics and basic [crash reports](https://github.com/flutter/flutter/wiki/Flutter-CLI-crash-reporting). This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, type flutter config --no-analytics. To display the current setting, type flutter config. If you opt out of analytics, an opt-out event is sent, and then no further information is sent by the Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service. Note: The Google [Privacy Policy](https://policies.google.com/privacy) describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and crash reports to Google.

Android setup

**Note:** Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step discusses that.

Install Android Studio

1. Download and install [Android Studio](https://developer.android.com/studio).
2. Start Android Studio, and go through the ‘Android Studio Setup Wizard’. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.

Set up your Android device

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

1. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the [Android documentation](https://developer.android.com/studio/debug/dev-options).
2. Windows-only: Install the [Google USB Driver](https://developer.android.com/studio/run/win-usb).
3. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
4. In the terminal, run the flutter devices command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your adb tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the ANDROID\_SDK\_ROOT environment variable to that installation directory.

Set up the Android emulator

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

1. Enable [VM acceleration](https://developer.android.com/studio/run/emulator-acceleration) on your machine.
2. Launch **Android Studio**, click the **AVD Manager** icon, and select **Create Virtual Device…**
   * In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device…**. (The **Android** submenu is only present when inside an Android project.)
   * If you do not have a project open, you can choose **Configure > AVD Manager** and select **Create Virtual Device…**
3. Choose a device definition and select **Next**.
4. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86\_64* image is recommended.
5. Under Emulated Performance, select **Hardware - GLES 2.0** to enable [hardware acceleration](https://developer.android.com/studio/run/emulator-acceleration).
6. Verify the AVD configuration is correct, and select **Finish**.

For details on the above steps, see [Managing AVDs](https://developer.android.com/studio/run/managing-avds).

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

Module Description:

The implementation took roughly three months and involved coding every single day to finally arrive at the final product. The implementation was about designing activities, which are similar to pages in web applications, creating and linking the graphical user interface with the functionalities. Every time we implemented a functionality, we tested it directly on the AVD emulator and my android phone so as to get a real life representation of what the application would look like on our clients’ mobile phones. We also managed to make the mobile application compatible with phones supporting different versions. Further, the mobile application has many features:

1.Smart search-filters:

Users can search or filter recipes by name-base ingredients and diet. When user searches for an ingredient, all the related recipes to that ingredient will be shown to the user and search based on diet is also displayed. For example, some people are allergic to some ingredients and wish not to include them in their meal.Sothe user will be able to search the recipe based on diet or other requirement factors.The search filter will return the appropriate recipes depending on the type of the search.

2.Categorized view:

In our application all recipes are categorized into various cuisine, food types, diets and varieties. We collected the exclusive authentic recipes of various countries and their cuisines are categorized on the type of food and the specific diet that the user is seeking for and varieties.

3.Curated Collections:

We’ve gathered recipes from famous chefs around the world and listed their signature recipes and collected various information and data about these recipes from popular website and articles.

4.Pantry Manager:

Every user can manage their own personal pantry with the help of a pantry manager, which we have included in our application.

5.Meal Planner:

One of the best solutions for busy people to eat better at home is to plan and prep ahead. Preparing nutritious and delicious meals at home is especially critical these days when eating out and food shopping are more challenging. For many people, it can be hard to get started.So, our Meal planner feature enables the user to plan their meal ahead of time or occasional dishes for special events and set a remainder with meal planner

6.Shopping list:

We have also given a feature where the user can add ingredients into the shopping list. So that the user can track all the missing ingredients and purchase them all at once later when he/she visits for a grocery shopping.These shopping list feature take out all the work for you, whether you want to share a virtual list with your family and friends, or [check out the nutrition facts](https://www.goodhousekeeping.com/health/diet-nutrition/g5047/cheap-healthy-foods/) for whatever you're buying.It can be used and created by more than just a cell phone, as some apps can integrate with smartwatches.

7.Skill Guide and Glossary:

Under the skill guide and glossary section, we’ll provide the user with various information on cooking techniques. Like knife skills, etc. and including a ‘Dictionary’ to understand common culinary terminology. This section is basically dedicated to the starters who does not have experience and match knowledge in cooking.

Friendly and flexible GUI

My app has a friendly and flexible Graphical User Interface. I have used Picasso as an image loader. Also, I made sure that every single layout is scrollable. I have used “Photoshop” as a software to design the buttons, layouts and “Text views” backgrounds.

* View RecipeDiagram

  Description automatically generated

Figure 11: User viewing the recipe

The user shall be able to see the recipes in the application. In other words, the user shall be able to access the Recipe Forum, where all the user’s recipes are posted and ordered by date of creation. Our application allows user to view recipe list with images and title. Single recipe can be viewed with images, recipe title, and details cooking directions.

A bowl of soup

Description automatically generated with medium confidence

Figure 12: Viewing the name of the recipe

* Search for a Recipe

Any registered user shall able to search for a recipe using the title as a criterion. The search functionality will provide the users with a shortcut to find their targeted recipes if posted before by the users.

* View Ingredients

The user of the app shall be able to view ingredients of a certain recipe. This option will allow him/her to mark those ingredients as needed, and ultimately find them in the shopping list.

Text

Description automatically generated

Figure 13: Ingredients of the above recipe

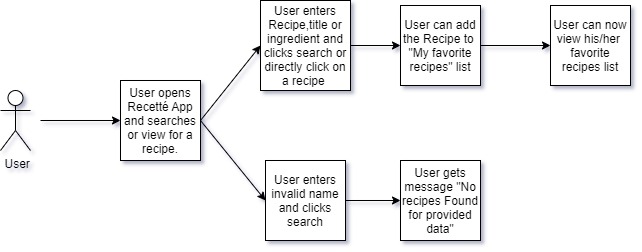


Figure 14: User adding the recipe as his/her favourite

* Add Recipe as Favourite

Once the user accesses the recipe information, he/she shall be able to mark a recipe as favourite. Our application allows user to save recipe as favourite. When user performs search operation, result is list of recipes. Each recipe in list has favourite button on it. User can add recipe by clicking on favourite button. Heart in Figure 31 is to add to favourite button.

* View My Favourite Recipes

The user shall be able to access the recipes that he/she marked as favourite.

Graphical user interface, text, application, chat or text message

Description automatically generated

Figure 15: User can view his /her favourite recipe

Software Description:

The choice of technology enablers that will be used for the development of the application is essential for its success. The technology enablers should provide a suitable way to fulfil the requirements stated before. Principles of enterprise class applications should be kept in mind during the choice of these technologies. The two main ones are that there is no best technology but instead suitable ones and that we shouldn’t reinvent the wheel which means that we should take advantage on what was already implemented and offered to the community.

