

# **Chef Hour: An App for Complete Food related Topics**

A Project Report

submitted as a part of the internal evaluation process for the course

**15CSE375 Android Application Development**

By

**GROUP VI**

**AKASH RAVI (AM.EN.U4CSE17004)**

**RAMKISHOR PRABHU (AM.EN.U4CSE17058)**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITAPURI CAMPUS (INDIA)**

**MAY 2020**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITAPURI CAMPUS**



**BONAFIDE CERTIFICATE**

This is to certify that the project report entitled **Chef Hour: An App for Complete Food related Topics** submitted by **Akash Ravi (AM.EN.U4CSE17004)**, **Ramkishor Prabhu R(AM.EN.U4CSE17058)** as a part of the internal evaluation process for the course **15CSE375 Android Application Development** is a bonafide record of the work carried out by them under the guidance supervision of me at the Department of Mechanical Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Amritapuri Campus.

**Mr. Rahul Varma**

Faculty Associate,

Department of Computer Science And Engineering,

Amrita School of Engineering,

Amritapuri

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITAPURI CAMPUS**

**DECLARATION**

We, **Akash Ravi (AM.EN.U4CSE17004)**, **Ramkishor Prabhu R(AM.EN.U4CSE17058)** hereby declare that this project report entitled **Chef Hour: An App for Complete Food related Topics** is the record of the original work done by us under the guidance of **Mr. Rahul Varma**, Faculty Associate, Department of Computer Science And Engineering, Amrita School of Engineering, Amritapuri.

Place: Amritapuri

Date: 30-04-2020

**AKASH RAVI (AM.EN.U4CSE17004)**

**RAMKISHOR PRABHU (AM.EN.U4CSE17058)**

## **ACKNOWLEDGEMENT**

We have given our full efforts in the completion of this project. However, it would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them. We are highly indebted to **Mr. Rahul Varma** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project. Both of us have put our efforts equally in the successful completion of this project. We are also thankful to the Institution and CSE Department for providing us with the opportunity to work on a project to enhance our skills in the field of Android Application Development.

**AKASH RAVI** (AM.EN.U4CSE17004)

**RAMKISHOR PRABHU** (AM.EN.U4CSE17058)

# CONTENTS

<i><b>SL No.</b></i>	<i><b>Title</b></i>	<i><b>Page No.</b></i>
<b>1</b>	<b>Executive Summary</b>	<b>06</b>
<b>2</b>	<b>System Requirements</b>	<b>07 - 09</b>
<b>3</b>	<b>System Architecture</b>	<b>10 - 11</b>
<b>4</b>	<b>User Interface</b>	<b>12 - 14</b>
<b>5</b>	<b>Android Components</b>	<b>15 - 16</b>
<b>6</b>	<b>Data Storage</b>	<b>17 - 19</b>
<b>7</b>	<b>Design Blueprint</b>	<b>20 -113</b>
<b>8</b>	<b>Result, Storyboard and future additions</b>	<b>114 - 124</b>
<b>9</b>	<b>References</b>	<b>125</b>

# **UNIT I**

## **EXECUTIVE SUMMARY**

### ***ABSTRACT***

ChefHour application is a very useful app for people who love to cook and try out new recipes. It enables the user with the luxury of choosing recipes from a wide variety of categories just a click away. Recipes are from everyday cooks and chefs that have perfected the recipe over time. Recipes range from the decadent to the simple for whatever mood user is in. Category based search is also provided to users. This application is a time saver providing recipes in a few clicks.

The application displays categorised food recipes and the user will be able to choose according to their wish. The application uses API to get the list of dishes and recipes and it also redirects to a youtube link if the user wishes to watch a video demonstration of the recipe in execution.

# UNIT II

## SYSTEM REQUIREMENTS

**Details of the System used in developing this Application:**

**Operating System:** *Windows 10 Home Single Language*

**Version:** *1903*

**OS Build:** *18362.778*

**Android Studio Specifications:**

*Android Studio 3.5.2*

*Build #AI-191.8026.42.35.5977832, built on October 31, 2019*

*JRE: 1.8.0\_202-release-1483-b03 amd64*

*JVM: OpenJDK 64-Bit Server VM by JetBrains s.r.o*

*Windows 10 10.0*

*Gradle: 3.3.2*

### ***FUNCTIONS***

**Users of QuickRecipes application have following options:**

- Users can get recipes from the cloud.
- Users can view selected recipes from a list of recipes.
- Users have the option to save recipes as favorites.

- Users can view the video of the corresponding recipes on YouTube.
- Each user can have their own accounts for saving their favourites
- The recipes are classified in categories, which allows the user to find the recipe they need faster.
- Users can navigate to the source web from which the recipe is taken.

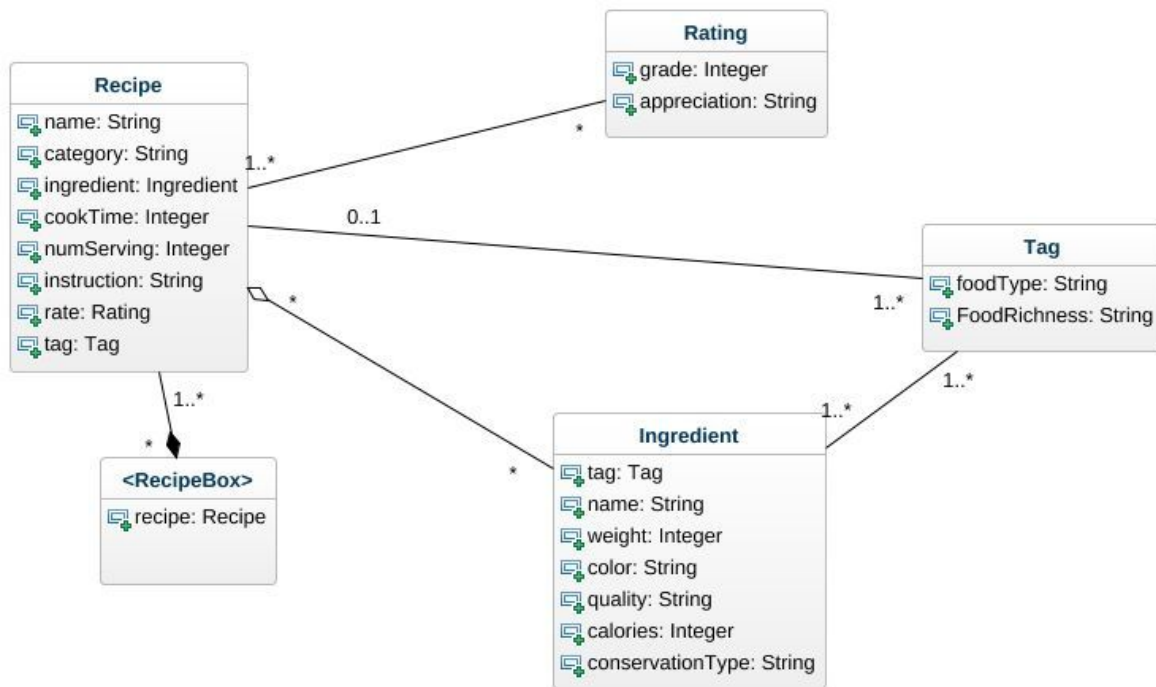
### **Recipes can be added to favourites.**

- Doing this will help in availing the recipe when offline.

### **Different Functions Used in the Application:**

- `showLoading()`: To show the loading animation
- `hideLoading()`: To hide the loading animation once the application is loaded
- `setMeal()` : To set the contents of the Meal.
- `setCategory`: To set the category of the meals.
- `onErrorLoading()`: To show if an error is encountered.
- `getcategories`: To show the categories.
- `getMeals()`: to show the Meals.
- `initIntent()`: To initialise the Intent.
- `initActionBar()`: To initialise the action bar.
- `LoginFunction()`: A function to control Logins.
- `CheckFinalResult()`: To see if an Email is already registered.
- `InsertDataIntoSQLiteTable()`: To insert data.

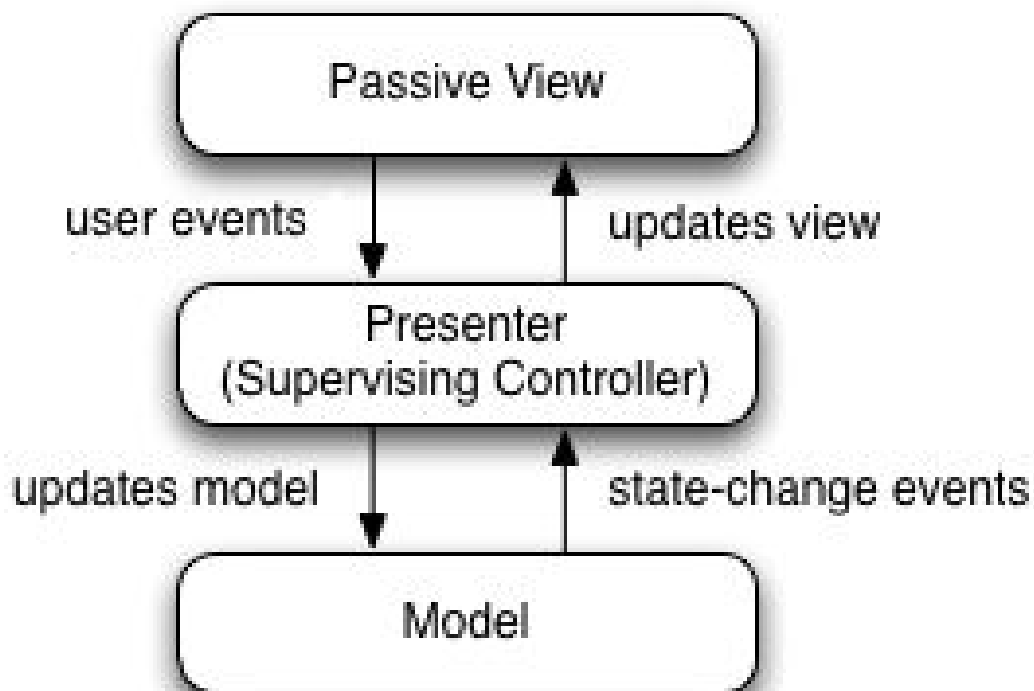




# UNIT III

## SYSTEM ARCHITECTURE

The application uses MVP architecture for the smooth execution of the required features. Model-View-Presenter is a derivation of Model-View-Controller (MVC) architectural pattern and is used mostly for building user interfaces. In MVP the presenter assumes the functionality of the “middle-man” and all presentation logic is pushed to the presenter.

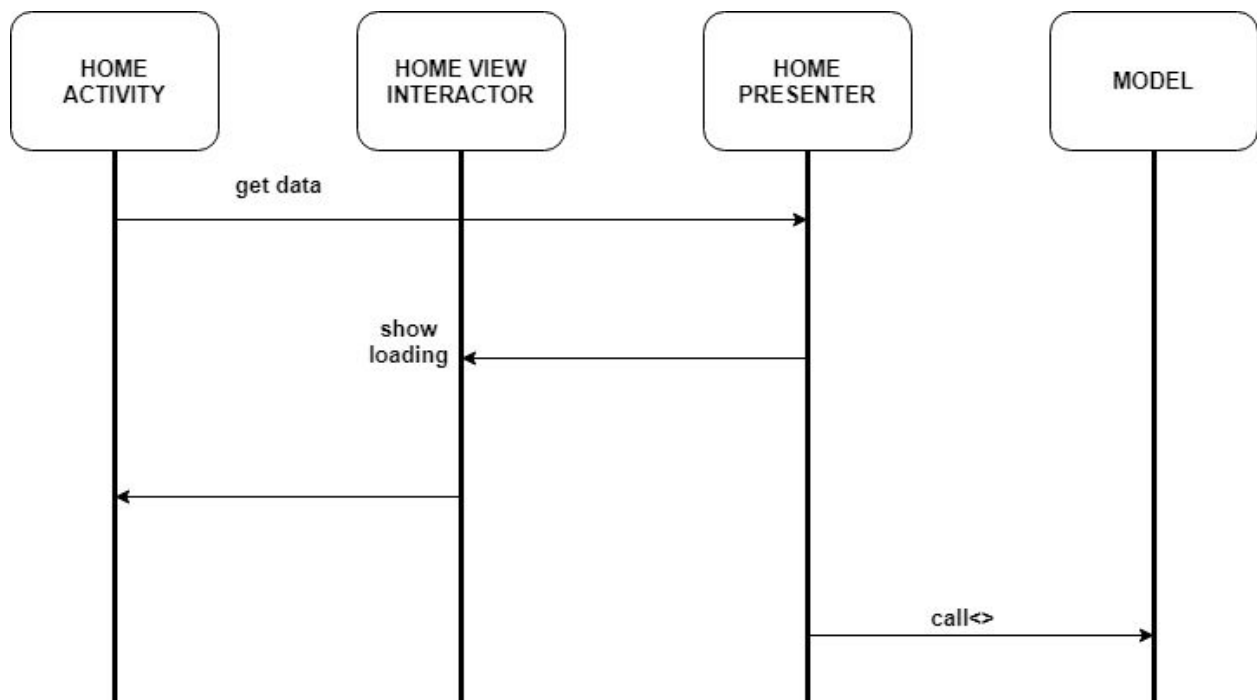


**Model:** It consist of Data and the API

**Presenter:** The presenter will give what is requested by the activity to the model and then return the data

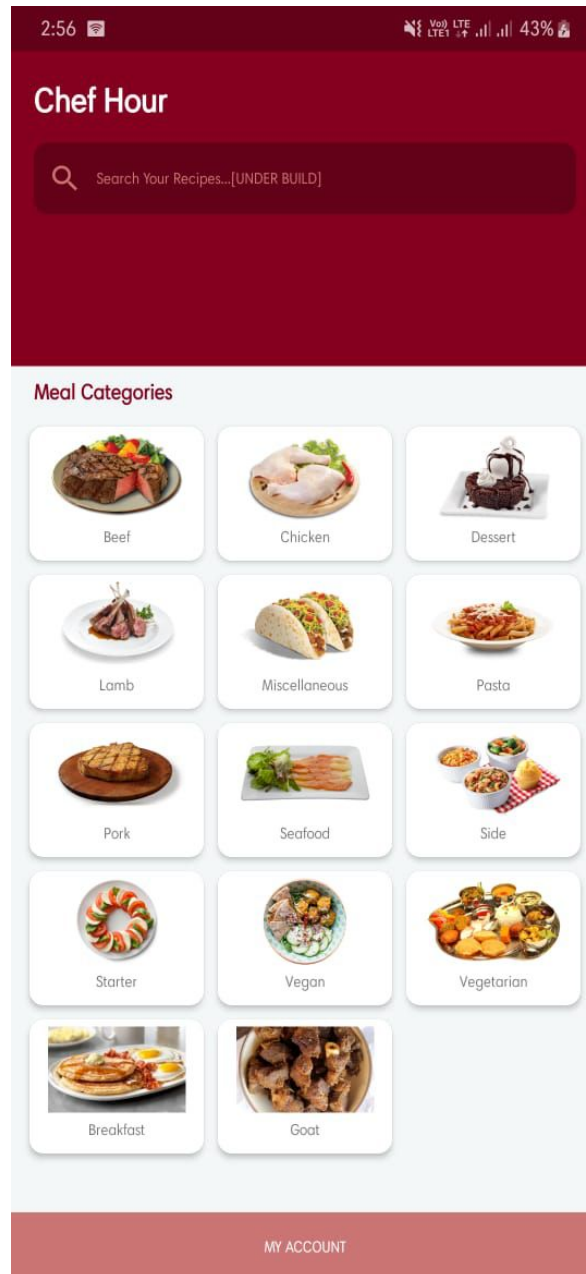
**View:** It consist of activities, fragments and various other UI components

The following diagram represents the execution of the application following the above mentioned architecture



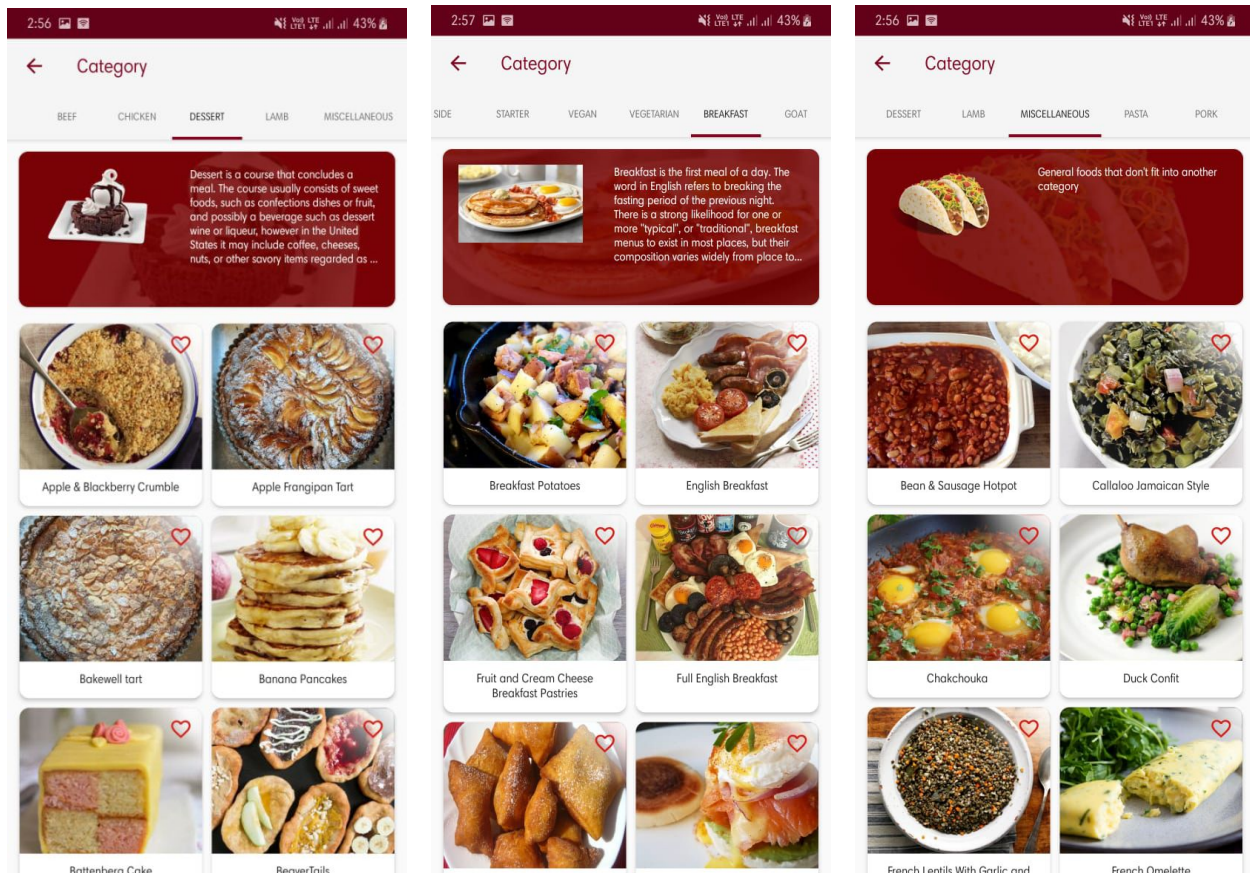
# UNIT IV

## USER INTERFACE

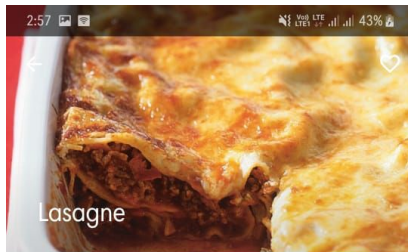


### Home Page

## Dishes sorted Category-wise



## Recipe of the selected dish



### Lasagne

Category

Pasta

Country

Italian

INSTRUCTIONS

Heat the oil in a large saucepan. Use kitchen scissors to snip the bacon into small pieces, or use a sharp knife to chop it on a chopping board. Add the bacon to the pan and cook for just a few mins until starting to turn golden. Add the onion, celery and carrot, and cook over a medium heat for 5 mins, stirring occasionally, until softened. Add the garlic and cook for 1 min, then tip in the mince and cook, stirring and breaking it up with a wooden spoon, for about 6 mins until browned all over.