ANDROID STUDIO:

* Android Studio is Android's official IDE. It is purpose built for Android to accelerate your development and help you build the highest-quality apps for every Android device. It offers tools custom-tailored for Android developers, including rich code editing, debugging, testing, and profiling tools. I have selected Android Studio for the following reasons:
* Instant Run: When you click Run or Debug, Android Studio's Instant Run feature pushes code and resource changes to your running app. It intelligently understands the changes and often delivers them without restarting your app or rebuilding your APK, so you can see the effects immediately.

Intelligent Code Editor **:** The code editor helps you write better code, work faster, and be more productive by offering advanced code completion, refactoring, and code analysis.

* Optimized For All Android Devices: Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

Android Virtual Device (AVD)

* AVD emulator that can be used instead of the Android device. In somecases, it's as good as or better than developing on actual devices.

**Notepad++**

* Notepad++ is a free source code editor and Notepad replacement that supports several languages.

SIMPLE CODE: def localProperties = new Properties ()  
def localPropertiesFile = rootProject.file('local.properties')  
if (localPropertiesFile.exists()) {  
 localPropertiesFile.withReader('UTF-8') **{** reader **->** localProperties.load(reader)  
 **}**}  
  
def flutterRoot = localProperties.getProperty('flutter.sdk')  
if (flutterRoot == null) {  
 throw new GradleException("Flutter SDK not found. Define location with flutter.sdk in the local.properties file.")  
}  
  
def flutterVersionCode = localProperties.getProperty('flutter.versionCode')  
if (flutterVersionCode == null) {  
 flutterVersionCode = '1'  
}  
def flutterVersionName = localProperties.getProperty('flutter.versionName')  
if (flutterVersionName == null) {  
 flutterVersionName = '1.0'  
}  
apply plugin: 'com.android.application'  
apply plugin: 'kotlin-android'  
apply from: "$flutterRoot/packages/flutter\_tools/gradle/flutter.gradle"  
  
android **{** compileSdkVersion 28  
  
 sourceSets **{** main.java.srcDirs += 'src/main/kotlin'  
 **}** lintOptions **{** disable 'InvalidPackage'  
 **}** defaultConfig **{** // *TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).* applicationId "com.hollsinthesun.cookbook"  
 minSdkVersion 16  
 targetSdkVersion 28  
 versionCode flutterVersionCode.toInteger()  
 versionName flutterVersionName  
 **}** buildTypes **{** release **{** // *TODO: Add your own signing config for the release build.* // Signing with the debug keys for now, so `flutter run --release` works.  
 signingConfig signingConfigs.debug  
 **}  
 }**flutter **{** source '../..'  
**}**dependencies **{** implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin\_version"  
**}**

Code for build.gradel

import 'package:flutter/material.dart';  
import 'package:flutter\_test/flutter\_test.dart';  
  
import 'package:cookbook/main.dart';  
  
void main() {  
 testWidgets('Counter increments smoke test', (WidgetTester tester) async {  
 // Build our app and trigger a frame.  
 await tester.pumpWidget(CookBook());  
  
 // Verify that our counter starts at 0.  
 expect(find.text('0'), findsOneWidget);  
 expect(find.text('1'), findsNothing);  
  
 // Tap the '+' icon and trigger a frame.  
 await tester.tap(find.byIcon(Icons.add));  
 await tester.pump();  
  
 // Verify that our counter has incremented.  
 expect(find.text('0'), findsNothing);  
 expect(find.text('1'), findsOneWidget);  
 });  
}

code for widget\_test.dart

<?xml version="1.0" encoding="UTF-8"?>  
<module type="JAVA\_MODULE" version="4">  
 <component name="FacetManager">  
 <facet type="android" name="Android">  
 <configuration>  
 <option name="ALLOW\_USER\_CONFIGURATION" value="false" />  
 <option name="MANIFEST\_FILE\_RELATIVE\_PATH" value="/app/src/main/AndroidManifest.xml" />  
 <option name="RES\_FOLDER\_RELATIVE\_PATH" value="/app/src/main/res" />  
 <option name="ASSETS\_FOLDER\_RELATIVE\_PATH" value="/app/src/main/assets" />  
 <option name="LIBS\_FOLDER\_RELATIVE\_PATH" value="/app/src/main/libs" />  
 <option name="PROGUARD\_LOGS\_FOLDER\_RELATIVE\_PATH" value="/app/src/main/proguard\_logs" />  
 </configuration>  
 </facet>  
 </component>  
 <component name="NewModuleRootManager" inherit-compiler-output="true">  
 <exclude-output />  
 <content url="file://$MODULE\_DIR$">  
 <sourceFolder url="file://$MODULE\_DIR$/app/src/main/java" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/app/src/main/kotlin" isTestSource="false" />  
 <sourceFolder url="file://$MODULE\_DIR$/gen" isTestSource="false" generated="true" />  
 </content>  
 <orderEntry type="jdk" jdkName="Android API 25 Platform" jdkType="Android SDK" />  
 <orderEntry type="sourceFolder" forTests="false" />  
 <orderEntry type="library" name="Flutter for Android" level="project" />  
 <orderEntry type="library" name="KotlinJavaRuntime" level="project" />  
 </component>  
</module>

Code for gridle android app

import 'package:cookbook/Pages/HomePage.dart';  
import 'package:flutter/material.dart';  
  
void main() {  
 runApp(CookBook());  
}  
  
class CookBook extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 title: "Recette",  
 theme: ThemeData.light(),  
 home: HomePage(),  
 );  
 }  
}

code for lib main.dart

List vikas = [  
 {  
 "serves": "4",  
 "prepTime": "10 minutes",  
 "cookTime": "45 minutes",  
 "asset": "assets/chefs/vikas/khubani.jpeg",  
 "name": "Khubani ka Shahi Tukda",  
 "items":  
 "For the rabdi:\n3 cups whole milk\n1/2 cup sugar\n1/4 tsp green cardamom powder\nFor the shahi tukda:\n1 cup granulated sugar\n1 cup whole milk\nGhee, for deep frying\n1/4 tsp saffron strands\n4 slices white bread\n100 gms dried apricots, soaked overnight and drained\n2 Tbsp pistachios, finely chopped\n2 sheets edible silver/gold leaf (Optional)",  
 "process":  
 "For the rabdi:\n1.Heat milk in a kadhai (wok) over medium heat and bring to a boil. Lower the heat and cook, stirring continuously, till it thickens to a consistency of a custard sauce.\n2.Remove from heat, add sugar and cardamom powder and stir till sugar dissolves. \nThe mixture will thicken on cooling.\nFor the shahi tukda:\n1.Combine the sugar with 2 cups water in a saucepan and cook over medium heat, stirring till sugar dissolves and the syrup thickens to one-string consistency.\n2.Add the drained apricots to the syrup and cook till tender. \nRemove apricots from syrup and set aside.\n3.In another saucepan, combine the milk with saffron and prepared sugar syrup (The syrup in which the apricots were cooked) and simmer for 5 to 7 minutes. \nRemove from heat and keep warm.\n4.Heat the clarified butter in a kadhai over medium heat, and deep fry the bread slices till golden and crisp. \nRemove and drain on absorbent paper.\n5.Soak the fried bread slices in the warm milk for 2 minutes.\n6.Sandwich the cooked apricots between the softened bread.\n7.To serve, arrange the sweet apricot sandwiches on a plate and spoon rabdi over. Sprinkle over with pistachios and garnish with silver leaf.\n8.Serve warm."  
 },  
 {  
 "serves": "4",  
 "prepTime": "10 minutes",  
 "cookTime": " 20 minutes",  
 "asset": "assets/chefs/vikas/kacche.jpeg",  
 "name": "Kacche Kele ki Asharfi Recipe",  
 "items":  
 "2 Tbsp oil\n2 large onions, finely sliced\n6 unripe green bananas, boiled in their skins\n4 potatoes, boiled\n1 1/2 Tbsp ginger-garlic paste\n2 Tbsp fresh coriander, finely chopped\n2 tsp red chilli powder\n2 Tbsp gram flour (besan)\n1/2 tsp green cardamom powder\nSalt, to taste\n2 medium capsicums, tops cut off and cleared of seeds and white pith\n1 1/2 clarified butter (ghee)",  
 "process":  
 "1.Heat oil in a fry pan over medium heat. Add onions and fry golden. Remove and drain on absorbent paper. \nFinely chop the browned onions.\n2.Peel the bananas and potatoes and mash well.\n3.In a mixing bowl, combine the mashed bananas and potatoes with chopped browned onions, ginger-garlic paste, chopped coriander, chilli powder, gram flour, cardamom powder & salt and mix thoroughly.\n4.Fill this mixture into the cleaned hollow capsicums and stuff well. \n Using a sharp knife cut the stuffed capsicum horizontally into 2-inch thick slices.\n5.Heat the clarified butter in a frying pan and shallow-fry the stuffed capsicum slices till the capsicum softens slightly and the filling is golden and crisp.\n6.Remove, drain on absorbent paper and serve hot with coriander and mint chutney."  
 },  
 {  
 "serves": "2",  
 "prepTime": "0",  
 "cookTime": "30 minutes",  
 "asset": "assets/chefs/vikas/cafreal.jpeg",  
 "name": "Chicken Cafreal",  
 "items":  
 "2 pieces chicken\n1 cup water\n1/2 tsp salt\n2 Tbsp chopped coriander\n1 piece cinnamon\n6 cloves\n1 small piece ginger\n5 green chillies\n6 pieces garlic\n1/2 tsp cumin seeds\n2 pieces cardamom\n1 tsp vinegar\n4 tsp white rum\n4 Tbsp oil\n10 onion rings",  
 "process":  
 "1.Add salt to chicken and cook in water on full flame for 3 minutes.\n2.Grind coriander, cinnamon, cloves, ginger, green chillies, garlic, cumin seeds, cardamom, vinegar and white rum to a paste.\n3.Add paste, two tbsp oil and onion rings to the cooking chicken and reduce flame to low.\n4.After ten minutes add two more tbsp oil and white rum.\n5.Simmer on fire till all the water evaporates and the chicken readies.\n6.Serve hot with salad and chips."  
 },  
 {  
 "serves": "2",  
 "prepTime": "10 minutes",  
 "cookTime": "40 minutes",  
 "asset": "assets/chefs/vikas/piccata.jpeg",  
 "name": "Chicken Piccata With Bread Salad",  
 "items":  
 "2 Chicken breast (batted thinly)\n1/3 Cup Flour\n1 Egg\n1/2 Cup Parmesan cheese\n1 tsp Parsley\n1 tsp Chives\n1 Tbsp Butter\n1 tsp Olive oil\nTo taste Salt & pepper\nFor Bread Salad:\n1 Small Ciabata bread loaf\n2 Tomatoes\n8 Olives\n1/2 Bell peppers\n3 Basil leaves\n1/2 Cup Olive oil\n2 tsp Red wine vinegar\n1/2 Cup Tomato juice\n1/3 Cup Sun dried tomatoes",  
 "process":  
 "Prepare the Chicken:\n1.Season the chicken breasts with salt and pepper. \nMake a batter with the Parmesan cheese, parsley, egg, chives.\n2.Beat well. Dip the seasoned chicken into the flour and then into the batter.\n3.Shallow fry in a mix of olive oil and butter until golden brown.\nPrepare the Salad:\n1.Cut the bread into chunks. Sprinkle some olive oil, and bake for 6 minutes.\n2.Set aside to cool slightly.\n3.In a bowl mix the tomatoes, olives, yellow bell peppers and the sun-dried tomatoes. \nSeason to taste with salt and cracked black pepper.\n4.Whisk together the tomato juice, red wine vinegar, olive oil, and the basil. \nPour over the bread. Add it to the tomato and olive mixture."  
 },  
 {  
 "serves": "2",  
 "prepTime": "0",  
 "cookTime": "30 minutes",  
 "asset": "assets/chefs/vikas/breakfast.jpeg",  
 "name": "French Breakfast Radish with Mustard Dressing",  
 "items":  
 "1 Small Baguette\n6 oz Fingerling Potatoes\n3 oz French Breakfast Radishes\n1 Fennel Bulb With Fronds\n1 head Butter Lettuce\n1 large bunch Mint\n1 Shallot\n1 Tbsp Red Wine Vinegar\n1 Tbsp Whole Grain Mustard\n¼ cup Red Quinoa\n4 oz Goat Cheese",  
 "process":  
 "1Cook the quinoa:\nPreheat the oven to 450°F. Heat 2 medium pots of salted water to boiling on high. Once the first pot of water is boiling, add the quinoa. \nCook 14 to 16 minutes, or until tender. \nDrain thoroughly and transfer to a large bowl. Set aside.\nPrepare the ingredients:\n2Prepare the ingredients:\nWhile the quinoa cooks, wash and dry the fresh produce. \nCut the baguette in half horizontally, then crosswise on an angle. \nCut off and discard the root end of the lettuce; separate the leaves. \nCut the potatoes into ½-inch-thick rounds. Trim off and discard the stem ends of the radishes; \ncut the radishes into bite-sized wedges. Cut off and discard the fennel stems and fronds; \nthinly slice the bulb. Pick the mint leaves off the stems; discard the stems. \nPeel and mince the shallot to get 2 tablespoons of minced shallot (you may have extra); \nplace the minced shallot in a medium bowl with the vinegar.\nCook the potatoes:\n3Cook the potatoes:\nAdd the potatoes to the second pot of boiling water. \nCook 8 to 10 minutes, or until tender when pierced with a fork. \nRemove from heat. Thoroughly drain the potatoes and set aside to cool slightly.\nMake the dressing:\n4Make the dressing:\nWhile the potatoes cook, add the mustard to the shallot-vinegar mixture. \nSlowly whisk in 2 tablespoons of olive oil until well combined; \nseason with salt and pepper to taste. \nSet aside.\nMake the goat cheese toast:\n5Make the goat cheese toast:\nWhile the potatoes continue to cook, place the baguette pieces on a sheet pan, cut sides up. \nDrizzle with olive oil and season with salt and pepper. Toast in the oven 6 to 8 minutes, or until golden brown. \nRemove from the oven and set aside to cool slightly. \nWhen cool enough to handle, spread the goat cheese onto the toasted baguette pieces. \nDrizzle with olive oil and season with salt and pepper to taste.\nFinish & plate your dish:\n6Finish & plate your dish:\nTo the bowl of cooked quinoa, add the cooked potatoes, radishes, fennel, lettuce and mint; \nseason with salt and pepper. Add enough of the dressing to coat the salad (you may have extra dressing); \ntoss to thoroughly combine and season with salt and pepper to taste. \nDivide the finished salad between 2 plates. \nServe with the goat cheese toast on the side. \nEnjoy!",  
 }  
];