Stir in the tomato purée and cook for 1 min, mixing in well with the beef and vegetables. Tip in the chopped tomatoes. Fill each can half full with water to rinse out any tomatoes left in the can, and add to the pan. Add the honey and season to taste. Simmer for 20 mins.

Heat oven to 200C/180C fan/gas 6. To assemble the lasagne, ladle a little of the ragu sauce into the bottom of the roasting tin or casserole dish, spreading the sauce all over the base. Place 2 sheets of lasagne on top of the sauce overlapping to make it fit, then repeat with more sauce and another layer of pasta. Repeat with a further 2 layers of sauce and pasta, finishing with a layer of pasta.

Put the crème fraîche in a bowl and mix with 2 tbsp water to loosen it and make a smooth pourable sauce. Pour this over the top of the pasta, then top with the mozzarella. Sprinkle Parmesan over the top and bake for 25–30 mins until golden and bubbling. Serve scattered with basil, if you like.

INGREDIENTS

- Olive Oil : 1 tblsp
- Bacon : 2
- Onion : 1 finely chopped
- Celery : 1 Stick



### Lasagne

Category

Pasta

Country

Italian

INSTRUCTIONS

Heat the oil in a large saucepan. Use kitchen scissors to snip the bacon into small pieces, or use a sharp knife to chop it on a chopping board. Add the bacon to the pan and cook for just a few mins until starting to turn golden. Add the onion, celery and carrot, and cook over a medium heat for 5 mins, stirring occasionally, until softened. Add the garlic and cook for 1 min, then tip in the mince and cook, stirring and breaking it up with a wooden spoon, for about 6 mins until browned all over.

Stir in the tomato purée and cook for 1 min, mixing in well with the beef and vegetables. Tip in the chopped tomatoes. Fill each can half full with water to rinse out any tomatoes left in the can, and add to the pan. Add the honey and season to taste. Simmer for 20 mins.

Heat oven to 200C/180C fan/gas 6. To assemble the lasagne, ladle a little of the ragu sauce into the bottom of the roasting tin or casserole dish, spreading the sauce all over the base. Place 2 sheets of lasagne on top of the sauce overlapping to make it fit, then repeat with more sauce and another layer of pasta. Repeat with a further 2 layers of sauce and pasta, finishing with a layer of pasta.

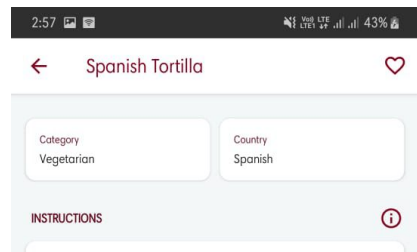
Put the crème fraîche in a bowl and mix with 2 tbsp water to loosen it and make a smooth pourable sauce. Pour this over the top of the pasta, then top with the mozzarella. Sprinkle Parmesan over the top and bake for 25–30 mins until golden and bubbling. Serve scattered with basil, if you like.

INGREDIENTS

- Olive Oil : 1 tblsp
- Bacon : 2
- Onion : 1 finely chopped
- Celery : 1 Stick
- Carrots : 1 medium
- Garlic : 2 cloves chopped
- Mince Beef : 500g
- Tomato Puree : 1 tbs
- Chopped Tomatoes : 800g
- Honey : 1 tblsp
- Lasagne Sheets : 500g
- Creme Fraiche : 400ml
- Mozzarella Balls : 125g
- Parmesan Cheese : 50g
- Basil Leaves : Topping

▶ YOUTUBE

↪ SOURCE



### Spanish Tortilla

Category

Vegetarian

Country

Spanish

INSTRUCTIONS

Put a large non-stick frying pan on a low heat. Cook the onion slowly in the oil and butter until soft but not brown – this should take about 15 mins. Add the potatoes, cover the pan and cook for a further 15–20 mins, stirring occasionally to make sure they fry evenly.

When the potatoes are soft and the onion is shiny, crush 2 garlic cloves and stir in, followed by the beaten eggs.

Put the lid back on the pan and leave the tortilla to cook gently. After 20 mins, the edges and base should be golden, the top set but the middle still a little wobbly. To turn it over, slide it onto a plate and put another plate on top, turn the whole thing over and slide it back into the pan to finish cooking. Once cooked, transfer to a plate and serve the tortilla warm or cold, scattered with the chopped parsley.

To accompany, take slices of warmed baguette, stab all over with a fork and rub with the remaining garlic, pile on grated tomatoes and season with sea salt and a drizzle of olive oil.

INGREDIENTS

- Onion : 1 sliced
- Olive Oil : 4 tblsp
- Butter : 25g
- Potatoes : 400g
- Garlic : 6 cloves
- Eggs : 8
- Parsley : Handful
- Baguette : 1
- Vine Tomatoes : 4
- Olive Oil : drizzle

▶ YOUTUBE

↪ SOURCE

# UNIT V

## ANDROID COMPONENTS

### 1. API

This application makes use of Meals API provided by <https://www.themealdb.com/>. TheMealDB is a free data source API for recipes online, for helping the developers.

### 2. Butterknife

Butterknife is a light weight library to inject views into Android components. It makes use of annotation processing. The *@BindView* annotation allow to inject views and performs the cast to the correct type for you. The *@@OnClick(R.id.yourid)* annotation allows you to add OnClickListener to a view. You can optionally define the method parameter of the view in case you want it injected. Butterknife includes also *findById* methods which simplify code that still has to find views on a View, Activity, or Dialog.

### 3. Retrofit2

Retrofit is a REST Client for Java and Android. It makes it relatively easy to retrieve and upload JSON (or other structured data) via a REST based

webservice. In Retrofit you configure which converter is used for the data serialization. Typically for JSON you use GSON, but you can add custom converters to process XML or other protocols. Retrofit uses the OkHttp library for HTTP requests.

#### **4. Shimmer FB**

Shimmer is an Android library that provides an easy way to add a shimmer effect to any view in your Android app. Shimmer for Android is implemented as a layout, which means that you can simply nest any view inside a *ShimmerFrameLayout* tag, and call to start the animation from your code. You can control the look and pace of the effect using a number of custom attributes on the *ShimmerFrameLayout* tag. Alternatively, you can set these values on the layout object itself



# UNIT VI

## DATA STORAGE

The Application uses SQLite as the database engine for the processing of data used in the application. SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

The Application basically uses the SQLite for managing the User accounts. The User account information is maintained in a table, also known as a relation. The table consists of four columns; which includes UserID, Email, Username, and password.

**UserID:** *[Integer]* A primary key non null attribute, which uses an auto-increment function to generally keep the UserID updated uniquely.

**EmailID:** *[Varchar]* This is also a primary attribute which specifies the User's Email ID. This Email is inputted only after confirming that no users exist with the similar Email ID.

**UserName:** *[Varchar]* This is not a unique attribute unlike the usernames you find. It specifies the Name of the Users.

**Password:** *[Varchar]* This is a privately secured varchar attribute stored in the database.

*Following is the structure of the database:*

User ID	Email	Username	Password
01	arun10@gmail.com	Arun S	*****
02	JVG12@xyz.com	Joseph VG	*****
03	natLyon@lol.com	Nathon Lyon	*****
04	perry12@gmail.com	Elyse Perry	*****

Meal information are taken from an API, therefore are not specifically stored in a database, but the structure of the database is as follows:

RecipeID	Recipe Name	Ingredients	Directions	Country	YouTube Link	Source
R01	Aalo Tikki	1 1/2 cups self-rising flour, plus more for kneading, Cooking spray, Greek Yogurt, toppings	If you want to make a good pizza that's fast and easy, this is it. Cooking at a higher temperature (500 degrees F (260 degrees C)) is the key to making this the perfect pizza. It doesn't get any	Indian	<a href="https://www.youtube.com/watch?v=RNZ1ouYfjZk">https://www.youtube.com/watch?v=RNZ1ouYfjZk</a>	<a href="https://www.indianhealthyrecipes.com/aloo-tikki-recipe/">https://www.indianhealthyrecipes.com/aloo-tikki-recipe/</a>

			easier than this folks. Brush dough with olive oil and add your favorite toppings. Enjoy!"			
R02	Oreo Cupcake	Oreo biscuits, Dairy Milk, 1/2 Cup Milk	Grind the oreo, Add milk and again grind it; Then melt the Diary Milk; Add The grinded oreo as well as dairy milk to a cupcake pan and Bake it.	Latin America	<a href="https://www.youtube.com/watch?v=yYMbhdPK-9M">https://www.youtube.com/watch?v=yYMbhdPK-9M</a>	<a href="https://www.cookwithmanali.com/oreo-cupcakes-cookies-cream-frosting/">https://www.cookwithmanali.com/oreo-cupcakes-cookies-cream-frosting/</a>

**NB:** *The Application only allows you to view the above datasets. But doesn't give you privilege in updating , deleting or including new additions to the dishes.*

# UNIT VII

## DESIGN BLUEPRINT

### CODE

#### AndroidManifest.XML

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.akash.chefhour">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".view.home.HomeActivity"
            android:theme="@style/AppTheme">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <action android:name="android.intent.action.VIEW" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".view.category.CategoryActivity"
            android:theme="@style/AppTheme"
            android:screenOrientation="portrait"
            android:label="Category" />
        <activity android:name=".view.detail.DetailActivity"
            android:theme="@style/AppTheme.NoActionBar" />
    </application>
</manifest>
```

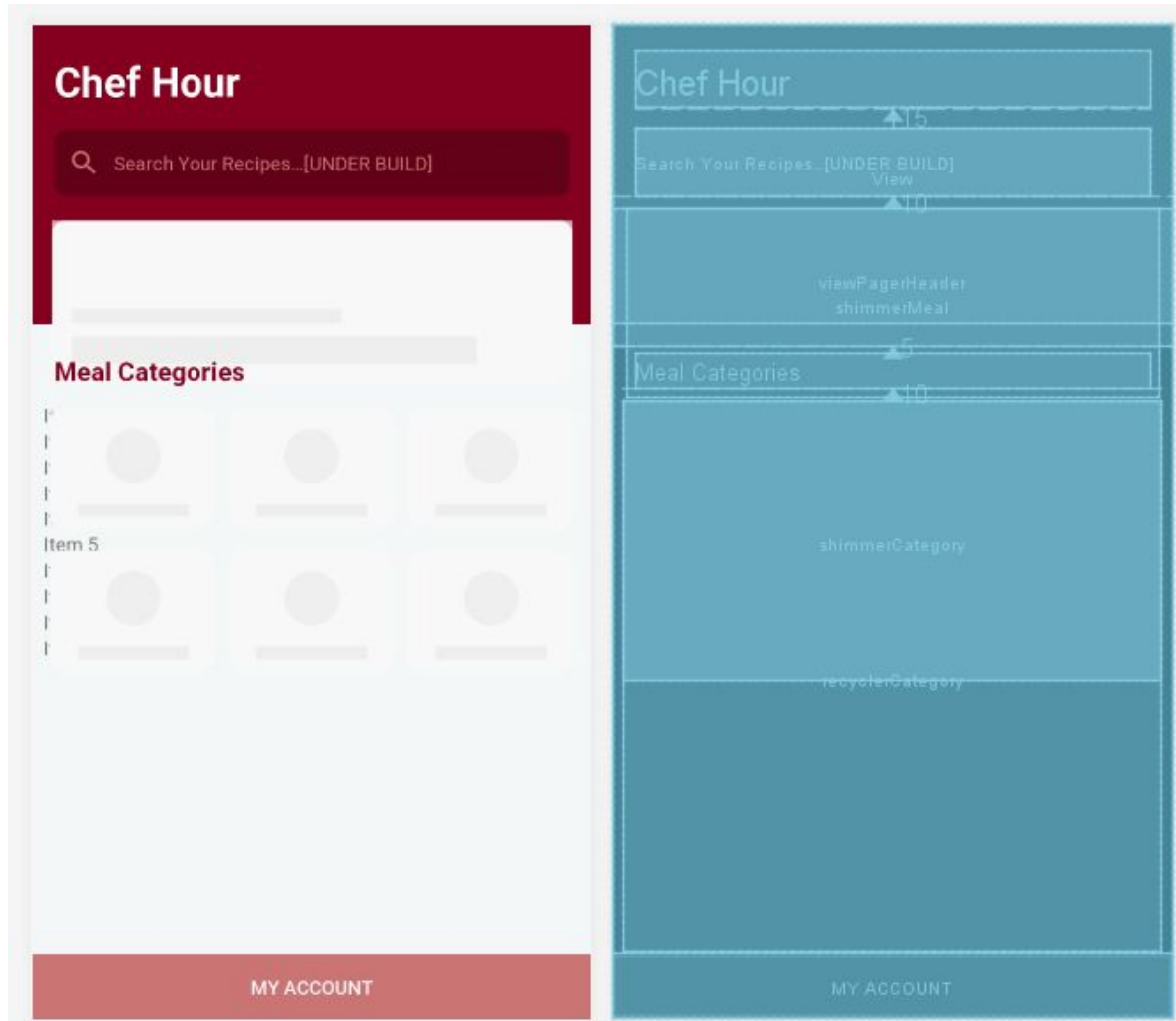
```
        <activity android:name=".view.Account" />
        <activity android:name=".view.DashboardActivity" />
        <activity android:name=".view.RegisterActivity" />
    </application>

</manifest>
```

[P.T.O.]

## LAYOUTS

### Activity\_home.xml



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context=".view.home.HomeActivity">

<android.support.v4.widget.NestedScrollView
app:layout_behavior="@string/appbar_scrolling_view_behavior"
```

```

android:fillViewport="true"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/colorBackground">
<RelativeLayout
    android:id="@+id/headerView"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="300dp">
    <View
        android:layout_width="match_parent"
        android:layout_height="220dp"
        android:background="@color/colorPrimary" />

    <TextView
        android:id="@+id/title"
        android:textColor="@color/colorWhite"
        android:text="@string/app_name"
        android:textStyle="bold"
        android:textSize="30sp"
        android:layout_marginLeft="16dp"
        android:layout_marginRight="16dp"
        android:layout_marginTop="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</android.support.v7.widget.CardView>
<android.support.v7.widget.CardView
    android:id="@+id/cardSearch"
    android:layout_below="@id/title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="15dp"
    app:cardBackgroundColor="@color/colorPrimaryDark"
    app:cardCornerRadius="10dp"
    app:cardElevation="0dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:padding="10dp"
        android:text="@string/search_your_recipes"
        android:textColor="@color/colorPrimaryLight"
        android:gravity="center_vertical"
        android:drawableLeft="@drawable/ic_search_primary_light"
        android:drawableStart="@drawable/ic_search_primary_light"
        android:drawablePadding="10dp" />
</android.support.v7.widget.CardView>

```

```

<android.support.v4.view.ViewPager
    android:id="@+id/viewPagerHeader"
    android:foregroundGravity="center"
    android:overScrollMode="never"
    android:clipToPadding="false"
    android:layout_below="@id/cardSearch"
    android:layout_marginTop="10dp"
    android:layout_width="match_parent"
    android:layout_height="100dp" />

<include android:id="@+id/shimmerMeal"
    layout="@layout/item_view_pager_header_shimmer" />

<TextView
    android:id="@+id/titleCategory"
    android:text="@string/meal_categories"
    android:textSize="19sp"
    android:textColor="@color/colorPrimary"
    android:textStyle="bold"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="5dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/viewPagerHeader"/>

<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerCategory"
    android:scrollbars="none"
    android:layout_marginTop="10dp"
    android:clipToPadding="false"
    android:layout_below="@id/titleCategory"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="50dp" />

<include android:id="@+id/shimmerCategory"
    layout="@layout/item_recycler_category_shimmer" />

<Button
    android:id="@+id/account"
    android:padding="10dp"
    android:gravity="center"
    android:textColor="@color/colorWhite"
    android:text="My Account"
    android:background="@color/colorPrimaryLight"
    android:layout_alignParentBottom="true"
    android:layout_width="match_parent"