Code for Chefs Curated

Text

Description automatically generated

Result for above code in android app

List Indian = [  
 {  
 "serves": "6",  
 "prepTime": "2 hours 45 mins",  
 "cookTime": "45 mins",  
 "asset": "assets/countries/indian/dum.jpg",  
 "name": "Hyderabadi dum biriyani",  
 "items":  
 "For Chicken Marination:\n11/2 kg chicken cut into large pieces\n2 tbsp ginger garlic paste\n2 tbsp red chilli powder\n1 tsp turmeric powder\nsalt as per taste\n1 bunch coriander leaves\n1 bunch mint leaves\n7 - 8 green chillies\n1 tsp caraway seeds shahi zeera\n3 - 4 cloves loung\n2 cinnamon sticks dalchini\n3 - 4 cardamoms elaichi\n3 - 4 black pepper corns kali mirch\n2 cups fried golden brown onions\n2 cups yogurt\n1 tbsp oil\n1 tbsp lemon juice\ntsp garam masala powder\nFor the Rice:\nWater for boiling\n1 kg basmati rice\n2-3 cloves loung\n1 cinnamon stick dalchini\n2-3 cardamoms elaichi\n1 tsp caraway seeds shahi zeera\n2 - 3 black pepper corns kali mirch\n1 tbsp lemon juice\nsalt - to taste\nFor Dum Seasoning:\n2 tbsp oil\n2 tbsp coriander leaves\n2 tbsp mint leaves\n2 tbsp fried onions\n1 tbsp lemon juice\n1 cup saffron flavored milk\n1 tbsp ghee\n1 cup water\nFor Garnishing:\nFried onions",  
 "process":  
 "For The Chicken Marination:\nWash the chicken pieces well with lemon juice and water and strain the chicken completely.\nAdd ginger garlic paste into it.\nAdd red chilli powder, turmeric powder, salt and mix well.\nIn a blender add a bunch of coriander leaves, a bunch of mint leaves, green chillies, add a pinch of salt and make the paste called as green paste or hara masala paste.\nAdd the green paste to the chicken marinade.\nAdd whole garam masala that is cardamom, cinnamon, cloves, black pepper corns, caraway seeds.\nMix the whole mixture well.\nCrush the onions and add into the marinade.\nAdd beaten yogurt and mix well.\nAdd oil over the marinade and mix well.\nAdd garam masala powder and lemon juice and mix well.\nCheck the taste of salt and add if required.\nRest this marinade inside the refrigerator for about 45 mins to 1 hour.\nProcedure To Cook The Rice:\nFirstly, wash the basmati rice and soak it for about 35 mins.\nTake a cooking pot, add water generously into it.\nInto the water, add salt as per taste, cinnamon stick, cloves, shahi zeera, black pepper corns, cardamoms.\nBring the water to a boil.\nWhen the water starts boiling, add the soaked rice into it.\nCook the rice for about 6 - 7 mins until it is three fourth done.\nStrain the rice completely and rest the strained rice aside.\nProcedure To Give The Dum:\nIn a cooking pot, to the base add oil and spill it all over the base.\nAdd the chicken marinade into it and spread it evenly.\nAdd the rice which has been cooked into it.\nSpread the rice evenly all over the marinade.\nSeason the rice with coriander leaves, mint leaves, fried onions.\nAdd lemon juice all over the rice.\nAdd saffron flavored milk(2 saffron strands added into the 1/4th cup of milk to get orange red color).\nAdd ghee into it.\nAdd little water all over the rice(optional).\nMake a chapathi dough and seal the lid and the cooking pot tightly with the dough.\nSwitch on the flame, put the cooking pot onto the flame.\nDum the biryani on sim flame for about 45 mins."  
 },  
 {  
 "serves": "4",  
 "prepTime": "10 mins",  
 "cookTime": "25 mins",  
 "asset": "assets/countries/indian/Chicken\_makhani.jpg",  
 "name": "Chicken Makhani (Indian Butter Chicken)",  
 "items":  
 "1 tablespoon peanut oil\n1 shallot, finely chopped\n¼ white onion, chopped\n2 tablespoons butter\n2 teaspoons lemon juice\n1 tablespoon ginger garlic paste\n1 teaspoon garam masala\n1 teaspoon chili powder\n1 teaspoon ground cumin\n1 bay leaf\n¼ cup plain yogurt\n1 cup half-and-half\n1 cup tomato puree\n¼ teaspoon cayenne pepper, or to taste\n1 pinch salt\n1 pinch black pepper\n1 tablespoon peanut oil\n1 pound boneless, skinless chicken thighs, cut into bite-size pieces\n1 teaspoon garam masala\n1 pinch cayenne pepper\n1 tablespoon cornstarch\n¼ cup water",  
 "process":  
 "Step 1\nHeat 1 tablespoon oil in a large saucepan over medium high heat. Sauter shallot and onion until soft and translucent. Stir in butter, lemon juice, ginger-garlic paste, 1 teaspoon garam masala, chili powder, cumin and bay leaf. Cook, stirring, for 1 minute. Add tomato sauce, and cook for 2 mins, stirring frequently. Stir in half-and-half and yogurt. Reduce heat to low, and simmer for 10 mins, stirring frequently. Season with salt pepper. Remove from heat and set aside.\nStep 2\nHeat 1 tablespoon oil in a large heavy skillet over medium heat. Cook chicken until lightly browned, about 10 mins. Reduce heat, and season with 1 teaspoon garam masala and cayenne. Stir in a few spoonfuls of sauce, and simmer until liquid has reduced, and chicken is no longer pink. Stir cooked chicken into sauce.\nStep 3\nMix together cornstarch and water, then stir into the sauce. Cook for 5 to 10 mins, or until thickened."  
 },  
 {  
 "serves": "3",  
 "prepTime": "10 mins",  
 "cookTime": "30 mins",  
 "asset": "assets/countries/indian/bisibela.jpg",  
 "name": "Bisi Bela Bath",  
 "items":  
 "BISI BELE BATH MASALA:\n4 tsp coriander seeds\n4 tsp chana dal\n2 tsp urad dal\n1 tsp jeera / cumin\n¼ tsp methi / fenugreek seeds\n½ tsp pepper\n3 pods cardamom / each\n1 inch cinnamon / dalchini\n4 cloves / lavang\n2 tbsp dry coconut / copra\n2 tsp poppy seeds / khus khus\n1 tsp sesame seeds / til\n1 tsp oil\n12 dried kashmiri red chilli\nfew curry leaves\npinch of hing / asafoetida\nOTHER INGRIDENTS:\n½ carrot, chopped\n5 beans, chopped\n2 tbsp matar / peas\n½ potato / aloo, cubed\n2 tbsp peanuts\n2 cups water\n¼ tsp turmeric / haldi\n1½ tsp salt\n¾ cup tamarind extract\n½ tsp jaggery / gud\n½ onions, petals\n1 cup toor dal, cooked\n2½ cup rice, cooked\n1 cup water\n1 tbsp ghee / clarified butter",  
 "process":  
 "firstly, cook vegetables, 2 tbsp peanuts, 2 cups water, ¼ tsp turmeric and 1 tsp salt.\nnow add ¾ cup tamarind extract, ½ tsp jaggery and ½ onions. boil for 10 mins.\nfurther add 1 cup of cooked toor dal, 2½ cup of cooked rice and 1 cup water.\nadd 4 tsp of bisi bisi bele bath masala and simmer for 20 mins.\npour the tempering and serve bisi bele bath with boodi or mixture."  
 },  
 {  
 "serves": "4",  
 "prepTime": "10 mins",  
 "cookTime": "30 mins",  
 "asset": "assets/countries/indian/matar.jpg",  
 "name": "Matar Paneer",  
 "items":  
 "FOR ONION TOMATO PASTE:\n2 tbsp oil\n1 onion, sliced\n3 clove garlic, chopped\n1 inch ginger\n3 tomato, sliced\nOTHER INGREDIENTS:\n2 tbsp oil\n1 bay leaf\n1 inch cinnamon stick\n2 pods cardamom\n1 tsp cumin / jeera\n¼ tsp turmeric\n1 tsp kashmiri red chilli powder\n1 tbsp besan / gram flour\n¼ tsp cumin powder\n1 tsp coriander powder\n1 tsp salt\n1 cup water\n1 cup peas / matar\n12 cubes paneer / cottage cheese\n2 tbsp coriander, finely chopped\n¼ tsp garam masala\n1 tsp kasuri methi",  
 "process":  
 "in a large kadai heat 2 tbsp oil and saute spices.\nadd ¼ tsp turmeric, 1 tsp chilli powder and 1 tbsp besan. roast well.\nadd in prepared onion tomato paste and saute well.\nfurther add ¼ tsp cumin powder, 1 tsp coriander powder and 1 tsp salt.\nnow add 1 cup water and stir well.\nadd in 1 cup peas and stir well. cover and cook for 10 mins.\nadd in 12 cubes paneer and simmer for 10 mins.\nnow add 2 tbsp coriander, ¼ tsp garam masala and 1 tsp kasuri methi.\nfinally, enjoy matar paneer with roti or rice."  
 },  
 {  
 "serves": "",  
 "prepTime": "15 mins",  
 "cookTime": "40 mins",  
 "asset": "assets/countries/indian/Tandoori-Chicken.jpg",  
 "name": "Tandoori chicken",  
 "items":  
 "3 Tbsp vegetable oil\n1 teaspoon ground coriander\n1 teaspoon ground cumin\n1 teaspoon ground turmeric\n1 teaspoon cayenne\n1 Tbsp garam masala\n1 Tbsp sweet (not hot) paprika\n1 cup plain yogurt (can sub buttermilk)\n2 Tbsp lemon juice\n4 minced garlic cloves\n2 Tbsp minced fresh ginger\n1 teaspoon salt\n4 whole chicken legs (drumsticks and thighs), or its equivalent, skinless, bone-in",  
 "process":  
 "1 Heat the spices in oil: Heat the oil in a small pan over medium heat, then cook the coriander, cumin, turmeric, cayenne, garam masala and paprika, stirring often, until fragrant (approximately 2-3 mins). Let cool completely.\n2 Whisk spices into yogurt, add lemon juice, garlic, salt, ginger: Whisk the cooled spice-oil mixture into the yogurt, then mix in the lemon juice, garlic, salt and ginger.\n3 Cut deep slashes into chicken, coat with marinade, chill: Cut deep slashes (to the bone) in 3-4 places on the leg/thigh pieces. Just make 2-3 cuts if you are using separate drumsticks and thighs. Coat the chicken in the marinade, cover and chill for at least an hour (preferably 6 hours), no more than 8 hours.\n4 Prepare grill: Prepare your grill so that one side is quite hot over direct heat, the other side cooler, not over direct heat. If using charcoal, leave one side of the grill without coals, so you have a hot side and a cooler side. If you are using a gas grill, just turn on one-half of the burners.\nUse tongs to wipe the grill grates with a paper towel soaked in vegetable oil.\n5 Shake off excess marinade, place chicken on hot side of grill: Take the chicken out of the marinade and shake off the excess. You want the chicken coated, but not gloppy.\nPut the chicken pieces on the hot side of the grill and cover. Cook 2-3 mins before checking.\nTurn the chicken so it is brown (even a little bit charred) on all sides\n6 Move to cool side of grill, cover and cook: Move chicken to the cool side of the grill. Cover and cook for at least 20 mins, up to 40 mins (or longer) depending on the size of the chicken and the temperature of the grill. The chicken is done when its juices run clear.\nLet it rest for at least 5 mins before serving. It’s also great at room temperature or even cold the next day.\nServe with naan, and Indian flatbread, or with Indian style rice, with yogurt-based raita on the side."  
 },  
 {  
 "serves": "6",  
 "prepTime": "20 mins",  
 "cookTime": "45 mins",  
 "asset": "assets/countries/indian/masala\_dosa.jpeg",  
 "name": "Masala Dosa",  
 "items":  
 "FOR BATTER:\n3 cup sona masuri rice\n½ tsp methi / fenugreek seeds\nwater, for soaking\n1 cup urad dal\n2 tbsp toor dal\n2 tbsp chana dal\n1 cup poha / avalakki, rinsed\nFOR ALOO BHAJI:\n2 tbsp oil\n1 tsp mustard\n1 tsp urad dal\n1 tsp chana dal\n1 dried red chilli\nfew curry leaves\npinch hing / asafoetida\n2 chilli, finely chopped\n1 inch ginger, finely chopped\n1 onion, sliced\n¼ tsp turmeric\n1 tsp salt\n3 potato, boiled & mashed\n2 tbsp coriander, finely chopped\n2 tbsp lemon juice",  
 "process":  
 "MASALA DOSA BATTER PREPARATION:\nfirstly, in a large bowl take 3 cup sona masuri rice and ½ tsp methi.\nrinse well and soak in enough water for 4 hours.\nin another bowl take 1 cup urad dal, 2 tbsp toor dal and 2 tbsp chana dal.\nrinse well and soak in enough water for 2 hours.\nafter soaking dal for 2 hours, drain off the water and transfer to the grinder. you can also grind in mixi if you do not have access to a grinder.\nadd water as required and blend to smooth paste.\nscrape sides. the smooth and fluffy batter will be ready after 40 mins.\ntransfer the batter to a large vessel and keep aside.\nin the same grinder add soaked rice and 1 cup rinsed poha.\nadd water slowly and scrape the sides. blend to a coarse paste.\ntransfer the rice batter to the same urad dal batter.\nmix well making sure everything is well combined.\nferment in a warm place for at least 8 hours or until the batter doubles in volume. if you are leaving in a cold climate, then you can place the batter in the warm oven (just heat the oven until it turns slightly warm and then turn off) to ferment.\nonce the batter is well fermented, mix gently, without disturbing the air pockets.\ntransfer 4 cups of fermented batter to a small bowl and add 1 tsp salt.\nmix well until the salt is well combined. masala dosa batter is ready. keep aside.\nALOO BHAJI PREPARATION:\nfirstly, in a large kadai heat 2 tbsp oil and splutter 1 tsp mustard, 1 tsp urad dal, 1 tsp chana dal, 1 dried red chilli, few curry leaves, pinch hing.\nnow add 2 chilli and 1 inch ginger. saute well.\nalso, add 1 onion and saute until onions shrink slightly.\nfurther, add ¼ tsp turmeric and 1 tsp salt. saute well.\nnow add 3 potato and mix well, mash slightly making sure everything is well combined.\nturn off the flame and add 2 tbsp coriander and 2 tbsp lemon juice.\nmix well and aloo bhaji for masala dosa is ready. keep aside.\nMASALA DOSA PREPARATION:\nfirstly, add a ladleful of batter on hot tawa.\nspread as thin as possible making a crispy dosa.\ntake 1 tsp of butter and spread uniformly.\nalso, place 2 tbsp of prepared aloo masala in the centre.\nroast until the dosa turns golden brown and crisp.\nscrape the sides of dosa and roll the dosa.\nfinally, masala dosa recipe is ready to serve with coconut chutney and sambar."  