```

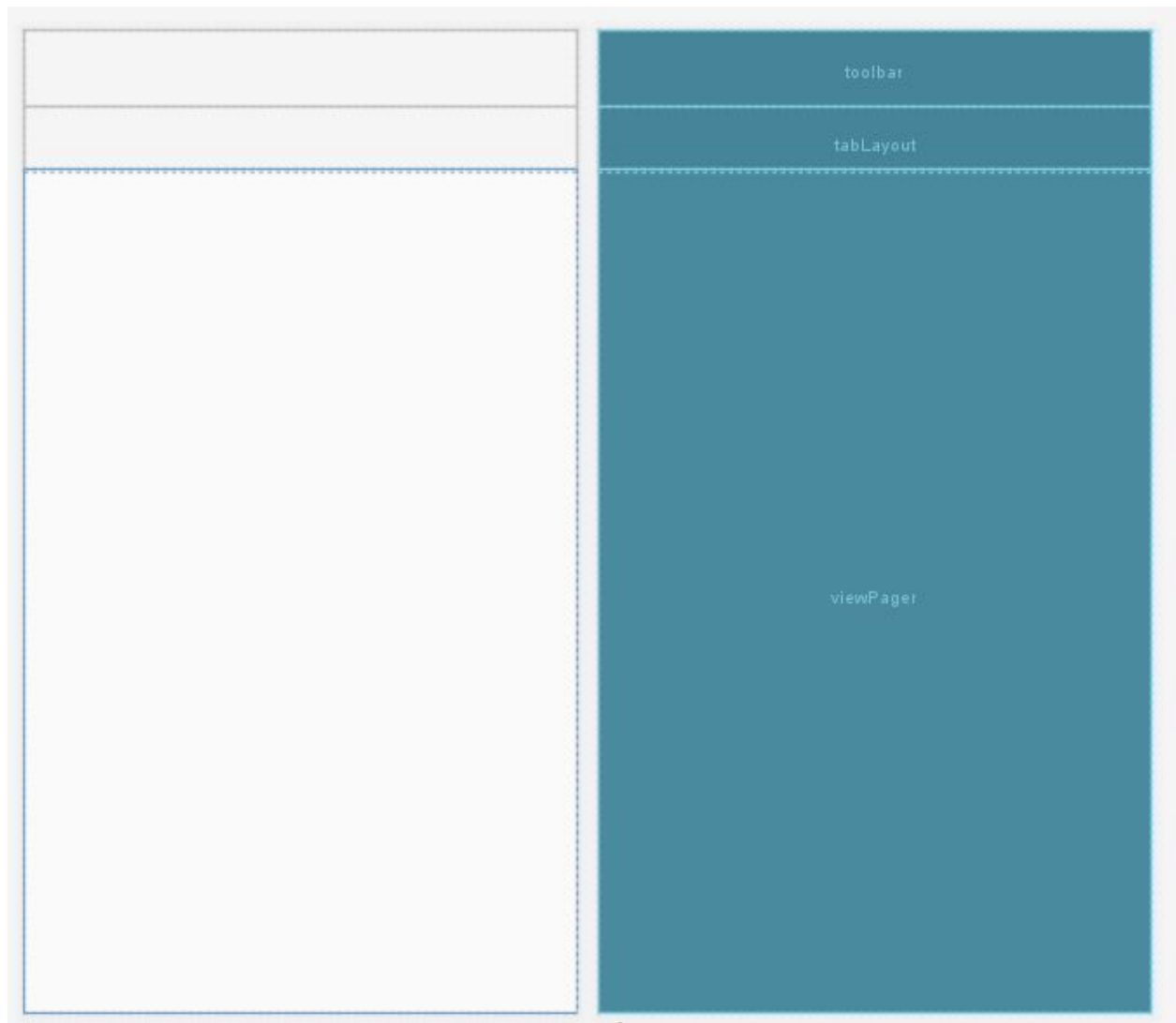


```
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"/>

    </RelativeLayout>
</android.support.v4.widget.NestedScrollView>

</android.support.design.widget.CoordinatorLayout>
```

## Activity\_Category.xml



```

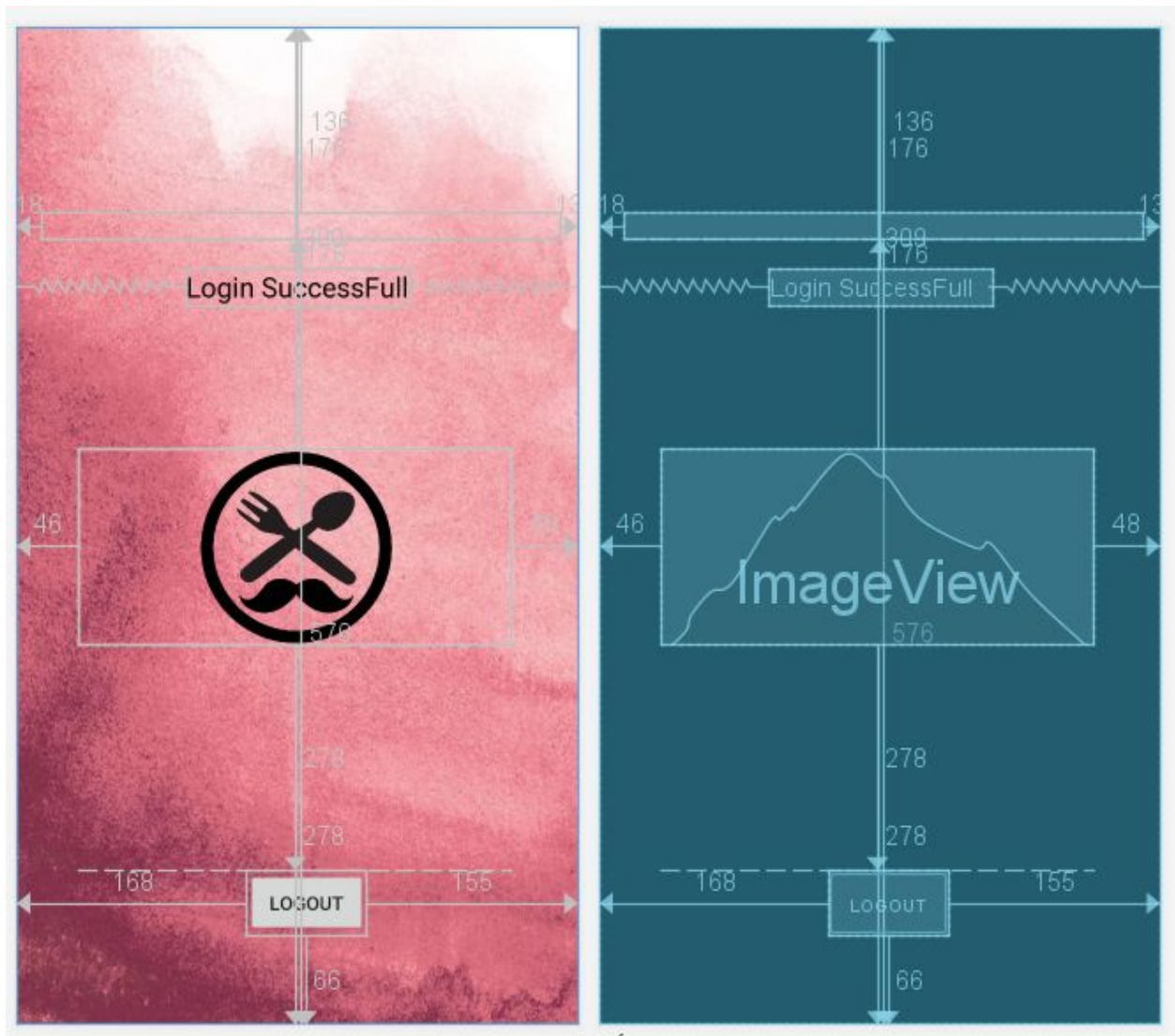
<?xml version="1.0" encoding="utf-8"?>

<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context=".view.category.CategoryActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppBarLayoutTheme">
        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>
        <android.support.design.widget.TabLayout
            android:id="@+id/tabLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:tabIndicatorHeight="3dp"
            app:tabIndicatorColor="@color/colorPrimary"
            app:tabMode="scrollable"
            app:tabContentStart="85dp"/>
    </android.support.design.widget.AppBarLayout>
    <android.support.v4.view.ViewPager
        android:id="@+id/viewPager"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</android.support.design.widget.CoordinatorLayout>

```

## Activity\_dashboard.xml



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_dashboard"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bk"

    tools:context="com.akash.chefhour.view.DashboardActivity">
```

```
<TextView
    android:text="Login SuccessFull"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="20dp"
    android:textColor="#000"
    android:id="@+id/textView1"
    android:layout_below="@+id/textView2"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="176dp" />
```

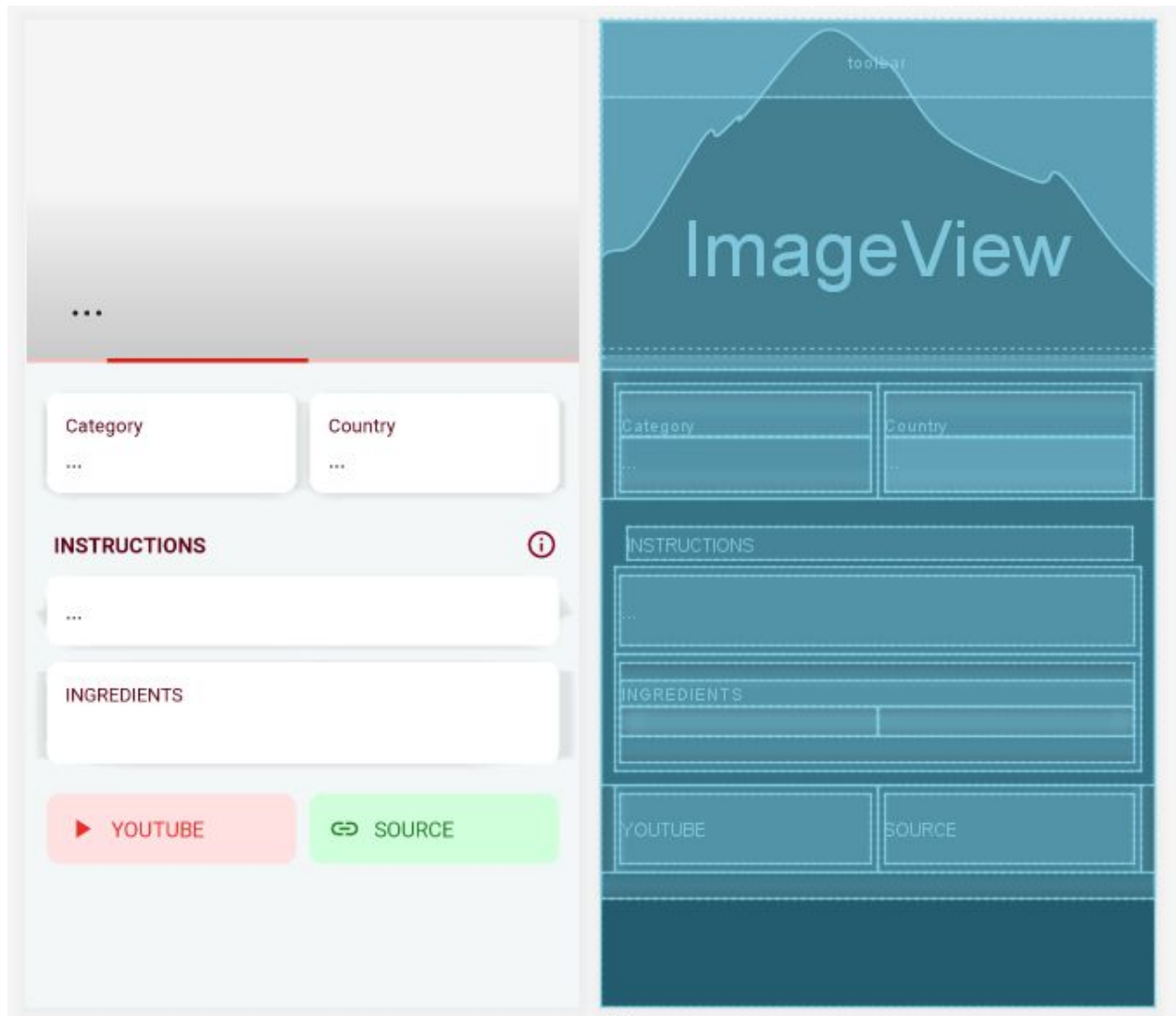
```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="168dp"
    android:layout_marginLeft="168dp"
    android:layout_marginTop="347dp"
    android:layout_marginEnd="155dp"
    android:layout_marginRight="155dp"
    android:layout_marginBottom="66dp"
    android:text="LOGOUT" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="380dp"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="18dp"
    android:layout_marginLeft="18dp"
    android:layout_marginTop="136dp"
    android:layout_marginEnd="13dp"
    android:layout_marginRight="13dp"
    android:layout_marginBottom="576dp"
    android:gravity="center"
    android:textColor="#000"
    android:textSize="16dp" />
```

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="317dp"
    android:layout_height="144dp"
    android:layout_above="@+id/button1"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="46dp"
    android:layout_marginLeft="46dp"
    android:layout_marginTop="309dp"
    android:layout_marginEnd="48dp"
    android:layout_marginRight="48dp"
    android:layout_marginBottom="278dp"
    app:srcCompat="@drawable/logo" />

</RelativeLayout>
```

## Activity\_detail.xml



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.design.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
android:background="@color/colorBackground"
tools:context=".view.detail.DetailActivity">
```

```

<android.support.design.widget.AppBarLayout
    android:id="@+id/appbar"
    android:layout_width="match_parent"
    android:layout_height="250dp"
    android:fitsSystemWindows="true"
    android:theme="@style/AppBarLayoutTheme">

    <android.support.design.widget.CollapsingToolbarLayout
        android:id="@+id/collapsing_toolbar"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_scrollFlags="scroll|exitUntilCollapsed"
        app:title="@string/dot_loading">
        <ImageView
            android:id="@+id/mealThumb"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            android:scaleType="centerCrop"
            app:layout_collapseMode="parallax"/>
        <ImageView
            android:src="@drawable/shadow_bottom_to_top"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:layout_collapseMode="pin"
            app:popupTheme="@style/ThemeOverlay.AppCompat.Light"/>

    </android.support.design.widget.CollapsingToolbarLayout>
</android.support.design.widget.AppBarLayout>

<android.support.v4.widget.NestedScrollView
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fitsSystemWindows="true">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <ProgressBar
            android:id="@+id/progressBar"
            style="@style/Base.Widget.AppCompat.ProgressBar.Horizontal"
            android:layout_marginTop="-7dp"
            android:indeterminate="true"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="10dp">
    <android.support.v7.widget.CardView
        app:cardUseCompatPadding="true"
        app:cardCornerRadius="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">
            <TextView
                android:id="@+id/textCategory"
                android:text="@string/category"
                android:paddingLeft="14dp"
                android:paddingRight="14dp"
                android:paddingTop="14dp"
                android:textColor="@color/colorPrimaryDark"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
            <TextView
                android:id="@+id/category"
                android:text="@string/dot_loading"
                android:paddingLeft="14dp"
                android:paddingRight="14dp"
                android:paddingTop="5dp"
                android:paddingBottom="14dp"
                android:textSize="16sp"
                android:textColor="@color/colorPrimaryText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
            </LinearLayout>
        </android.support.v7.widget.CardView>
    <android.support.v7.widget.CardView
        app:cardUseCompatPadding="true"
        app:cardCornerRadius="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"

```



```

        android:orientation="vertical">
        <TextView
            android:id="@+id/Country"
            android:text="@string/country"
            android:paddingLeft="14dp"
            android:paddingRight="14dp"
            android:paddingTop="14dp"
            android:textColor="@color/colorPrimaryDark"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <TextView
            android:id="@+id/country"
            android:text="@string/dot_loading"
            android:paddingLeft="14dp"
            android:paddingRight="14dp"
            android:paddingTop="5dp"
            android:paddingBottom="14dp"
            android:textSize="16sp"
            android:textColor="@color/colorPrimaryText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
    </LinearLayout>
</android.support.v7.widget.CardView>
</LinearLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/instructions"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="20dp"
    android:layout_marginBottom="5dp"
    android:textSize="16sp"
    android:textStyle="bold"
    android:textColor="@color/colorPrimaryDark"
    android:gravity="center_vertical"
    android:drawablePadding="10dp"
    android:drawableEnd="@drawable/ic_info"
    android:drawableRight="@drawable/ic_info" />

<android.support.v7.widget.CardView
    app:cardUseCompatPadding="true"
    app:cardCornerRadius="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp">
    <TextView
        android:id="@+id/instructions"

```

```

        android:text="@string/dot_loading"
        android:paddingLeft="14dp"
        android:paddingRight="14dp"
        android:paddingTop="15dp"
        android:paddingBottom="15dp"
        android:textSize="16sp"
        android:textColor="@color/colorPrimaryText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
    app:cardUseCompatPadding="true"
    app:cardCornerRadius="10dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:paddingLeft="14dp"
            android:paddingRight="14dp"
            android:layout_marginTop="14dp"
            android:text="@string/Ingredients"
            android:textColor="@color/colorPrimaryDark"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />
        <LinearLayout
            android:layout_marginBottom="20dp"
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:layout_weight="1">
                <TextView
                    android:id="@+id/ingredient"
                    android:paddingLeft="14dp"
                    android:paddingRight="14dp"
                    android:textSize="16sp"
                    android:textColor="@color/colorPrimaryText"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content" />
            </LinearLayout>

```

```

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_weight="1">
            <TextView
                android:id="@+id/measure"
                android:paddingLeft="14dp"
                android:paddingRight="14dp"
                android:textSize="16sp"
                android:textColor="@color/colorPrimaryText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content" />
            </LinearLayout>
        </LinearLayout>
    </LinearLayout>
</android.support.v7.widget.CardView>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="20dp">
    <android.support.v7.widget.CardView
        android:layout_weight="1"
        app:cardUseCompatPadding="true"
        app:cardBackgroundColor="#FFE1E1"
        app:cardElevation="0dp"
        app:cardCornerRadius="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp">
        <TextView
            android:background="?attr/selectableItemBackground"
            android:clickable="true"
            android:focusable="true"
            android:id="@+id/youtube"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/youtube"
            android:padding="14dp"
            android:textSize="16sp"
            android:textColor="#EC2D2D"
            android:gravity="center_vertical"
            android:drawablePadding="10dp"
            android:drawableLeft="@drawable/ic_play"
            android:drawableStart="@drawable/ic_play" />
    </android.support.v7.widget.CardView>
    <android.support.v7.widget.CardView
        android:layout_weight="1"

```

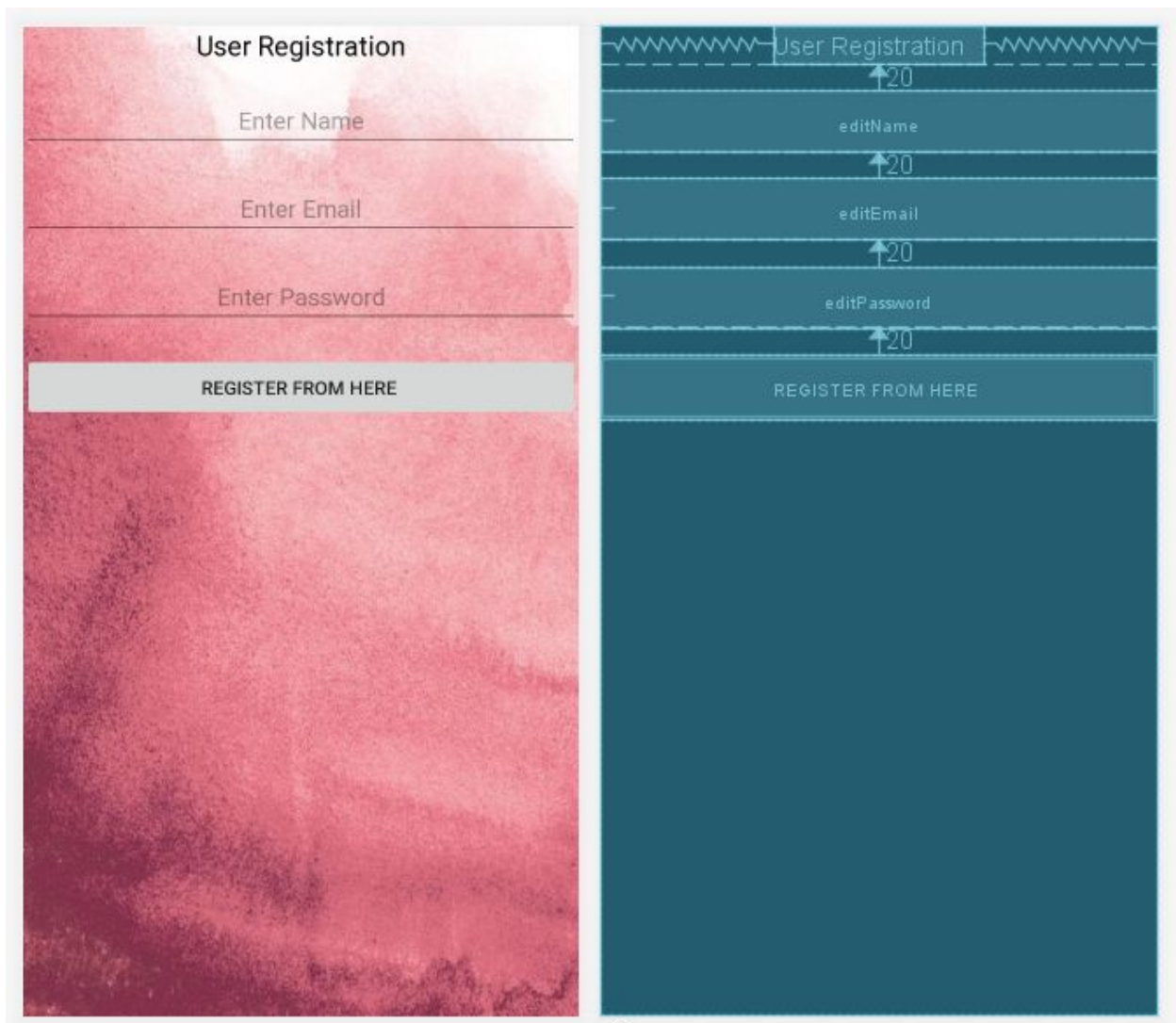
```

        app:cardUseCompatPadding="true"
        app:cardBackgroundColor="#D0FFDB"
        app:cardElevation="0dp"
        app:cardCornerRadius="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginRight="10dp">
        <TextView
            android:id="@+id/source"
            android:clickable="true"
            android:focusable="true"
            android:background="?attr/selectableItemBackground"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/source"
            android:padding="14dp"
            android:textSize="16sp"
            android:textColor="#2A7E2F"
            android:gravity="center_vertical"
            android:drawablePadding="10dp"
            android:drawableLeft="@drawable/ic_link"
            android:drawableStart="@drawable/ic_link" />
    </android.support.v7.widget.CardView>
</LinearLayout>
</LinearLayout>
</android.support.v4.widget.NestedScrollView>