 },  
 {  
 "serves": "6",  
 "prepTime": "10 mins",  
 "cookTime": "45 mins",  
 "asset": "assets/countries/indian/rogan-josh.jpg",  
 "name": "Rogan Josh",  
 "items":  
 "1 Kg meat\n1 cup mustard/refined oil\n3 tsp red chili powder\n3 tsp fennel powder\n2 tsp ginger powder\n2 tsp cumin powder\n3 tsp brown cardamom powder\n1 tsp asafoetida\n4 Green cardamom\n2 Cinnamon sticks\n2 Bay leaves\n2 Cloves\n1/3 tsp saffron (optional)\n1 cup curd\nA pinch of salt",  
 "process":  
 "1.Wash the meat properly. Heat oil in a pressure cooker.\n2.Put cinnamon, bay leaves, green cardamom, cloves, a teaspoon of salt, asafoetida, and meat together.\n3.Fry meat till it turns brown. Once browned, pour a cup of water.\n4.Add the red chili powder and saffron into the meat. Keep stirring for about a minute.\n5.Mix the curd nicely in the mixer and pour it into the pressure cooker.\n6.Keep on stirring till you get a reddish tinge.\n7.Add 2 cups of water, fennel powder, ginger powder, and pressure cook for 2 mins.\n8.Check if the meat is tender. Peel and grind green and brown cardamom and add to the meat dish.\n9.Finally sprinkle cumin powder and simmer for a minute and serve."  
 },  
 {  
 "serves": "2",  
 "prepTime": "5 mins",  
 "cookTime": "20 mins",  
 "asset": "assets/countries/indian/ven-pongal.jpg",  
 "name": "Pongal",  
 "items":  
 "FOR PRESSURE COOKING:\n1 tsp ghee / clarified butter\n½ cup rice, rinsed\n½ cup moong dal, rinsed\n4 cup water\n½ tsp salt\nFOR TEMPERING:\n2 tbsp ghee / clarified butter\n1 tsp cumin / jeera\n½ tsp pepper, crushed\n1 inch ginger, finely chopped\n2 chilli, slit\n10 cashew / kaju, halves\npinch hing / asafoetida",  
 "process":  
 "firstly, in a pressure cooker heat 1 tsp ghee.\nadd ½ cup rice, ½ cup moong dal and saute for a minute or until it turns aromatic.\nfurther, add 4 cup water and ½ tsp salt. mix well.\ncover and pressure cook for 5 whistles on medium flame.\nonce the pressure settles down, open the cooker and give a good mix.\nfurthermore, in a pan heat 2 tbsp ghee.\nalso add 1 tsp cumin, ½ tsp pepper, 1 inch ginger, 2 chilli, 10 cashew and pinch hing.\nsaute on low flame until cashew turns golden brown.\npour the tempering over cooked rice and dal mixture.\ngive a good mix, add more ghee if required.\nfinally, serve ven pongal/khara pongal with coconut chutney and sambar."  
 },  
 {  
 "serves": "21 pieces",  
 "prepTime": "30 mins",  
 "cookTime": "12 mins",  
 "asset": "assets/countries/indian/Dhokla.png",  
 "name": "Dhokla",  
 "items":  
 "FOR RAVA DHOKLA:\n1 cup rava / bombay rava / semolina / sooji\n1 cup thick curd / yogurt, slightly sour\n½ tsp sugar\n1 tsp ginger / adrak, grated\n1 tsp chili paste\n2 tsp oil\nsalt to taste\n¼ cup water, or as required\n1 tsp eno fruit salt or ¾ tsp baking soda\nFOR TEMPERING:\n2 tsp oil\n1 tsp mustard seeds / rai\n½ tsp jeera / cumin seeds\n½ tsp sesame seeds / til\npinch of hing / asafoetida\nfew curry leaves\n2 green chili, slit\n2 tbsp coriander leaves, finely chopped",  
 "process":  
 " firstly, in a large mixing bowl take rava and curd.\nadditionally add sugar, ginger, chili paste and salt.\nmix well to form a thick batter.\nrest for 30 mins or till rava absorbs moisture.\nfurther add water as required and mix well.\nget the dhokla to idli batter consistency.\nadditionally add eno fruit salt.\nmix gently till the batter turns frothy.\ntransfer to the greased plate.\nsteam the dhokla batter for 12 mins.\nprepare the tempering by heating oil.\nfurther add mustard seeds, jeera, sesame seeds and hing.\nonce the mustard seeds splutter add in curry leaves and green chili.\nsaute for a while and once the tempering splutters spread the tempering over dhokla.\nalso sprinkle coriander leaves.\nand cut the dhokla to desired shape.\nfinally serve soft and spongy rava dhokla with green chutney."  
 },  
 {  
 "serves": "4 to 5",  
 "prepTime": "10 mins",  
 "cookTime": "45 mins",  
 "asset": "assets/countries/indian/chettinad.jpg",  
 "name": "Chettinad Fish Curry",  
 "items": "Fish – 1/2 kg ( We used Paarai fish) \nOil – 1 tblspn\nMustard Seeds "  
 "/ Kaduku – 1 tsp\nOnion – 1 small chopped finely\nTomato – 1 large cut into long slices\nCurry Leaves a sprig\nCoriander Leaves – 2 tblspn finely chopped\nTamarind Pulp – 3 tblspn\nFor Sauteing and Grinding:\nOil – 1 tblspn\nFennel Seeds / Saunf / Sombu – 1 tblspn\nOnions – 1 large sliced\nGarlic Paste – 1 tblspn\nTomato – 1 large chopped\nChilli Powder – 1 tblspn or to taste\nCoriander Powder – 2 tblspn \nTurmeric Powder / Manjal Podi – 1 tsp\nCoconut – 1/2 cup grated",  
 "process":  
 "Heat oil in a earthernware kadai. Add in fennel seeds and saute it for a min.\nAdd in onions and saute for a min. Add in garlic and saute for 30 sec. Add in tomatoes and saute well.\nAdd in spice powders and saute for a min. Add in coconut and mix well. Take that in blender and puree it well. \nNow in the same kadai, add in oil and heat it. Add in mustard seeds and crackle it.\nAdd in onions and curry leaves. Saute till is golden.\nAdd in coconut masala and mix well. Cook till oil separates. Add in the long sliced tomatoes and toss well.\nAdd in tamarind pulp and some water. Mix it well and bring it to boil. \nAdd in fish pieces and mix well. Season with salt.\nSimmer this for 8 to 10 mins till the fish is done. Add in coriander leaves and mix well.\nServe with rice.."  
 }  
];

Code for Indian Cuisines

# 6.SYSTEM TESTING

Tests:

The app development will consist of five parts. The first part will be devoted to data gathering and software requirements specification. Consequently, I will have a look at different mobile apps which target the same goal. They are plenty of Recipe Organizer apps. Each one has some various features. The second part will be dedicated to the design phase, including the app and the database. Also, in this phase, the software tools to be used will be specified. For example, the IDE, the database, the modelling language for the design, and finally the software testing tools. The third part will be the implementation phase, here, the design will be converted to code in order to develop the targeted app. The fourth step will be devoted to testing the app. In this phase, two testing methods will be used, namely: Black Box testing and White box testing. The last phase will be the deployment phase.

# 7.RESULTSCREEN SHORTS

Graphical user interface, application

Description automatically generated

Figure 16: App interface (i)

* Recipes are categorized in the form of cuisines (based on countries and origin).
* Fast food items are categorized into Burgers, Pizza, Noodles and Sausages.
* We have also collected the signature dishes of some of the world-famous chefs under Chefs Curated category.
* We have a category named ‘Meals’, where further divided into 6 categories:

- Lunch

- Breakfast

- Dessert

- Dinner

- Salad

- Soups

* We also have a special Category. Here, the user specific recipes are collected. For instance collection of vegan recipes are included.