</android.support.design.widget.CoordinatorLayout>

```

## Activity\_register.xml



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_register"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bk"
    tools:context="com.akash.chefhour.view.RegisterActivity">

    <TextView
        android:text="User Registration"
        android:gravity="center"
        android:textSize="20dp"
        android:textColor="#000"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:id="@+id/textView" />
```

<EditText

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:hint="Enter Name"
    android:textColor="#000"
    android:ems="10"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:id="@+id/editName"
    android:gravity="center"/>
```

<EditText

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:hint="Enter Email"
    android:textColor="#000"
    android:ems="10"
    android:layout_below="@+id/editName"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:id="@+id/editEmail"
    android:gravity="center"/>
```

<EditText

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:hint="Enter Password"
    android:textColor="#000"
    android:ems="10"
    android:layout_below="@+id/editEmail"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:id="@+id/editPassword"
    android:gravity="center"/>
```

<Button

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/buttonRegister"
    android:layout_below="@+id/editPassword"
```

```

        android:layout_marginTop="20dp"
        android:text="Register From here " />

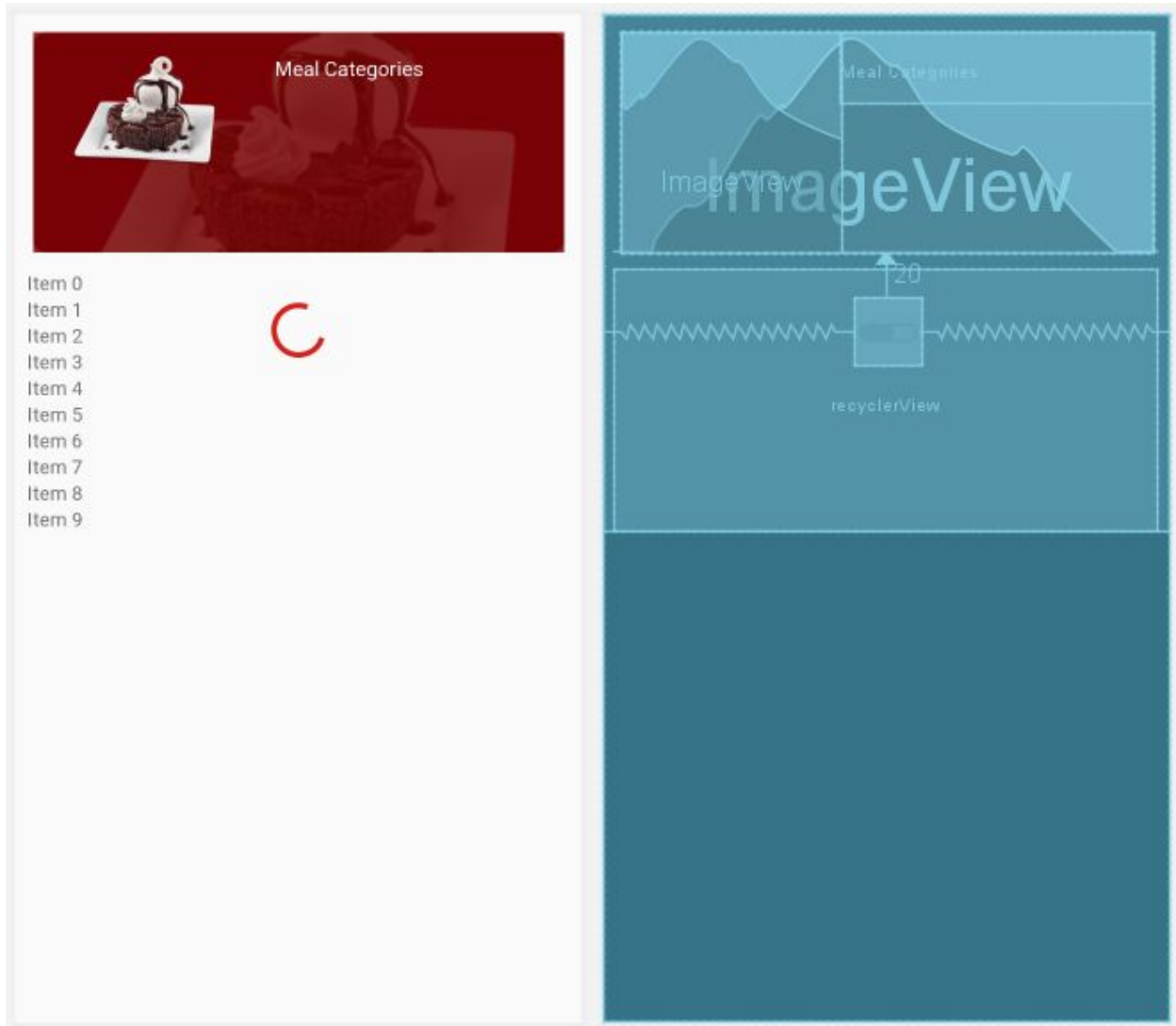
```

```

</RelativeLayout>

```

## Fragment\_category.xml



```

<?xml version="1.0" encoding="utf-8"?>

```

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```

tools:context=".view.category.CategoryFragment">

<android.support.v4.widget.NestedScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <android.support.v7.widget.CardView
            android:id="@+id/cardCategory"
            android:layout_width="match_parent"
            android:layout_height="160dp"
            android:layout_margin="12dp"
            app:cardCornerRadius="10dp"
            app:cardBackgroundColor="@color/colorPrimary"
            android:foreground="?attr/selectableItemBackground">
            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content">
                <ImageView
                    android:id="@+id/imageCategoryBg"
                    android:src="@drawable/sample_image_category"
                    android:scaleType="centerCrop"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />
                <ImageView
                    android:background="#D9770202"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent" />
                <ImageView
                    android:id="@+id/imageCategory"
                    android:src="@drawable/sample_image_category"
                    android:layout_width="160dp"
                    android:layout_height="160dp"
                    android:scaleType="fitStart"
                    android:padding="16dp"/>
                <TextView
                    android:id="@+id/textCategory"
                    android:layout_toRightOf="@id/imageCategory"
                    android:layout_toEndOf="@id/imageCategory"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:padding="16dp"
                    android:maxLines="7"
                    android:ellipsize="end"
                    android:textColor="@color/colorWhite"
                    android:textSize="15sp"
                    android:text="@string/meal_categories"/>
            </RelativeLayout>
        </android.support.v7.widget.CardView>

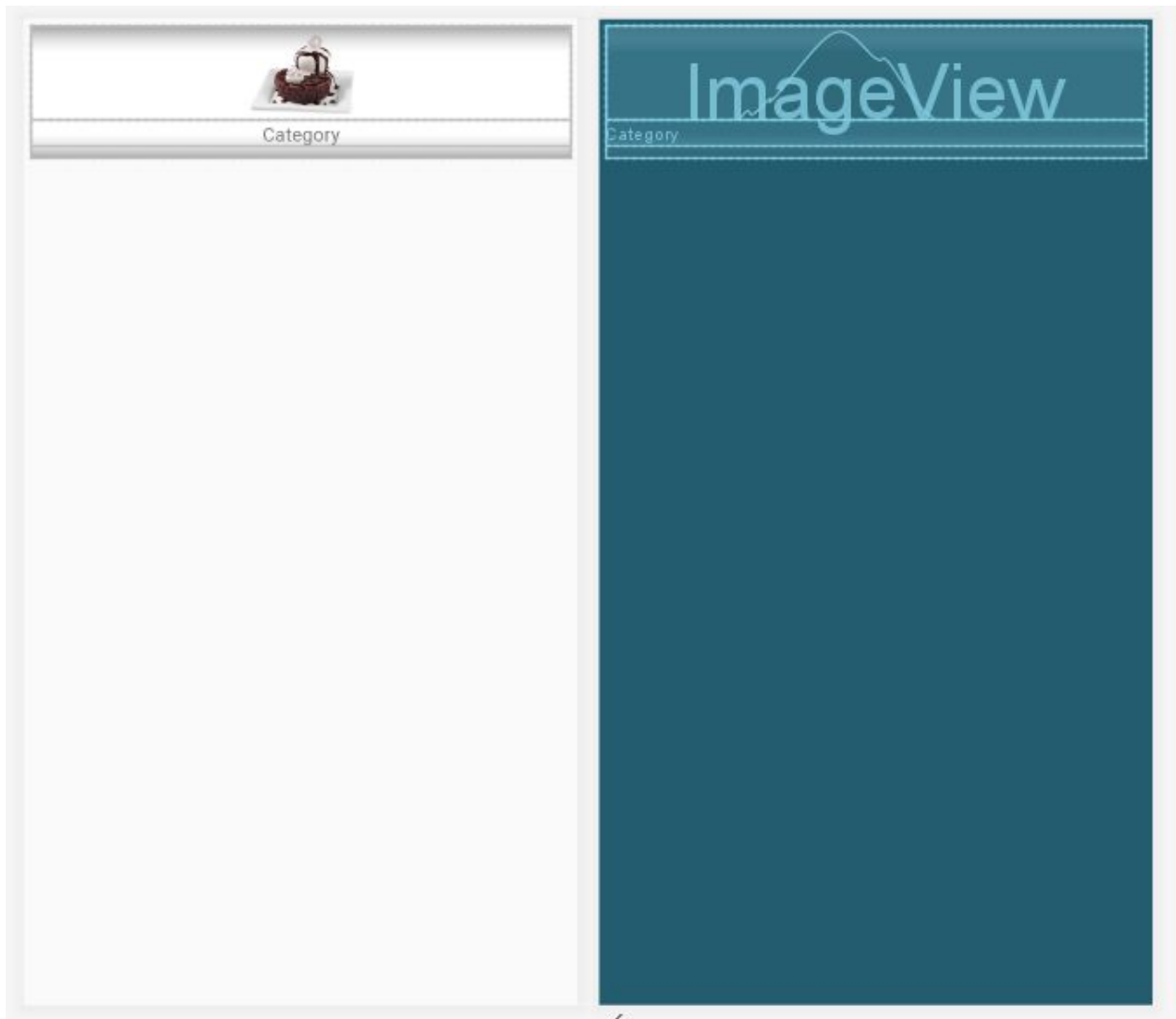
```



```
<android.support.v7.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:layout_below="@id/cardCategory"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_below="@id/cardCategory"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</RelativeLayout>
</android.support.v4.widget.NestedScrollView>

</FrameLayout>
```

## Item\_recycler\_category.xml



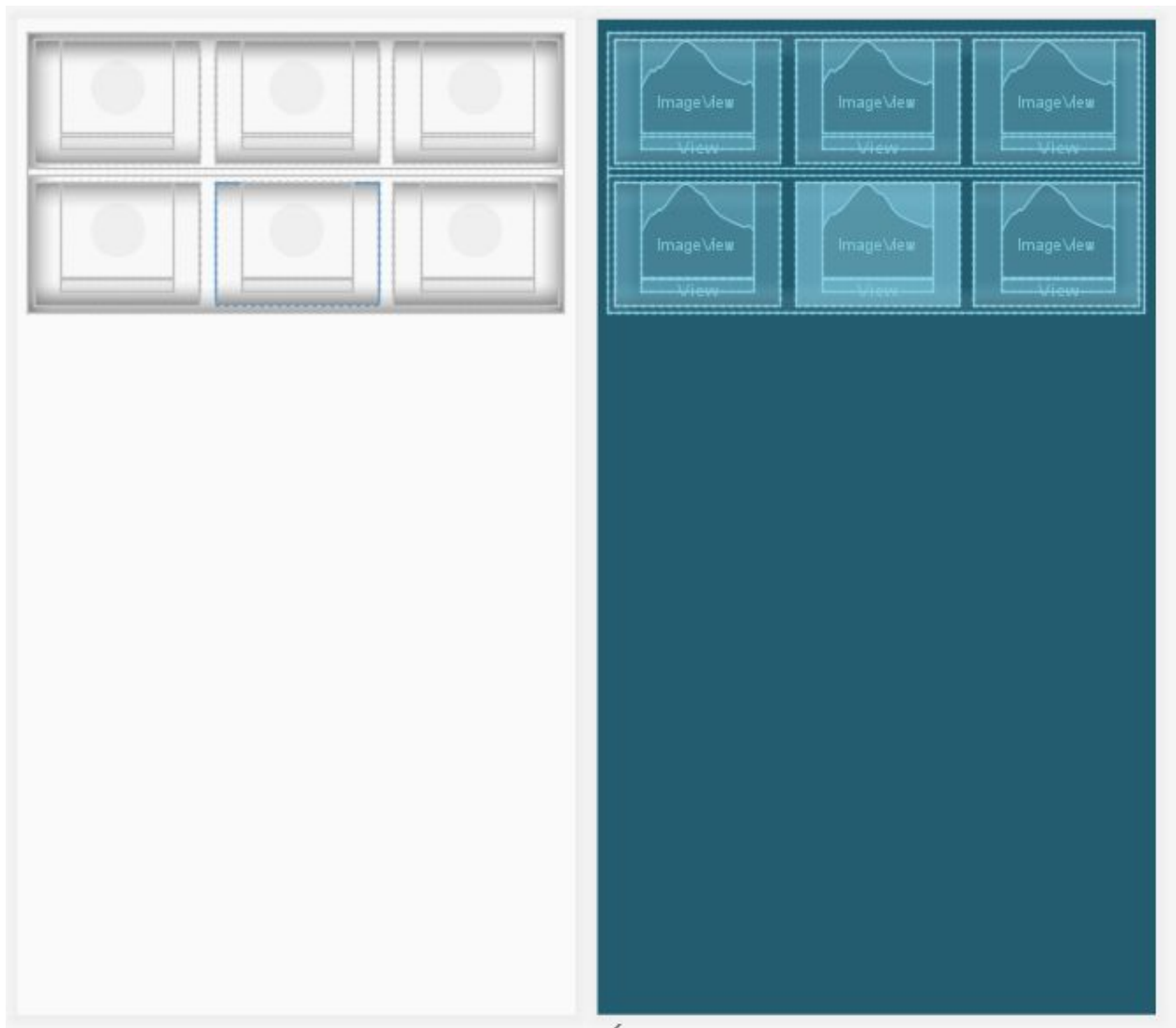
```
<?xml version="1.0" encoding="utf-8"?>

<android.support.v7.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp"
    android:foreground="?attr/selectableItemBackground"
    android:clickable="true"
    android:focusable="true">
```

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:id="@+id/categoryThumb"
        android:scaleType="centerInside"
        android:padding="5dp"
        android:src="@drawable/sample_image_category"
        android:layout_width="match_parent"
        android:layout_height="70dp"/>
    <TextView
        android:text="Category"
        android:gravity="center"
        android:id="@+id/categoryName"
        android:layout_marginBottom="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>

</android.support.v7.widget.CardView>
```

## Item\_recycler\_category\_shimmer.xml



```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="50dp"
    android:layout_below="@id/titleCategory">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="3">

                                <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_weight="1"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp">

        <com.facebook.shimmer.ShimmerFrameLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:paddingLeft="20dp"
                android:paddingRight="20dp">

                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="70dp"
                    android:padding="5dp"
                    android:src="@drawable/ic_circle" />

                <View
                    android:layout_width="match_parent"
                    android:layout_height="10dp"
                    android:layout_gravity="center_horizontal"
                    android:layout_marginBottom="10dp"
                    android:background="#DFDFDF" />

            </LinearLayout>
        </com.facebook.shimmer.ShimmerFrameLayout>
    </android.support.v7.widget.CardView>

                                <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_weight="1"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp">

```

```

<com.facebook.shimmer.ShimmerFrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="20dp"
        android:paddingRight="20dp">

        <ImageView
            android:layout_width="match_parent"
            android:layout_height="70dp"
            android:padding="5dp"
            android:src="@drawable/ic_circle" />

        <View
            android:layout_width="match_parent"
            android:layout_height="10dp"
            android:layout_gravity="center_horizontal"
            android:layout_marginBottom="10dp"
            android:background="#DFDFDF" />

    </LinearLayout>
</com.facebook.shimmer.ShimmerFrameLayout>
</android.support.v7.widget.CardView>

```

```

<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_weight="1"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp">

    <com.facebook.shimmer.ShimmerFrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingLeft="20dp"
            android:paddingRight="20dp">

            <ImageView

```

```

        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:padding="5dp"
        android:src="@drawable/ic_circle" />

    <View
        android:layout_width="match_parent"
        android:layout_height="10dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="10dp"
        android:background="#DFDFDF" />
    </LinearLayout>
</com.facebook.shimmer.ShimmerFrameLayout>
</android.support.v7.widget.CardView>
</LinearLayout>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:orientation="horizontal"
    android:weightSum="3">

    <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_weight="1"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp">

    <com.facebook.shimmer.ShimmerFrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingLeft="20dp"
            android:paddingRight="20dp">

            <ImageView
                android:layout_width="match_parent"
                android:layout_height="70dp"
                android:padding="5dp"
                android:src="@drawable/ic_circle" />