A collage of food

Description automatically generated with medium confidence

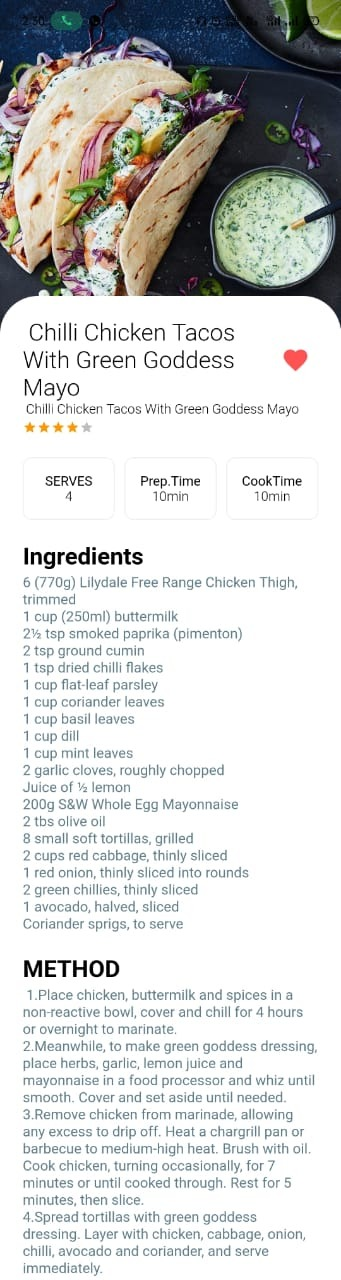


Figure 18: App interface (iii)

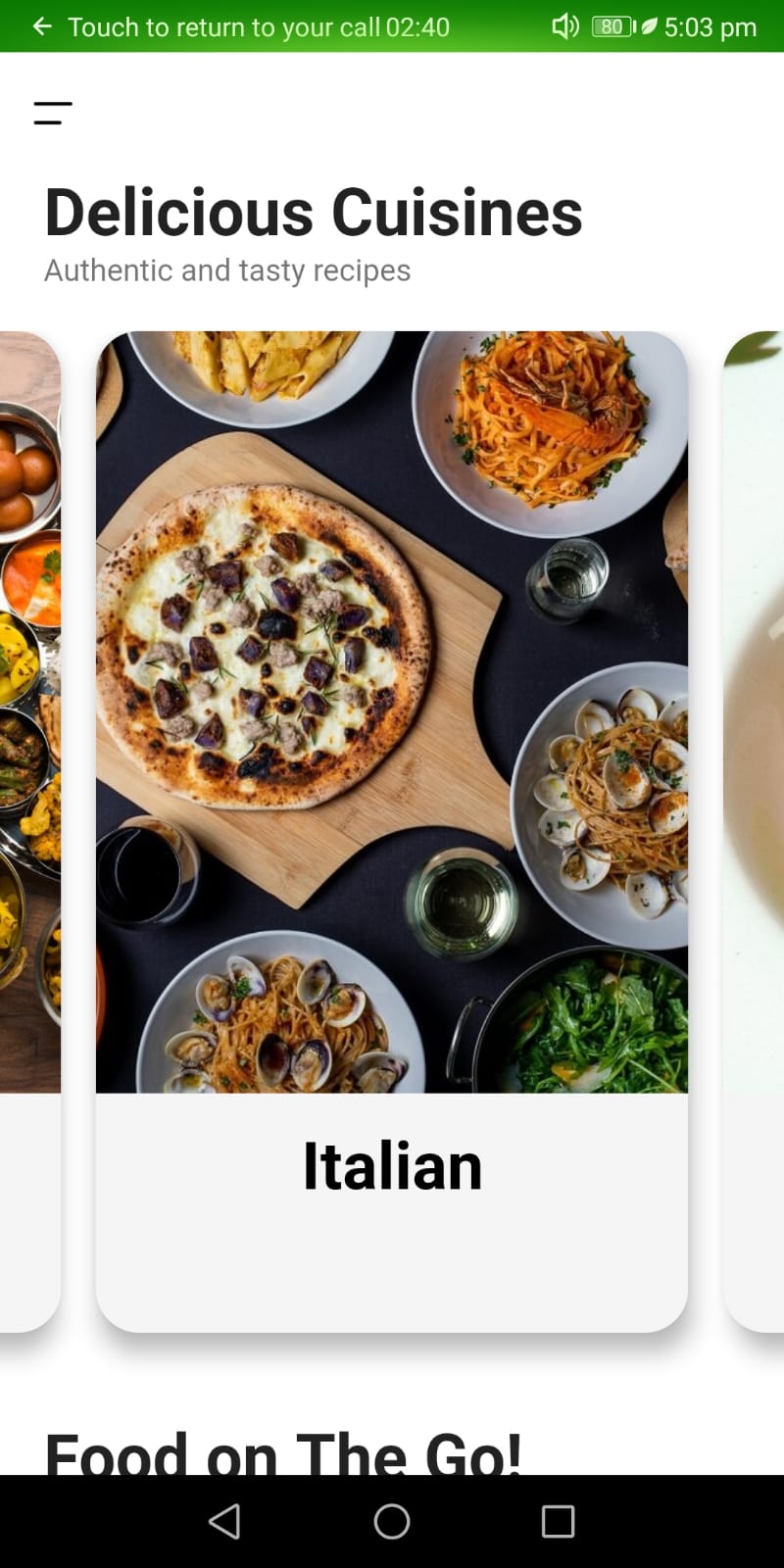


Figure 19: App interface (iv)

A menu for a restaurant

Description automatically generated with low confidence

Figure 20: App interface (v)

A picture containing text, food, snack food

Description automatically generated

Figure 21: App interface (vi)

Figure 22: App interface (vii)Text

Description automatically generated

# 8. CONCLUSION

The Recetté application meets with the enterprise class application principles. It is designed to be performing, scalable, extensible and highly available. Given that it may be improved in many ways, the application is also easily maintainable.

This document summarizes the work that has been done since the beginning of this semester. Indeed, it starts by giving an overview about the project specification and requirements.

The document also states the methodology followed and which consists of 5 main parts:

The first part will be devoted to data gathering and software requirements specification. Consequently, I will have a look at different mobile apps which target the same goal. They are plenty of Cookbook apps. Each one has some various features.

The second part will be dedicated to the design phase, including the app and the database. Also, in this phase, the software tools to be used will be specified. For example, the IDE, the modelling language for the design, and finally the software testing tools.

The third part will be the implementation phase, here, the design will be converted to code in order to develop the targeted app.The last phase will be the deployment phase.

# 9.BIBLIOGRAPHY

1. [flutter\_svg | Flutter Package (pub.dev)](https://pub.dev/packages/flutter_svg)
2. [Recipes | Recipes for Breakfast, Lunch & Dinner (delicious.com.au)](https://www.delicious.com.au/recipes)
3. [Meal and Recipe Finder: cooking instructions to bake healthy food (allthemeals.com)](https://allthemeals.com/)
4. [Cookbook - Flutter](https://flutter.dev/docs/cookbook)