```

```

        <View
            android:layout_width="match_parent"
            android:layout_height="10dp"
            android:layout_gravity="center_horizontal"
            android:layout_marginBottom="10dp"
            android:background="#DFDFDF" />
        </LinearLayout>
    </com.facebook.shimmer.ShimmerFrameLayout>
</android.support.v7.widget.CardView>

<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    android:layout_weight="1"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp">

    <com.facebook.shimmer.ShimmerFrameLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:paddingLeft="20dp"
            android:paddingRight="20dp">

            <ImageView
                android:layout_width="match_parent"
                android:layout_height="70dp"
                android:padding="5dp"
                android:src="@drawable/ic_circle" />

            <View
                android:layout_width="match_parent"
                android:layout_height="10dp"
                android:layout_gravity="center_horizontal"
                android:layout_marginBottom="10dp"
                android:background="#DFDFDF" />
            </LinearLayout>
        </com.facebook.shimmer.ShimmerFrameLayout>
    </android.support.v7.widget.CardView>

    <android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"

```



```

        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="5dp"
        android:layout_weight="1"
        app:cardCornerRadius="10dp"
        app:cardElevation="3dp">

        <com.facebook.shimmer.ShimmerFrameLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:paddingLeft="20dp"
                android:paddingRight="20dp">

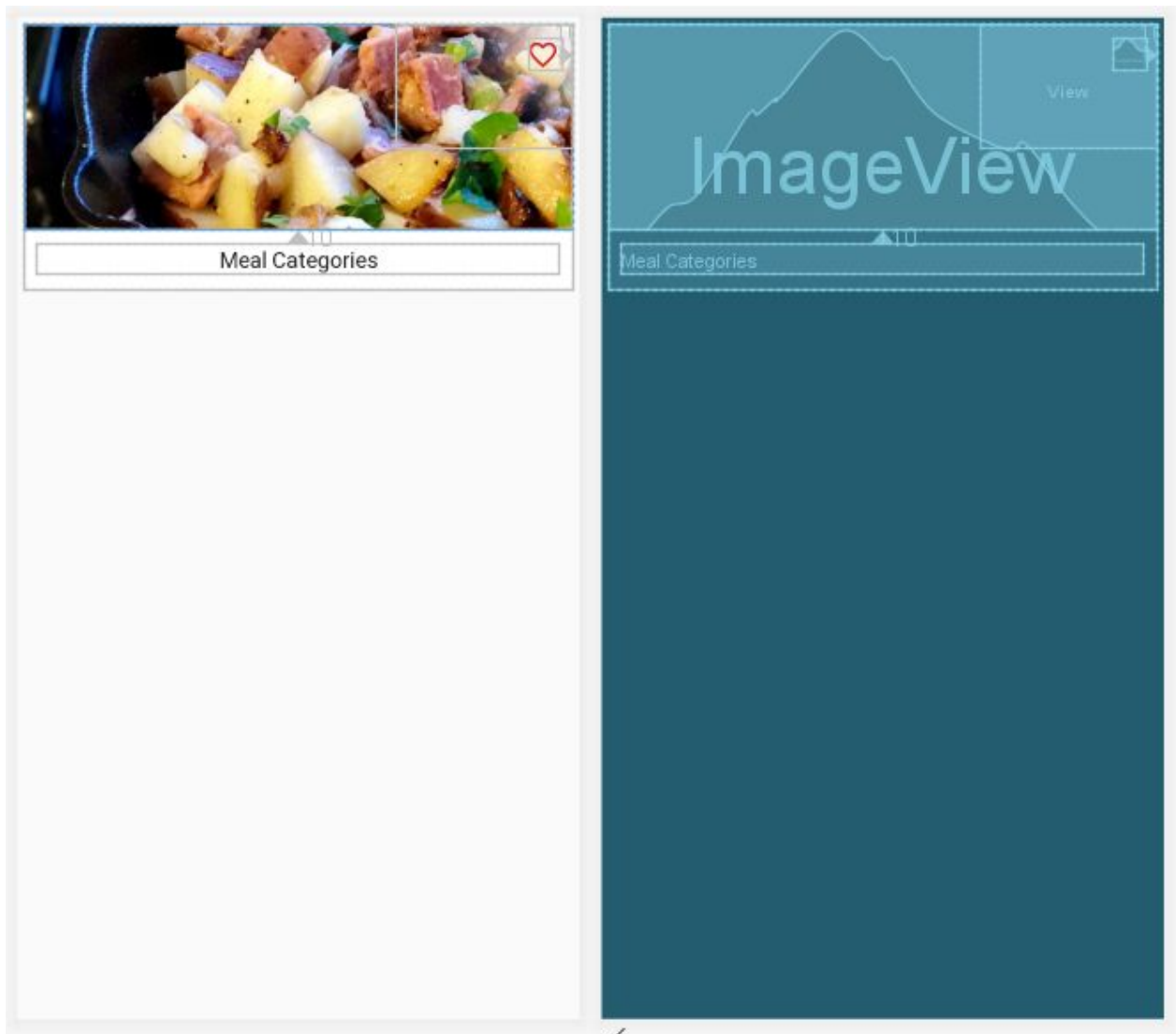
                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="70dp"
                    android:padding="5dp"
                    android:src="@drawable/ic_circle" />

                <View
                    android:layout_width="match_parent"
                    android:layout_height="10dp"
                    android:layout_gravity="center_horizontal"
                    android:layout_marginBottom="10dp"
                    android:background="#DFDFDF" />

            </LinearLayout>
        </com.facebook.shimmer.ShimmerFrameLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
</LinearLayout>

```

## Item\_recycler\_meal.xml



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="5dp"
    app:cardCornerRadius="10dp"
    app:cardElevation="3dp"
    android:foreground="?attr/selectableItemBackground"
    android:clickable="true"
    android:focusable="true">
```

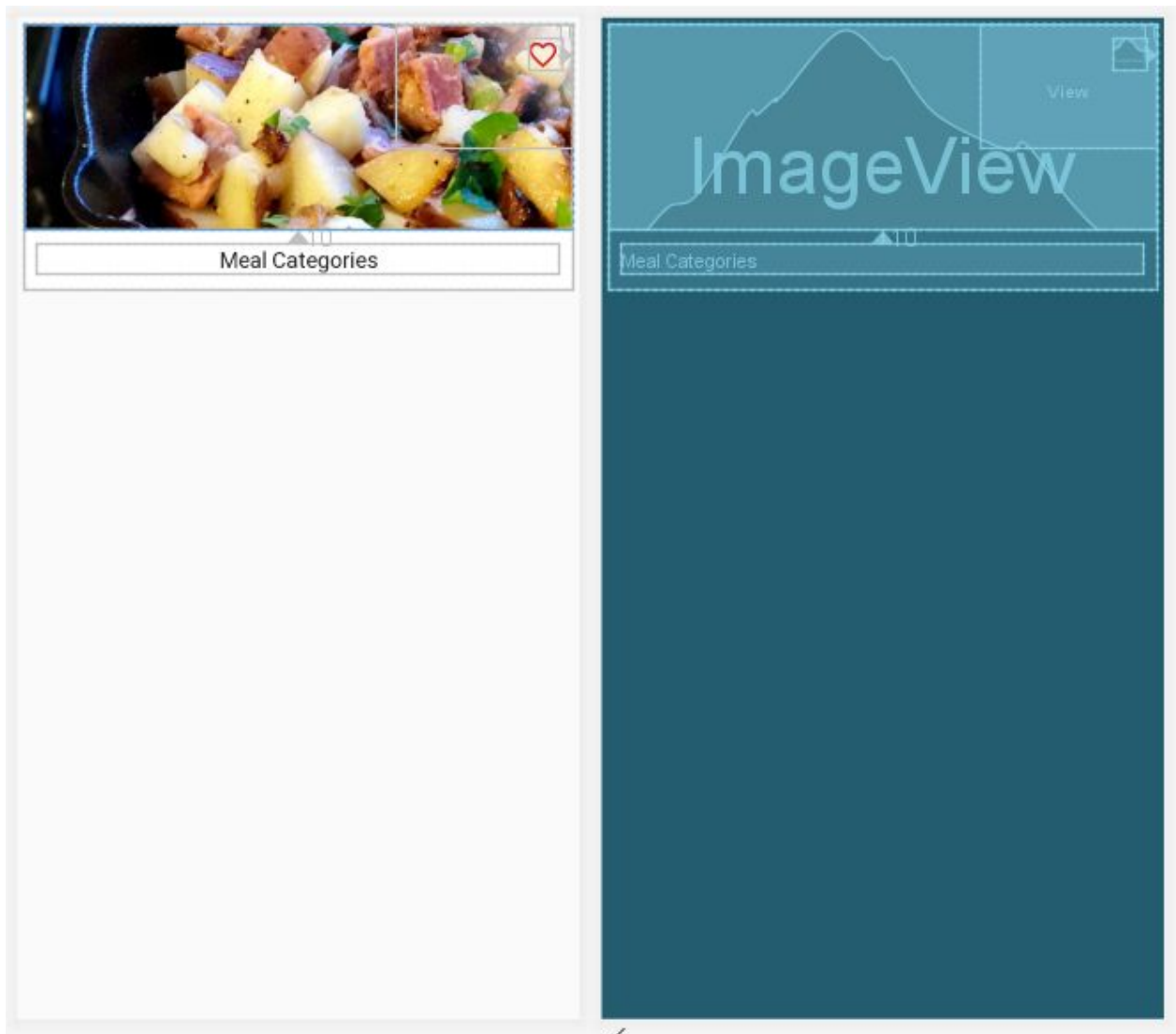
```

<RelativeLayout
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <ImageView
        android:id="@+id/mealThumb"
        android:scaleType="centerCrop"
        android:src="@drawable/sample_image_meal"
        android:layout_width="match_parent"
        android:layout_height="150dp"/>
    <View
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:background="@drawable/shadow_for_love_white"
        android:layout_width="130dp"
        android:layout_height="90dp"/>
    <ImageView
        android:id="@+id/love"
        android:tint="@color/colorAccent"
        android:src="@drawable/ic_favorite_border"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:layout_margin="10dp"
        android:clickable="true"
        android:focusable="true"
        android:background="?android:attr/selectableItemBackgroundBorderless"
        tools:ignore="NewApi" />
    <TextView
        android:text="@string/meal_categories"
        android:textSize="16sp"
        android:textColor="@color/colorPrimaryText"
        android:gravity="center"
        android:maxLines="2"
        android:ellipsize="end"
        android:id="@+id/mealName"
        android:layout_below="@id/mealThumb"
        android:layout_marginTop="10dp"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="12dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</RelativeLayout>

</android.support.v7.widget.CardView>

```

## Item\_view\_pager\_header.xml



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginBottom="10dp"
```

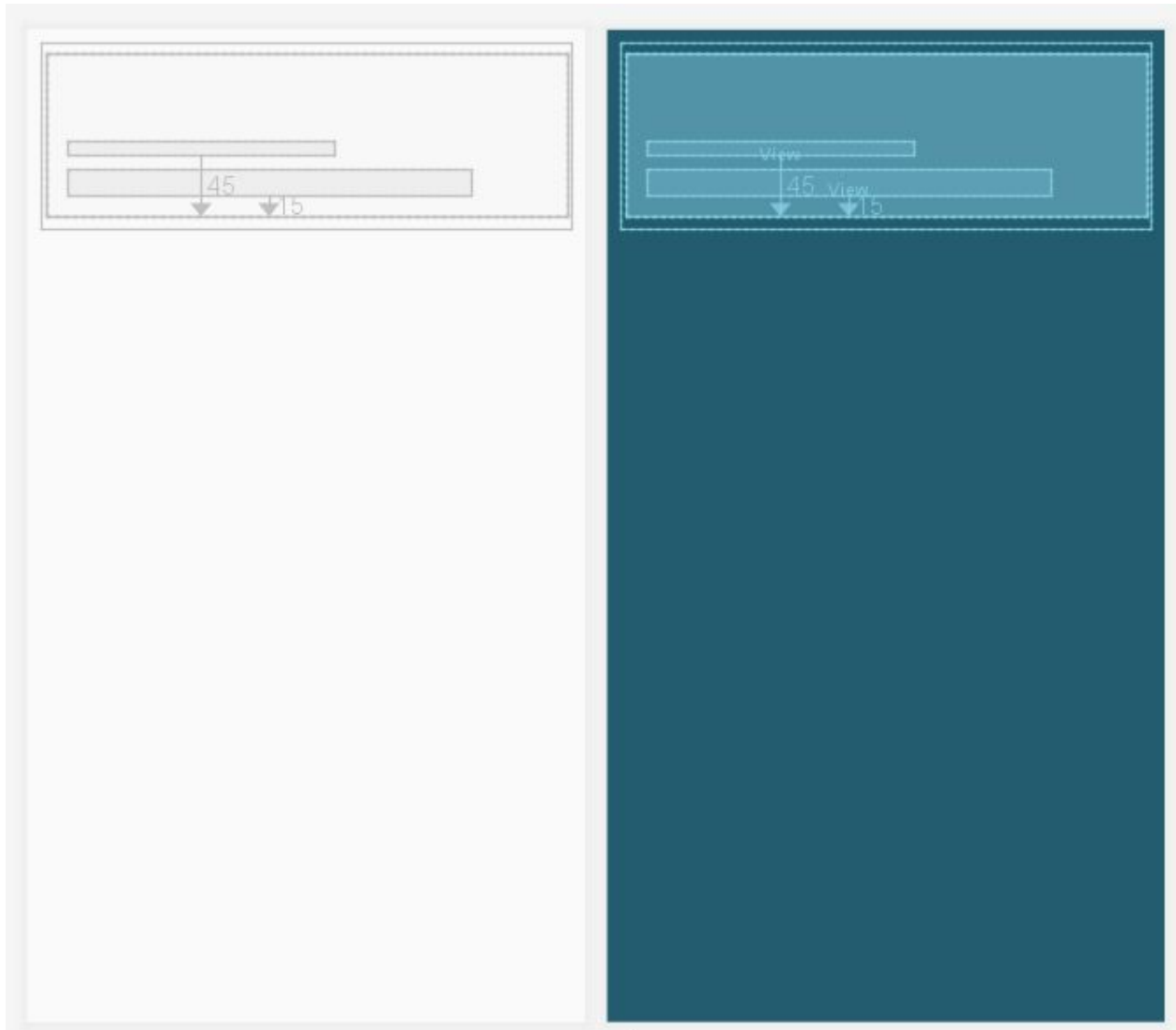
```

app:cardCornerRadius="10dp"
app:cardElevation="5dp">
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/mealThumb"
        android:src="@drawable/sample_image_meal"
        android:layout_width="match_parent"
        android:scaleType="centerCrop"
        android:layout_height="match_parent" />
    <ImageView
        android:src="@drawable/shadow_bottom_to_top"
        android:layout_width="match_parent"
        android:scaleType="centerCrop"
        android:layout_alignParentBottom="true"
        android:layout_height="match_parent" />
    <TextView
        android:id="@+id/mealName"
        android:textColor="@color/colorWhite"
        android:textSize="17sp"
        android:layout_margin="15dp"
        android:layout_alignParentBottom="true"
        android:text="Meal Name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    </RelativeLayout>
</android.support.v7.widget.CardView>

</FrameLayout>

```

## Item\_view\_pager\_header\_shimmer.xml



```
<?xml version="1.0" encoding="utf-8"?>
```

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/cardSearch"
    android:layout_margin="10dp">
```

```
    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:layout_marginTop="8dp"
        android:layout_marginLeft="4dp">
```

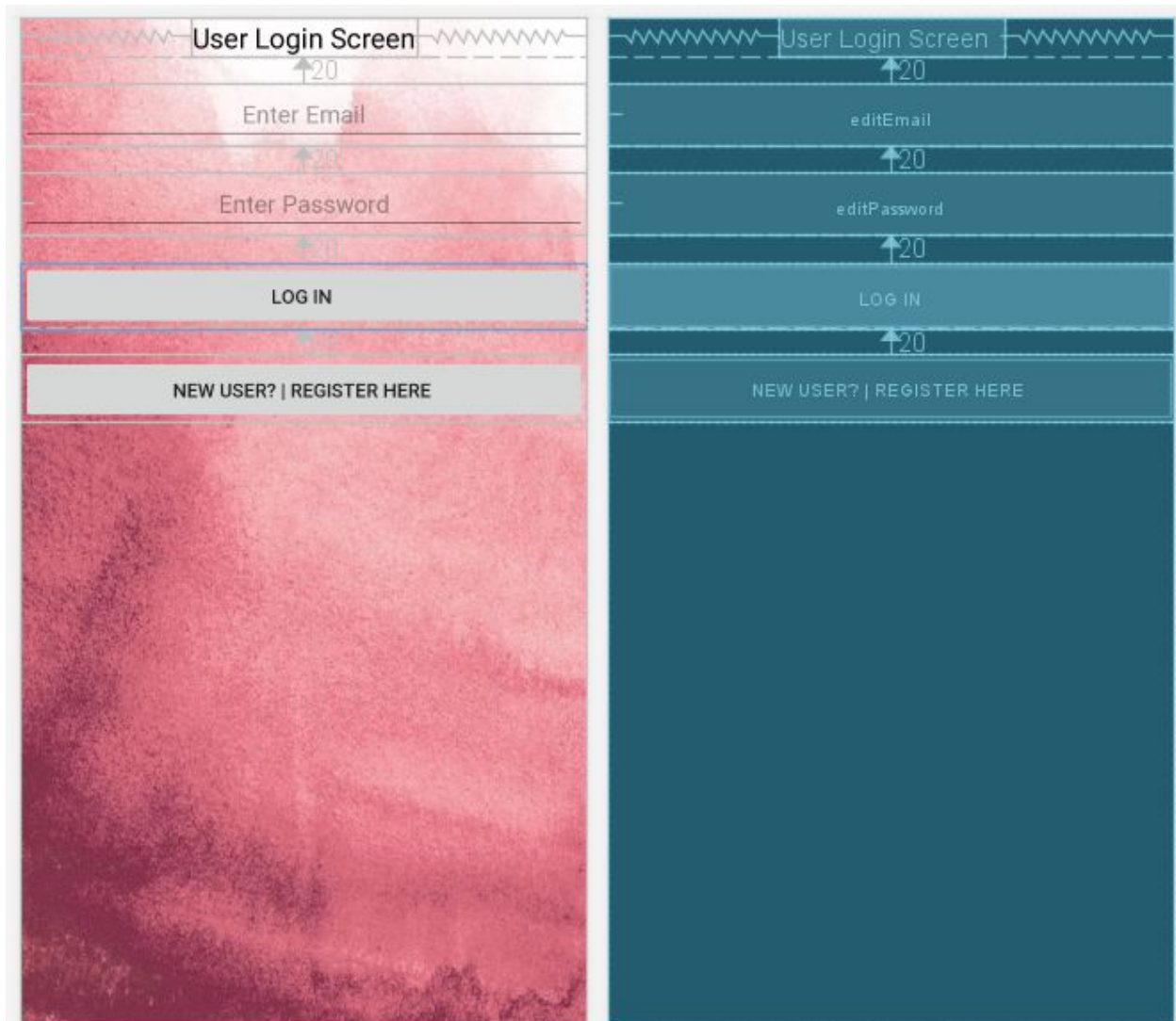
```

        android:layout_marginStart="4dp"
        android:layout_marginEnd="4dp"
        android:layout_marginRight="4dp"
        android:layout_marginBottom="10dp"
        app:cardCornerRadius="10dp"
        app:cardElevation="1dp">
        <com.facebook.shimmer.ShimmerFrameLayout
            android:id="@+id/shimmer_view_container2"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent">
                <View
                    android:layout_marginBottom="45dp"
                    android:layout_marginLeft="15dp"
                    android:layout_marginRight="170dp"
                    android:layout_alignParentBottom="true"
                    android:layout_width="match_parent"
                    android:layout_height="10dp"
                    android:background="#DFDFDF" />
                <View
                    android:layout_marginBottom="15dp"
                    android:layout_marginLeft="15dp"
                    android:layout_marginRight="70dp"
                    android:layout_alignParentBottom="true"
                    android:layout_width="match_parent"
                    android:layout_height="20dp"
                    android:background="#DFDFDF" />
            </RelativeLayout>
        </com.facebook.shimmer.ShimmerFrameLayout>
    </android.support.v7.widget.CardView>

</FrameLayout>

```

## myAccount.xml



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bk"
    tools:context="com.akash.chefhour.view.Account">

    <TextView
        android:text="User Login Screen"
        android:gravity="center"
        android:textSize="20dp"
        android:textColor="#000"
```



```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:id="@+id/textView" />
```

<EditText

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textEmailAddress"
    android:hint="Enter Email"
    android:textColor="#000"
    android:ems="10"
    android:layout_below="@+id/textView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:id="@+id/editEmail"
    android:gravity="center"/>
```

<EditText

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword"
    android:hint="Enter Password"
    android:textColor="#000"
    android:ems="10"
    android:layout_below="@+id/editEmail"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:id="@+id/editPassword"
    android:gravity="center"/>
```

<Button

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/buttonLogin"
    android:layout_below="@+id/editPassword"
    android:layout_marginTop="20dp"
    android:text="Log IN "/>
```

<Button

```
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/buttonRegister"
    android:layout_below="@+id/buttonLogin"
    android:layout_marginTop="20dp"
    android:text="New User? | Register here "/>
```

</RelativeLayout>

## Menu\_detail.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/favorite"
        android:title="@string/favorite"
        android:icon="@drawable/ic_favorite_border"
        app:showAsAction="always"/>
</menu>
```

## Ic\_launcher.xml

```
<?xml version="1.0" encoding="utf-8"?>

<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@drawable/ic_launcher_background" />
    <foreground android:drawable="@drawable/ic_launcher_foreground" />
</adaptive-icon>
```

## Ic\_launcher\_round.xml

```
<?xml version="1.0" encoding="utf-8"?>

<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@drawable/ic_launcher_background" />
    <foreground android:drawable="@drawable/ic_launcher_foreground" />
</adaptive-icon>
```

## VALUES

### Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>

<resources>
    <color name="colorPrimary">#85001F</color>
    <color name="colorPrimaryDark">#630118</color>
    <color name="colorAccent">#D8211B</color>
    <color name="colorNull">#00000000</color>
    <color name="colorIcons">#7C7F81</color>
    <color name="colorPrimaryText">#1e1e1e</color>
    <color name="colorWhite">#FFFFFF</color>
    <color name="colorWhiteDark">#DBDBDB</color>
    <color name="colorPrimaryLight">#CA7474</color>
    <color name="colorBackground">#F4F7F7</color>
</resources>
```

### String.xml

```
<resources>
    <string name="app_name">Chef Hour</string>
    <string name="search_your_recipes">Search Your Recipes...[UNDER BUILD]</string>
    <string name="meal_categories">Meal Categories</string>
    <string name="category">Category</string>
    <string name="country">Country</string>
    <string name="instructions">INSTRUCTIONS</string>
    <string name="dot_loading">...</string>
    <string name="Ingredients">INGREDIENTS</string>
    <string name="youtube">YOUTUBE</string>
    <string name="source">SOURCE</string>
    <string name="favorite">Favorite</string>
</resources>
```

### Styles.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    </style>
</resources>
```

```

        <item name="colorAccent">@color/colorAccent</item>
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>

    <style name="AppTheme.NoActionBar" parent="AppTheme">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
        <item name="android:statusBarColor"
tools:targetApi="lollipop">@android:color/transparent</item>
        <item name="android:windowTranslucentStatus"
tools:targetApi="kitkat">true</item>
    </style>

    <style name="AppBarLayoutTheme" parent="@style/Theme.AppCompat.Light.NoActionBar">
        <item name="colorControlNormal">@color/colorPrimary</item>
        <item name="titleTextColor">@color/colorPrimary</item>
    </style>

    <style name="AppTheme.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar"
/>
    <style name="AppTheme.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />

</resources>

```

## JAVA Files

### HomeActivity.java

```
package com.akash.chefhour.view.home;

import android.content.Intent;
import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.akash.chefhour.R;
import com.akash.chefhour.Utills;
import com.akash.chefhour.adapter.RecyclerViewHomeAdapter;
import com.akash.chefhour.adapter.ViewPagerHeaderAdapter;
import com.akash.chefhour.model.Categories;
import com.akash.chefhour.model.Meals;
import com.akash.chefhour.view.category.CategoryActivity;
import com.akash.chefhour.view.detail.DetailActivity;
import com.akash.chefhour.view.Account;

import org.w3c.dom.Text;

import java.io.Serializable;
import java.util.List;

import butterknife.BindView;
import butterknife.ButterKnife;

public class HomeActivity extends AppCompatActivity implements HomeView {

    public static final String EXTRA_CATEGORY = "category";
    public static final String EXTRA_POSITION = "position";
    public static final String EXTRA_DETAIL = "detail";

    @BindView(R.id.viewPagerHeader) ViewPager viewPagerMeal;
    @BindView(R.id.recyclerCategory) RecyclerView recyclerViewCategory;

    HomePresenter presenter;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

```

        setContentView(R.layout.activity_home);
        ButterKnife.bind(this);

        presenter = new HomePresenter(this);
        presenter.getMeals();
        presenter.getCategories();
        Button acc = (Button) findViewById(R.id.account);
        acc.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent intent = new Intent(HomeActivity.this, Account.class);
                HomeActivity.this.startActivity(intent);
            }
        });
    }

    @Override
    public void showLoading() {
        findViewById(R.id.shimmerMeal).setVisibility(View.VISIBLE);
        findViewById(R.id.shimmerCategory).setVisibility(View.VISIBLE);
    }

    @Override
    public void hideLoading() {
        findViewById(R.id.shimmerMeal).setVisibility(View.GONE);
        findViewById(R.id.shimmerCategory).setVisibility(View.GONE);
    }

    @Override
    public void setMeal(List<Meals.Meal> meal) {
        ViewPagerHeaderAdapter headerAdapter = new ViewPagerHeaderAdapter(meal, this);
        viewPagerMeal.setAdapter(headerAdapter);
        viewPagerMeal.setPadding(20, 0, 150, 0);
        headerAdapter.notifyDataSetChanged();

        headerAdapter.setOnItemClickListener((view, position) -> {
            TextView mealName = view.findViewById(R.id.mealName);
            Intent intent = new Intent(getApplicationContext(), DetailActivity.class);
            intent.putExtra(EXTRA_DETAIL, mealName.getText().toString());
            startActivity(intent);
        });
    }

    @Override
    public void setCategory(List<Categories.Category> category) {
        RecyclerViewHomeAdapter homeAdapter = new RecyclerViewHomeAdapter(category,
this);
        recyclerViewCategory.setAdapter(homeAdapter);
        GridLayoutManager layoutManager = new GridLayoutManager(this, 3,

```

```

        GridLayoutManager.VERTICAL, false);
recyclerViewCategory.setLayoutManager(layoutManager);
recyclerViewCategory.setNestedScrollingEnabled(true);
homeAdapter.notifyDataSetChanged();

homeAdapter.setOnItemClickListener((view, position) -> {
    Intent intent = new Intent(this, CategoryActivity.class);
    intent.putExtra(EXTRA_CATEGORY, (Serializable) category);
    intent.putExtra(EXTRA_POSITION, position);
    startActivity(intent);
});
}

@Override
public void onErrorLoading(String message) {
    Utils.showDialogMessage(this, "Title", message);
}
}

```

## HomePresenter.java

```

/*-----
-
- Last modified 3/17/19 5:24 AM
-
-
-----*/

package com.akash.chefhour.view.home;

import android.support.annotation.NonNull;

import com.akash.chefhour.Utils;
import com.akash.chefhour.model.Categories;
import com.akash.chefhour.model.Meals;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

class HomePresenter {

    private HomeView view;

    public HomePresenter(HomeView view) {
        this.view = view;
    }
}

```

```

void getMeals() {

    view.showLoading();

    Call<Meals> mealsCall = Utils.getApi().getMeal();
    mealsCall.enqueue(new Callback<Meals>() {
        @Override
        public void onResponse(@NonNull Call<Meals> call, @NonNull Response<Meals>
response) {
            view.hideLoading();

            if (response.isSuccessful() && response.body() != null) {

                view.setMeal(response.body().getMeals());

            }
            else {
                view.onErrorLoading(response.message());
            }
        }

        @Override
        public void onFailure(@NonNull Call<Meals> call, @NonNull Throwable t) {
            view.hideLoading();
            view.onErrorLoading(t.getLocalizedMessage());
        }
    });
}

```

```

void getCategories() {

    view.showLoading();

    Call<Categories> categoriesCall = Utils.getApi().getCategories();
    categoriesCall.enqueue(new Callback<Categories>() {
        @Override
        public void onResponse(@NonNull Call<Categories> call,
                                @NonNull Response<Categories> response) {

            view.hideLoading();
            if (response.isSuccessful() && response.body() != null) {

                view.setCategory(response.body().getCategories());

            }
            else {
                view.onErrorLoading(response.message());
            }
        }
    })
}

```



```

        @Override
        public void onFailure(@NonNull Call<Categories> call, @NonNull Throwable t)
        {
            view.hideLoading();
            view.onErrorLoading(t.getLocalizedMessage());
        }
    });
}
}

```

## HomeView.java

```

/*-----
-                                     -
- Last modified 3/17/19 5:24 AM          -
-                                     -
-                                     -
-----*/

package com.akash.chefhour.view.home;

import com.akash.chefhour.model.Categories;
import com.akash.chefhour.model.Meals;

import java.util.List;

public interface HomeView {
    void showLoading();
    void hideLoading();
    void setMeal(List<Meals.Meal> meal);
    void setCategory(List<Categories.Category> category);
    void onErrorLoading(String message);
}

```

## CategoryActivity.java

```

/*-----
-                                     -
- Last modified 3/24/19 12:55 PM          -
-                                     -
-                                     -
-----*/

package com.akash.chefhour.view.category;

import android.content.Intent;
import android.os.Bundle;
import android.support.design.widget.TabLayout;
import android.support.v4.app.FragmentManager;
import android.support.v4.view.ViewPager;

```

```

import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.MenuItem;

import com.akash.chefhour.R;
import com.akash.chefhour.adapter.ViewPagerCategoryAdapter;
import com.akash.chefhour.model.Categories;
import com.akash.chefhour.view.home.HomeActivity;

import java.util.List;

import butterknife.BindView;
import butterknife.ButterKnife;

public class CategoryActivity extends AppCompatActivity {

    @BindView(R.id.toolbar)
    Toolbar toolbar;
    @BindView(R.id.tabLayout)
    TabLayout tabLayout;
    @BindView(R.id.viewPager)
    ViewPager viewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_category);
        ButterKnife.bind(this);

        initActionBar();
        initIntent();
    }

    private void initIntent() {
        Intent intent = getIntent();
        List<Categories.Category> categories =
                                                    (List<Categories.Category>)
intent.getSerializableExtra(HomeActivity.EXTRA_CATEGORY);
        int position = intent.getIntExtra(HomeActivity.EXTRA_POSITION, 0);

        ViewPagerCategoryAdapter adapter = new ViewPagerCategoryAdapter(
            getSupportFragmentManager(),
            categories);
        viewPager.setAdapter(adapter);
        tabLayout.setupWithViewPager(viewPager);
        viewPager.setCurrentItem(position, true);
        adapter.notifyDataSetChanged();
    }
}

```

```

private void initActionBar() {
    setSupportActionBar(toolbar);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            onBackPressed();
            break;
    }
    return true;
}
}

```

## CategoryFragment.java

```

/*-----
-
- Last modified 3/24/19 1:02 PM
-
-
-----*/

package com.akash.chefhour.view.category;

import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v7.app.AlertDialog;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

import com.akash.chefhour.R;
import com.akash.chefhour.Utills;
import com.akash.chefhour.adapter.RecyclerViewMealByCategory;
import com.akash.chefhour.model.Meals;

```

```

import com.akash.chefhour.view.detail.DetailActivity;
import com.squareup.picasso.Picasso;

import java.util.List;

import butterknife.BindView;
import butterknife.ButterKnife;
import butterknife.OnClick;

import static com.akash.chefhour.view.home.HomeActivity.EXTRA_DETAIL;

public class CategoryFragment extends Fragment implements CategoryView {

    @BindView(R.id.recyclerView)
    RecyclerView recyclerView;
    @BindView(R.id.progressBar)
    ProgressBar progressBar;
    @BindView(R.id.imageCategory)
    ImageView imageCategory;
    @BindView(R.id.imageCategoryBg)
    ImageView imageCategoryBg;
    @BindView(R.id.textCategory)
    TextView textCategory;

    AlertDialog.Builder descDialog;

    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_category, container, false);
        ButterKnife.bind(this, view);
        return view;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState)
    {
        super.onViewCreated(view, savedInstanceState);

        if (getArguments() != null) {
            textCategory.setText(getArguments().getString("EXTRA_DATA_DESC"));
            Picasso.get()
                    .load(getArguments().getString("EXTRA_DATA_IMAGE"))
                    .into(imageCategory);
            Picasso.get()
                    .load(getArguments().getString("EXTRA_DATA_IMAGE"))
                    .into(imageCategoryBg);
            descDialog = new AlertDialog.Builder(getActivity())
                    .setTitle(getArguments().getString("EXTRA_DATA_NAME"))
                    .setMessage(getArguments().getString("EXTRA_DATA_DESC"));
        }
    }
}

```

```

        CategoryPresenter presenter = new CategoryPresenter(this);
        presenter.getMealByCategory(getArguments().getString("EXTRA_DATA_NAME"));
    }
}

@Override
public void showLoading() {
    progressBar.setVisibility(View.VISIBLE);
}

@Override
public void hideLoading() {
    progressBar.setVisibility(View.GONE);
}

@Override
public void setMeals(List<Meals.Meal> meals) {
    RecyclerViewMealByCategory adapter =
        new RecyclerViewMealByCategory(getActivity(), meals);
    recyclerView.setLayoutManager(new GridLayoutManager(getActivity(), 2));
    recyclerView.setClipToPadding(false);
    recyclerView.setAdapter(adapter);
    adapter.notifyDataSetChanged();

    adapter.setOnItemClickListener((view, position) -> {
        TextView mealName = view.findViewById(R.id.mealName);
        Intent intent = new Intent(getActivity(), DetailActivity.class);
        intent.putExtra(EXTRA_DETAIL, mealName.getText().toString());
        startActivity(intent);
    });
}

@Override
public void onErrorLoading(String message) {
    Utils.showDialogMessage(getActivity(), "Error ", message);
}

@OnClick(R.id.cardCategory)
public void onClick() {
    descDialog.setPositiveButton("CLOSE", (dialog, which) -> dialog.dismiss());
    descDialog.show();
}
}

```

## CategoryPresenter.java

```

/*-----

```

```

-
- Last modified 3/24/19 5:13 AM
-
-
-----*/
package com.akash.chefhour.view.category;

import android.support.annotation.NonNull;

import com.akash.chefhour.Utils;
import com.akash.chefhour.model.Meals;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class CategoryPresenter {
    private CategoryView view;

    public CategoryPresenter(CategoryView view) {
        this.view = view;
    }

    void getMealByCategory(String category) {

        view.showLoading();
        Call<Meals> mealsCall = Utils.getApi().getMealByCategory(category);
        mealsCall.enqueue(new Callback<Meals>() {
            @Override
            public void onResponse(@NonNull Call<Meals> call, @NonNull Response<Meals>
response) {
                view.hideLoading();
                if (response.isSuccessful() && response.body() != null) {
                    view.setMeals(response.body().getMeals());
                } else {
                    view.onErrorLoading(response.message());
                }
            }

            @Override
            public void onFailure(@NonNull Call<Meals> call, @NonNull Throwable t) {
                view.hideLoading();
                view.onErrorLoading(t.getLocalizedMessage());
            }
        });
    }
}

```

## CategoryView.java

```
/*-----  
- - - - -  
- Last modified 3/24/19 3:33 AM -  
- - - - -  
-----*/  
  
package com.akash.chefhour.view.category;  
  
import com.akash.chefhour.model.Meals;  
  
import java.util.List;  
  
public interface CategoryView {  
    void showLoading();  
    void hideLoading();  
    void setMeals(List<Meals.Meal> meals);  
    void onErrorLoading(String message);  
}
```

## DetailActivity.java

```
/*-----  
- - - - -  
- Last modified 4/7/19 7:33 PM -  
- - - - -  
-----*/  
  
package com.akash.chefhour.view.detail;  
  
import android.content.Intent;  
import android.graphics.PorterDuff;  
import android.graphics.drawable.Drawable;  
import android.net.Uri;  
import android.os.Bundle;  
import android.support.design.widget.AppBarLayout;  
import android.support.design.widget.CollapsingToolbarLayout;  
import android.support.v4.view.ViewCompat;  
import android.support.v7.app.AppCompatActivity;  
import android.support.v7.widget.Toolbar;  
import android.util.Log;  
import android.view.Menu;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.ImageView;  
import android.widget.ProgressBar;  
import android.widget.TextView;
```

```

import com.akash.chefhour.R;
import com.akash.chefhour.Utils;
import com.akash.chefhour.model.Meals;
import com.squareup.picasso.Picasso;

import butterknife.BindView;
import butterknife.ButterKnife;

import static com.akash.chefhour.view.home.HomeActivity.EXTRA_DETAIL;

public class DetailActivity extends AppCompatActivity implements DetailView {

    @BindView(R.id.toolbar)
    Toolbar toolbar;

    @BindView(R.id.appbar)
    AppBarLayout appBarLayout;

    @BindView(R.id.collapsing_toolbar)
    CollapsingToolbarLayout collapsingToolbarLayout;

    @BindView(R.id.mealThumb)
    ImageView mealThumb;

    @BindView(R.id.category)
    TextView category;

    @BindView(R.id.country)
    TextView country;

    @BindView(R.id.instructions)
    TextView instructions;

    @BindView(R.id.ingredient)
    TextView ingredients;

    @BindView(R.id.measure)
    TextView measures;

    @BindView(R.id.progressBar)
    ProgressBar progressBar;

    @BindView(R.id.youtube)
    TextView youtube;

    @BindView(R.id.source)
    TextView source;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```



```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);
        ButterKnife.bind(this);

        setupActionBar();

        Intent intent = getIntent();

        String mealName = intent.getStringExtra(EXTRA_DETAIL);

        DetailPresenter presenter = new DetailPresenter(this);
        presenter.getMealById(mealName);
    }

    private void setupActionBar() {
        setSupportActionBar(toolbar);

        collapsingToolbarLayout.setContentScrimColor(getResources().getColor(R.color.colorWhite));

        collapsingToolbarLayout.setCollapsedTitleTextColor(getResources().getColor(R.color.colorPrimary));

        collapsingToolbarLayout.setExpandedTitleColor(getResources().getColor(R.color.colorWhite));

        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }
    }

    void setupColorActionBarIcon(Drawable favoriteItemColor) {
        appBarLayout.addOnOffsetChangedListener((appBarLayout, verticalOffset) -> {
            if ((collapsingToolbarLayout.getHeight() + verticalOffset) < (2 *
ViewCompat.getMinimumHeight(collapsingToolbarLayout))) {
                if (toolbar.getNavigationIcon() != null)

toolbar.getNavigationIcon().setColorFilter(getResources().getColor(R.color.colorPrimary), PorterDuff.Mode.SRC_ATOP);

favoriteItemColor.mutate().setColorFilter(getResources().getColor(R.color.colorPrimary),
PorterDuff.Mode.SRC_ATOP);

            } else {
                if (toolbar.getNavigationIcon() != null)

toolbar.getNavigationIcon().setColorFilter(getResources().getColor(R.color.colorWhite), PorterDuff.Mode.SRC_ATOP);

favoriteItemColor.mutate().setColorFilter(getResources().getColor(R.color.colorWhite),

```

```

        PorterDuff.Mode.SRC_ATOP);
    }
});
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_detail, menu);
    MenuItem favoriteItem = menu.findItem(R.id.favorite);
    Drawable favoriteItemColor = favoriteItem.getIcon();
    setupColorActionBarIcon(favoriteItemColor);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home :
            onBackPressed();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

@Override
public void showLoading() {
    progressBar.setVisibility(View.VISIBLE);
}

@Override
public void hideLoading() {
    progressBar.setVisibility(View.INVISIBLE);
}

@Override
public void setMeal(Meals.Meal meal) {
    Picasso.get().load(meal.getStrMealThumb()).into(mealThumb);
    collapsingToolbarLayout.setTitle(meal.getStrMeal());
    category.setText(meal.getStrCategory());
    country.setText(meal.getStrArea());
    instructions.setText(meal.getStrInstructions());
    setupActionBar();

    //===

    if (!meal.getStrIngredient1().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient1());
    }
    if (!meal.getStrIngredient2().isEmpty()) {

```

```

        ingredients.append("\n \u2022 " + meal.getStrIngredient2());
    }
    if (!meal.getStrIngredient3().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient3());
    }
    if (!meal.getStrIngredient4().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient4());
    }
    if (!meal.getStrIngredient5().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient5());
    }
    if (!meal.getStrIngredient6().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient6());
    }
    if (!meal.getStrIngredient7().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient7());
    }
    if (!meal.getStrIngredient8().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient8());
    }
    if (!meal.getStrIngredient9().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient9());
    }
    if (!meal.getStrIngredient10().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient10());
    }
    if (!meal.getStrIngredient11().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient11());
    }
    if (!meal.getStrIngredient12().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient12());
    }
    if (!meal.getStrIngredient13().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient13());
    }
    if (!meal.getStrIngredient14().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient14());
    }
    if (!meal.getStrIngredient15().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient15());
    }
    if (!meal.getStrIngredient16().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient16());
    }
    if (!meal.getStrIngredient17().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient17());
    }
    if (!meal.getStrIngredient18().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient18());
    }
}

```

```

    if (!meal.getStrIngredient19().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient19());
    }
    if (!meal.getStrIngredient20().isEmpty()) {
        ingredients.append("\n \u2022 " + meal.getStrIngredient20());
    }

        if (!meal.getStrMeasure1().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure1().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure1());
        }
        if (!meal.getStrMeasure2().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure2().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure2());
        }
        if (!meal.getStrMeasure3().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure3().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure3());
        }
        if (!meal.getStrMeasure4().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure4().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure4());
        }
        if (!meal.getStrMeasure5().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure5().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure5());
        }
        if (!meal.getStrMeasure6().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure6().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure6());
        }
        if (!meal.getStrMeasure7().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure7().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure7());
        }
        if (!meal.getStrMeasure8().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure8().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure8());
        }
        if (!meal.getStrMeasure9().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure9().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure9());
        }
        if (!meal.getStrMeasure10().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure10().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure10());
        }
        if (!meal.getStrMeasure11().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure11().charAt(0)) {
            measures.append("\n : " + meal.getStrMeasure11());
        }

```

```

    }

    if (!meal.getStrMeasure12().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure12().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure12());
}

    if (!meal.getStrMeasure13().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure13().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure13());
}

    if (!meal.getStrMeasure14().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure14().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure14());
}

    if (!meal.getStrMeasure15().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure15().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure15());
}

    if (!meal.getStrMeasure16().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure16().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure16());
}

    if (!meal.getStrMeasure17().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure17().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure17());
}

    if (!meal.getStrMeasure18().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure18().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure18());
}

    if (!meal.getStrMeasure19().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure19().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure19());
}

    if (!meal.getStrMeasure20().isEmpty()) &&
!Character.isWhitespace(meal.getStrMeasure20().charAt(0)) {
    measures.append("\n : " + meal.getStrMeasure20());
}

}

youtube.setOnClickListener(v -> {
    Intent intentYoutube = new Intent(Intent.ACTION_VIEW);
    intentYoutube.setData(Uri.parse(meal.getStrYoutube()));
    startActivity(intentYoutube);
});

source.setOnClickListener(v -> {
    Intent intentSource = new Intent(Intent.ACTION_VIEW);
    intentSource.setData(Uri.parse(meal.getStrSource()));
    startActivity(intentSource);
});

```

```

    }

    @Override
    public void onErrorLoading(String message) {
        Utils.showDialogMessage(this, "Error", message);
    }
}

```

## DetailPresenter.java

```

/*-----
-
- Last modified 4/7/19 7:07 PM
-
-
-----*/
package com.akash.chefhour.view.detail;

import android.support.annotation.NonNull;

import com.akash.chefhour.Utils;
import com.akash.chefhour.model.Meals;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class DetailPresenter {
    private DetailView view;

    public DetailPresenter(DetailView view) {
        this.view = view;
    }

    void getMealById(String mealName) {

        view.showLoading();

        Utils.getApi().getMealByName(mealName).enqueue(new Callback<Meals>() {
            @Override
            public void onResponse(@NonNull Call<Meals> call, @NonNull Response<Meals>
response) {
                view.hideLoading();
                if(response.isSuccessful() && response.body() != null){
                    view.setMeal(response.body().getMeals().get(0));
                }
                else{
                    view.onErrorLoading(response.message());
                }
            }
        });
    }
}

```

```

        }

        @Override
        public void onFailure(@NonNull Call<Meals> call, @NonNull Throwable t) {
            view.hideLoading();
            view.onErrorLoading(t.getLocalizedMessage());
        }
    });
}
}
}

```

## DetailView.java

```

/*-----
-                                     -
- Last modified 4/7/19 5:49 PM                                     -
-                                     -
-                                     -
-----*/
package com.akash.chefhour.view.detail;

import com.akash.chefhour.model.Meals;

public interface DetailView {
    void showLoading();
    void hideLoading();
    void setMeal(Meals.Meal meal);
    void onErrorLoading(String message);
}

```

## Account.java

```

package com.akash.chefhour.view;

import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

```

```

import com.akash.chefhour.R;

public class Account extends AppCompatActivity {

    Button LogInButton, RegisterButton ;
    EditText Email, Password ;
    String EmailHolder, PasswordHolder;
    Boolean EditTextEmptyHolder;
    SQLiteDatabase sqLiteDatabaseObj;
    SQLiteHelper sqLiteHelper;
    Cursor cursor;
    String TempPassword = "NOT_FOUND" ;
    public static final String userEmail = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.myaccount);

        LogInButton = (Button)findViewById(R.id.buttonLogin);

        RegisterButton = (Button)findViewById(R.id.buttonRegister);

        Email = (EditText)findViewById(R.id.editEmail);
        Password = (EditText)findViewById(R.id.editPassword);

        sqLiteHelper = new SQLiteHelper(this);

        //Adding click listener to log in button.
        LogInButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                // Calling EditText is empty or no method.
                CheckEditTextStatus();

                // Calling login method.
                LoginFunction();

            }
        });

        // Adding click listener to register button.
        RegisterButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                // Opening new user registration activity using intent on button click.
                Intent intent = new Intent(Account.this, RegisterActivity.class);

```



```

        startActivity(intent);

    }

});

}

// Login function starts from here.
public void LoginFunction() {

    if(EditTextEmptyHolder) {

        // Opening SQLite database write permission.
        sqLiteDatabaseObj = sqLiteHelper.getWritableDatabase();

        // Adding search email query to cursor.
        cursor = sqLiteDatabaseObj.query(SQLiteHelper.TABLE_NAME, null, " " +
SQLiteHelper.Table_Column_2_Email + "=?", new String[]{EmailHolder}, null, null,
null);

        while (cursor.moveToNext()) {

            if (cursor.isFirst()) {

                cursor.moveToFirst();

                // Storing Password associated with entered email.
                TempPassword =
cursor.getString(cursor.getColumnIndex(SQLiteHelper.Table_Column_3_Password));

                // Closing cursor.
                cursor.close();
            }
        }

        // Calling method to check final result ..
        CheckFinalResult();

    }
    else {

        //If any of login EditText empty then this block will be executed.
        Toast.makeText(Account.this,"Please Enter UserName or
Password.",Toast.LENGTH_LONG).show();

    }

}

// Checking EditText is empty or not.

```

```

public void CheckEditTextStatus(){

    // Getting value from All EditText and storing into String Variables.
    EmailHolder = Email.getText().toString();
    PasswordHolder = Password.getText().toString();

    // Checking EditText is empty or no using TextUtils.
    if( TextUtils.isEmpty(EmailHolder) || TextUtils.isEmpty>PasswordHolder)){

        EditTextEmptyHolder = false ;

    }
    else {

        EditTextEmptyHolder = true ;

    }
}

// Checking entered password from SQLite database email associated password.
public void CheckFinalResult(){

    if(TempPassword.equalsIgnoreCase>PasswordHolder))
    {

        Toast.makeText(Account.this,"Login Successfully",Toast.LENGTH_LONG).show();

        // Going to Dashboard activity after login success message.
        Intent intent = new Intent(Account.this, DashboardActivity.class);

        // Sending Email to Dashboard Activity using intent.
        intent.putExtra(UserEmail, EmailHolder);

        startActivity(intent);

    }
    else {

        Toast.makeText(Account.this,"UserName or Password is Wrong, Please Try Again.",Toast.LENGTH_LONG).show();

    }
    TempPassword = "NOT_FOUND" ;

}

}

```

## DashboardActivity.java

```
package com.akash.chefhour.view;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import com.akash.chefhour.R;

public class DashboardActivity extends AppCompatActivity {

    String EmailHolder;
    TextView Email;
    Button LogOUT ;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dashboard);

        Email = (TextView) findViewById(R.id.textView2);
        LogOUT = (Button) findViewById(R.id.button1);

        Intent intent = getIntent();

        // Receiving User Email Send By MainActivity.
        EmailHolder = intent.getStringExtra(Account.USER_EMAIL);

        // Setting up received email to TextView.
        Email.setText(Email.getText().toString() + EmailHolder);

        // Adding click listener to Log Out button.
        LogOUT.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                //Finishing current DashBoard activity on button click.
                finish();

                Toast.makeText(DashboardActivity.this, "Log Out Successfull",
                    Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

```
}  
}
```

## RegisterActivity.java

```
package com.akash.chefhour.view;  
  
import android.content.Context;  
import android.content.Intent;  
import android.database.Cursor;  
import android.database.sqlite.SQLiteDatabase;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.akash.chefhour.R;  
  
public class RegisterActivity extends AppCompatActivity {  
  
    EditText Email, Password, Name ;  
    Button Register;  
    String NameHolder, EmailHolder, PasswordHolder;  
    Boolean EditTextEmptyHolder;  
    SQLiteDatabase sqLiteDatabaseObj;  
    String SQLiteDataBaseQueryHolder ;  
    SQLiteHelper sqLiteHelper;  
    Cursor cursor;  
    String F_Result = "Not_Found";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_register);  
  
        Register = (Button)findViewById(R.id.buttonRegister);  
  
        Email = (EditText)findViewById(R.id.editEmail);  
        Password = (EditText)findViewById(R.id.editPassword);  
        Name = (EditText)findViewById(R.id.editName);  
  
        sqLiteHelper = new SQLiteHelper(this);  
  
        // Adding click listener to register button.
```

```

Register.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // Creating SQLite database if dosen't exists
        SQLiteDataBaseBuild();

        // Creating SQLite table if dosen't exists.
        SQLiteTableBuild();

        // Checking EditText is empty or Not.
        CheckEditTextStatus();

        // Method to check Email is already exists or not.
        CheckingEmailAlreadyExistsOrNot();

        // Empty EditText After done inserting process.
        EmptyEditTextAfterDataInsert();

    }
});

}

// SQLite database build method.
public void SQLiteDataBaseBuild(){

    sqLiteDatabaseObj = openOrCreateDatabase(SQLiteHelper.DATABASE_NAME,
Context.MODE_PRIVATE, null);

}

// SQLite table build method.
public void SQLiteTableBuild() {

    sqLiteDatabaseObj.execSQL("CREATE TABLE IF NOT EXISTS " +
SQLiteHelper.TABLE_NAME + "(" + SQLiteHelper.Table_Column_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT NOT NULL, " + SQLiteHelper.Table_Column_1_Name + " VARCHAR, " +
SQLiteHelper.Table_Column_2_Email + " VARCHAR, " +
SQLiteHelper.Table_Column_3_Password + " VARCHAR);");

}

// Insert data into SQLite database method.
public void InsertDataIntoSQLiteDatabase(){

    // If editText is not empty then this block will executed.
    if(EditTextEmptyHolder == true)
    {

```

```

        // SQLite query to insert data into table.
        SQLiteDatabaseQueryHolder = "INSERT INTO "+SQLiteHelper.TABLE_NAME+"
(name,email,password)          VALUES ('"+NameHolder+"',          '"+EmailHolder+"',
 '"+PasswordHolder+"');"

        // Executing query.
        sqLiteDatabaseObj.execSQL(SQLiteDatabaseQueryHolder);

        // Closing SQLite database object.
        sqLiteDatabaseObj.close();

        // Printing toast message after done inserting.
        Toast.makeText(RegisterActivity.this,"User Registered Successfully",
Toast.LENGTH_LONG).show();

    }

    // This block will execute if any of the registration EditText is empty.
    else {

        // Printing toast message if any of EditText is empty.
        Toast.makeText(RegisterActivity.this,"Please Fill All The Required
Fields.", Toast.LENGTH_LONG).show();

    }

}

// Empty edittext after done inserting process method.
public void EmptyEditTextAfterDataInsert(){

    Name.getText().clear();

    Email.getText().clear();

    Password.getText().clear();

}

// Method to check EditText is empty or Not.
public void CheckEditTextStatus(){

    // Getting value from All EditText and storing into String Variables.
    NameHolder = Name.getText().toString();
    EmailHolder = Email.getText().toString();
    PasswordHolder = Password.getText().toString();

    if(TextUtils.isEmpty(NameHolder) || TextUtils.isEmpty(EmailHolder) ||
TextUtils.isEmpty(PasswordHolder)){

```

```

        EditTextEmptyHolder = false ;

    }
    else {

        EditTextEmptyHolder = true ;
    }
}

// Checking Email is already exists or not.
public void CheckingEmailAlreadyExistsOrNot(){

    // Opening SQLite database write permission.
    sqLiteDatabaseObj = sqLiteHelper.getWritableDatabase();

    // Adding search email query to cursor.
    cursor = sqLiteDatabaseObj.query(SQLiteHelper.TABLE_NAME, null, " " +
    SQLiteHelper.Table_Column_2_Email + "=?", new String[]{EmailHolder}, null, null,
    null);

    while (cursor.moveToNext()) {

        if (cursor.isFirst()) {

            cursor.moveToFirst();

            // If Email is already exists then Result variable value set as Email
Found.

            F_Result = "Email Found";

            // Closing cursor.
            cursor.close();
        }
    }

    // Calling method to check final result and insert data into SQLite database.
    CheckFinalResult();
}

// Checking result
public void CheckFinalResult(){

    // Checking whether email is already exists or not.
    if(F_Result.equalsIgnoreCase("Email Found"))
    {

        // If email is exists then toast msg will display.

```

```

Toast.makeText(RegisterActivity.this, "Email Already
Exists", Toast.LENGTH_LONG).show();

    }
    else {

        // If email already dose n't exists then user registration details will
        entered to SQLite database.
        InsertDataIntoSQLiteDatabase();

    }

    F_Result = "Not_Found" ;

}
}

```

## SQLiteHelper.java

```

package com.akash.chefhour.view;

import android.content.Context;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase;

public class SQLiteHelper extends SQLiteOpenHelper {

    static String DATABASE_NAME="UserDataBase";

    public static final String TABLE_NAME="UserTable";

    public static final String Table_Column_ID="id";

    public static final String Table_Column_1_Name="name";

    public static final String Table_Column_2_Email="email";

    public static final String Table_Column_3_Password="password";

    public SQLiteHelper(Context context) {

        super(context, DATABASE_NAME, null, 1);

    }

    @Override
    public void onCreate(SQLiteDatabase database) {

```



```

        String CREATE_TABLE="CREATE TABLE IF NOT EXISTS "+TABLE_NAME+"
("+Table_Column_ID+" INTEGER PRIMARY KEY, "+Table_Column_1_Name+" VARCHAR,
"+Table_Column_2_Email+" VARCHAR, "+Table_Column_3_Password+" VARCHAR)";
        database.execSQL(CREATE_TABLE);

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
        onCreate(db);
    }
}

```

## User.java

```

/*-----
- Developed by akash Muttaqin -
- Last modified 14/5/20 2:38 AM -
- Subscribe : https://www.youtube.com/akashmuttaqin -
- Copyright (c) 2020. All rights reserved -
-----*/

package com.akash.chefhour.view;

public class User {
    //variables
    int id;
    String name;
    String password;
    // Constructor with two parameters name and password
    public User(String name,String password)
    {
        this.name=name;
        this.password=password;
    }
    //Parameter constructor containing all three parameters
    public User(int id,String name,String psd)
    {
        this.id=id;
        this.name=name;
        this.password=psd;
    }
    //getting id
    public int getId() {
        return id;
    }
}

```

```

//setting id
public void setId(int id) {
    this.id = id;
}
//getting name
public String getName() {
    return name;
}
//setting name
public void setName(String name) {
    this.name = name;
}
//getting password
public String getPassword() {
    return password;
}
//setting password
public void setPassword(String password) {
    this.password = password;
}
}

```

## Utils.java

```

package com.akash.chefhour;

import android.app.AlertDialog;
import android.content.Context;

import com.akash.chefhour.api.FoodApi;
import com.akash.chefhour.api.FoodClient;

public class Utils {

    public static FoodApi getApi() {
        return FoodClient.getFoodClient().create(FoodApi.class);
    }

    public static AlertDialog showDialogMessage(Context context, String title, String
message) {
        AlertDialog alertDialog = new
AlertDialog.Builder(context).setTitle(title).setMessage(message).show();
        if (alertDialog.isShowing()) {
            alertDialog.cancel();
        }
        return alertDialog;
    }
}

```

## MODELS

### Categories.java

```
package com.akash.chefhour.model;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

import java.io.Serializable;
import java.util.List;

public class Categories implements Serializable {

    @SerializedName("categories")
    @Expose
    private List<Category> categories;

    public List<Category> getCategories() {
        return categories;
    }

    public void setCategories(List<Category> categories) {
        this.categories = categories;
    }

    public static class Category implements Serializable {

        @SerializedName("idCategory")
        @Expose
        private String idCategory;
        @SerializedName("strCategory")
        @Expose
        private String strCategory;
        @SerializedName("strCategoryThumb")
        @Expose
        private String strCategoryThumb;
        @SerializedName("strCategoryDescription")
        @Expose
        private String strCategoryDescription;

        public String getIdCategory() {
            return idCategory;
        }

        public void setIdCategory(String idCategory) {
            this.idCategory = idCategory;
        }
    }
}
```

```

    public String getStrCategory() {
        return strCategory;
    }

    public void setStrCategory(String strCategory) {
        this.strCategory = strCategory;
    }

    public String getStrCategoryThumb() {
        return strCategoryThumb;
    }

    public void setStrCategoryThumb(String strCategoryThumb) {
        this.strCategoryThumb = strCategoryThumb;
    }

    public String getStrCategoryDescription() {
        return strCategoryDescription;
    }

    public void setStrCategoryDescription(String strCategoryDescription) {
        this.strCategoryDescription = strCategoryDescription;
    }
}
}

```

## Meals.java

```

package com.akash.chefhour.model;

import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

import java.io.Serializable;
import java.util.List;

public class Meals {

    @SerializedName("meals")
    @Expose
    private List<Meal> meals = null;
    public List<Meal> getMeals() {
        return meals;
    }
    public void setMeals(List<Meal> meals) {

```

```

        this.meals = meals;
    }

    public class Meal {

        @SerializedName("idMeal")
        @Expose
        private String idMeal;
        @SerializedName("strMeal")
        @Expose
        private String strMeal;
        @SerializedName("strDrinkAlternate")
        @Expose
        private Object strDrinkAlternate;
        @SerializedName("strCategory")
        @Expose
        private String strCategory;
        @SerializedName("strArea")
        @Expose
        private String strArea;
        @SerializedName("strInstructions")
        @Expose
        private String strInstructions;
        @SerializedName("strMealThumb")
        @Expose
        private String strMealThumb;
        @SerializedName("strTags")
        @Expose
        private String strTags;
        @SerializedName("strYoutube")
        @Expose
        private String strYoutube;
        @SerializedName("strIngredient1")
        @Expose
        private String strIngredient1;
        @SerializedName("strIngredient2")
        @Expose
        private String strIngredient2;
        @SerializedName("strIngredient3")
        @Expose
        private String strIngredient3;
        @SerializedName("strIngredient4")
        @Expose
        private String strIngredient4;
        @SerializedName("strIngredient5")
        @Expose
        private String strIngredient5;
        @SerializedName("strIngredient6")
        @Expose
        private String strIngredient6;
    }

```

```

@SerializedName("strIngredient7")
@Expose
private String strIngredient7;
@SerializedName("strIngredient8")
@Expose
private String strIngredient8;
@SerializedName("strIngredient9")
@Expose
private String strIngredient9;
@SerializedName("strIngredient10")
@Expose
private String strIngredient10;
@SerializedName("strIngredient11")
@Expose
private String strIngredient11;
@SerializedName("strIngredient12")
@Expose
private String strIngredient12;
@SerializedName("strIngredient13")
@Expose
private String strIngredient13;
@SerializedName("strIngredient14")
@Expose
private String strIngredient14;
@SerializedName("strIngredient15")
@Expose
private String strIngredient15;
@SerializedName("strIngredient16")
@Expose
private String strIngredient16;
@SerializedName("strIngredient17")
@Expose
private String strIngredient17;
@SerializedName("strIngredient18")
@Expose
private String strIngredient18;
@SerializedName("strIngredient19")
@Expose
private String strIngredient19;
@SerializedName("strIngredient20")
@Expose
private String strIngredient20;
@SerializedName("strMeasure1")
@Expose
private String strMeasure1;
@SerializedName("strMeasure2")
@Expose
private String strMeasure2;
@SerializedName("strMeasure3")
@Expose

```

```

private String strMeasure3;
@SerializedName("strMeasure4")
@Expose
private String strMeasure4;
@SerializedName("strMeasure5")
@Expose
private String strMeasure5;
@SerializedName("strMeasure6")
@Expose
private String strMeasure6;
@SerializedName("strMeasure7")
@Expose
private String strMeasure7;
@SerializedName("strMeasure8")
@Expose
private String strMeasure8;
@SerializedName("strMeasure9")
@Expose
private String strMeasure9;
@SerializedName("strMeasure10")
@Expose
private String strMeasure10;
@SerializedName("strMeasure11")
@Expose
private String strMeasure11;
@SerializedName("strMeasure12")
@Expose
private String strMeasure12;
@SerializedName("strMeasure13")
@Expose
private String strMeasure13;
@SerializedName("strMeasure14")
@Expose
private String strMeasure14;
@SerializedName("strMeasure15")
@Expose
private String strMeasure15;
@SerializedName("strMeasure16")
@Expose
private String strMeasure16;
@SerializedName("strMeasure17")
@Expose
private String strMeasure17;
@SerializedName("strMeasure18")
@Expose
private String strMeasure18;
@SerializedName("strMeasure19")
@Expose
private String strMeasure19;
@SerializedName("strMeasure20")

```

```

@Expose
private String strMeasure20;
@SerializedName("strSource")
@Expose
private String strSource;
@SerializedName("dateModified")
@Expose
private Object dateModified;

public String getIdMeal() {
    return idMeal;
}

public void setIdMeal(String idMeal) {
    this.idMeal = idMeal;
}

public String getStrMeal() {
    return strMeal;
}

public void setStrMeal(String strMeal) {
    this.strMeal = strMeal;
}

public Object getStrDrinkAlternate() {
    return strDrinkAlternate;
}

public void setStrDrinkAlternate(Object strDrinkAlternate) {
    this.strDrinkAlternate = strDrinkAlternate;
}

public String getStrCategory() {
    return strCategory;
}

public void setStrCategory(String strCategory) {
    this.strCategory = strCategory;
}

public String getStrArea() {
    return strArea;
}

public void setStrArea(String strArea) {
    this.strArea = strArea;
}

public String getStrInstructions() {

```



```

        return strInstructions;
    }

    public void setStrInstructions(String strInstructions) {
        this.strInstructions = strInstructions;
    }

    public String getStrMealThumb() {
        return strMealThumb;
    }

    public void setStrMealThumb(String strMealThumb) {
        this.strMealThumb = strMealThumb;
    }

    public String getStrTags() {
        return strTags;
    }

    public void setStrTags(String strTags) {
        this.strTags = strTags;
    }

    public String getStrYoutube() {
        return strYoutube;
    }

    public void setStrYoutube(String strYoutube) {
        this.strYoutube = strYoutube;
    }

    public String getStrIngredient1() {
        return strIngredient1;
    }

    public void setStrIngredient1(String strIngredient1) {
        this.strIngredient1 = strIngredient1;
    }

    public String getStrIngredient2() {
        return strIngredient2;
    }

    public void setStrIngredient2(String strIngredient2) {
        this.strIngredient2 = strIngredient2;
    }

    public String getStrIngredient3() {
        return strIngredient3;
    }

```

```

public void setStrIngredient3(String strIngredient3) {
    this.strIngredient3 = strIngredient3;
}

public String getStrIngredient4() {
    return strIngredient4;
}

public void setStrIngredient4(String strIngredient4) {
    this.strIngredient4 = strIngredient4;
}

public String getStrIngredient5() {
    return strIngredient5;
}

public void setStrIngredient5(String strIngredient5) {
    this.strIngredient5 = strIngredient5;
}

public String getStrIngredient6() {
    return strIngredient6;
}

public void setStrIngredient6(String strIngredient6) {
    this.strIngredient6 = strIngredient6;
}

public String getStrIngredient7() {
    return strIngredient7;
}

public void setStrIngredient7(String strIngredient7) {
    this.strIngredient7 = strIngredient7;
}

public String getStrIngredient8() {
    return strIngredient8;
}

public void setStrIngredient8(String strIngredient8) {
    this.strIngredient8 = strIngredient8;
}

public String getStrIngredient9() {
    return strIngredient9;
}

public void setStrIngredient9(String strIngredient9) {

```

```

        this.strIngredient9 = strIngredient9;
    }

    public String getStrIngredient10() {
        return strIngredient10;
    }

    public void setStrIngredient10(String strIngredient10) {
        this.strIngredient10 = strIngredient10;
    }

    public String getStrIngredient11() {
        return strIngredient11;
    }

    public void setStrIngredient11(String strIngredient11) {
        this.strIngredient11 = strIngredient11;
    }

    public String getStrIngredient12() {
        return strIngredient12;
    }

    public void setStrIngredient12(String strIngredient12) {
        this.strIngredient12 = strIngredient12;
    }

    public String getStrIngredient13() {
        return strIngredient13;
    }

    public void setStrIngredient13(String strIngredient13) {
        this.strIngredient13 = strIngredient13;
    }

    public String getStrIngredient14() {
        return strIngredient14;
    }

    public void setStrIngredient14(String strIngredient14) {
        this.strIngredient14 = strIngredient14;
    }

    public String getStrIngredient15() {
        return strIngredient15;
    }

    public void setStrIngredient15(String strIngredient15) {
        this.strIngredient15 = strIngredient15;
    }

```

```

public String getStrIngredient16() {
    return strIngredient16;
}

public void setStrIngredient16(String strIngredient16) {
    this.strIngredient16 = strIngredient16;
}

public String getStrIngredient17() {
    return strIngredient17;
}

public void setStrIngredient17(String strIngredient17) {
    this.strIngredient17 = strIngredient17;
}

public String getStrIngredient18() {
    return strIngredient18;
}

public void setStrIngredient18(String strIngredient18) {
    this.strIngredient18 = strIngredient18;
}

public String getStrIngredient19() {
    return strIngredient19;
}

public void setStrIngredient19(String strIngredient19) {
    this.strIngredient19 = strIngredient19;
}

public String getStrIngredient20() {
    return strIngredient20;
}

public void setStrIngredient20(String strIngredient20) {
    this.strIngredient20 = strIngredient20;
}

public String getStrMeasure1() {
    return strMeasure1;
}

public void setStrMeasure1(String strMeasure1) {
    this.strMeasure1 = strMeasure1;
}

public String getStrMeasure2() {

```

```
        return strMeasure2;
    }

    public void setStrMeasure2(String strMeasure2) {
        this.strMeasure2 = strMeasure2;
    }

    public String getStrMeasure3() {
        return strMeasure3;
    }

    public void setStrMeasure3(String strMeasure3) {
        this.strMeasure3 = strMeasure3;
    }

    public String getStrMeasure4() {
        return strMeasure4;
    }

    public void setStrMeasure4(String strMeasure4) {
        this.strMeasure4 = strMeasure4;
    }

    public String getStrMeasure5() {
        return strMeasure5;
    }

    public void setStrMeasure5(String strMeasure5) {
        this.strMeasure5 = strMeasure5;
    }

    public String getStrMeasure6() {
        return strMeasure6;
    }

    public void setStrMeasure6(String strMeasure6) {
        this.strMeasure6 = strMeasure6;
    }

    public String getStrMeasure7() {
        return strMeasure7;
    }

    public void setStrMeasure7(String strMeasure7) {
        this.strMeasure7 = strMeasure7;
    }

    public String getStrMeasure8() {
        return strMeasure8;
    }
}
```

```

public void setStrMeasure8(String strMeasure8) {
    this.strMeasure8 = strMeasure8;
}

public String getStrMeasure9() {
    return strMeasure9;
}

public void setStrMeasure9(String strMeasure9) {
    this.strMeasure9 = strMeasure9;
}

public String getStrMeasure10() {
    return strMeasure10;
}

public void setStrMeasure10(String strMeasure10) {
    this.strMeasure10 = strMeasure10;
}

public String getStrMeasure11() {
    return strMeasure11;
}

public void setStrMeasure11(String strMeasure11) {
    this.strMeasure11 = strMeasure11;
}

public String getStrMeasure12() {
    return strMeasure12;
}

public void setStrMeasure12(String strMeasure12) {
    this.strMeasure12 = strMeasure12;
}

public String getStrMeasure13() {
    return strMeasure13;
}

public void setStrMeasure13(String strMeasure13) {
    this.strMeasure13 = strMeasure13;
}

public String getStrMeasure14() {
    return strMeasure14;
}

public void setStrMeasure14(String strMeasure14) {

```

```

        this.strMeasure14 = strMeasure14;
    }

    public String getStrMeasure15() {
        return strMeasure15;
    }

    public void setStrMeasure15(String strMeasure15) {
        this.strMeasure15 = strMeasure15;
    }

    public String getStrMeasure16() {
        return strMeasure16;
    }

    public void setStrMeasure16(String strMeasure16) {
        this.strMeasure16 = strMeasure16;
    }

    public String getStrMeasure17() {
        return strMeasure17;
    }

    public void setStrMeasure17(String strMeasure17) {
        this.strMeasure17 = strMeasure17;
    }

    public String getStrMeasure18() {
        return strMeasure18;
    }

    public void setStrMeasure18(String strMeasure18) {
        this.strMeasure18 = strMeasure18;
    }

    public String getStrMeasure19() {
        return strMeasure19;
    }

    public void setStrMeasure19(String strMeasure19) {
        this.strMeasure19 = strMeasure19;
    }

    public String getStrMeasure20() {
        return strMeasure20;
    }

    public void setStrMeasure20(String strMeasure20) {
        this.strMeasure20 = strMeasure20;
    }

```

```
public String getStrSource() {  
    return strSource;  
}  
  
public void setStrSource(String strSource) {  
    this.strSource = strSource;  
}  
  
public Object getDateModified() {  
    return dateModified;  
}  
  
public void setDateModified(Object dateModified) {  
    this.dateModified = dateModified;  
}  
}  
}
```



## ADAPTERS

### RecyclerViewHomeAdapter.java

```
package com.akash.chefhour.adapter;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.akash.chefhour.R;
import com.akash.chefhour.model.Categories;
import com.squareup.picasso.Picasso;

import java.util.List;

import butterknife.BindView;
import butterknife.ButterKnife;

public class RecyclerViewHomeAdapter extends
RecyclerView.Adapter<RecyclerViewHomeAdapter.RecyclerViewHolder> {

    private List<Categories.Category> categories;
    private Context context;
    private static ClickListener clickListener;

    public RecyclerViewHomeAdapter(List<Categories.Category> categories, Context
context) {
        this.categories = categories;
        this.context = context;
    }

    @NonNull
    @Override
    public RecyclerViewHomeAdapter.RecyclerViewHolder onCreateViewHolder(@NonNull
ViewGroup viewGroup, int i) {
        View view =
LayoutInflater.from(context).inflate(R.layout.item_recycler_category,
viewGroup, false);
        return new RecyclerViewHolder(view);
    }

    @Override
```

```

        public void onBindViewHolder(@NonNull RecyclerViewHomeAdapter.RecyclerViewHolder
viewHolder, int i) {

            String strCategoryThum = categories.get(i).getStrCategoryThumb();

            Picasso.get().load(strCategoryThum).placeholder(R.drawable.ic_circle).into(viewHolder.
categoryThumb);

            String strCategoryName = categories.get(i).getStrCategory();
            viewHolder.categoryName.setText(strCategoryName);
        }

        @Override
        public int getItemCount() {
            return categories.size();
        }

        static class RecyclerViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {
            @BindView(R.id.categoryThumb)
            ImageView categoryThumb;
            @BindView(R.id.categoryName)
            TextView categoryName;
            RecyclerViewHolder(@NonNull View itemView) {
                super(itemView);
                ButterKnife.bind(this, itemView);
                itemView.setOnClickListener(this);
            }

            @Override
            public void onClick(View v) {
                clickListener.onClick(v, getAdapterPosition());
            }
        }

        public void setOnItemClickListener(ClickListener clickListener) {
            RecyclerViewHomeAdapter.clickListener = clickListener;
        }

        public interface ClickListener {
            void onClick(View view, int position);
        }
    }
}

```

## RecyclerViewMealByCategory.java

```
package com.akash.chefhour.adapter;

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.akash.chefhour.R;
import com.akash.chefhour.model.Meals;
import com.squareup.picasso.Picasso;

import java.util.List;

import butterknife.BindView;
import butterknife.ButterKnife;

public class RecyclerViewMealByCategory extends
RecyclerView.Adapter<RecyclerViewMealByCategory.RecyclerViewHolder> {

    private List<Meals.Meal> meals;
    private Context context;
    private static ClickListener clickListener;

    public RecyclerViewMealByCategory(Context context, List<Meals.Meal> meals) {
        this.meals = meals;
        this.context = context;
    }

    @NonNull
    @Override
    public RecyclerViewMealByCategory.RecyclerViewHolder onCreateViewHolder(@NonNull
ViewGroup viewGroup, int i) {
        View view = LayoutInflater.from(context).inflate(R.layout.item_recycler_meal,
viewGroup, false);
        return new RecyclerViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull RecyclerViewMealByCategory.RecyclerViewHolder
viewHolder, int i) {

        String strMealThumb = meals.get(i).getStrMealThumb();
```

```

Picasso.get().load(strMealThumb).placeholder(R.drawable.shadow_bottom_to_top).into(viewHolder.mealThumb);

        String strMealName = meals.get(i).getStrMeal();
        viewHolder.mealName.setText(strMealName);
    }

    @Override
    public int getItemCount() {
        return meals.size();
    }

    static class RecyclerViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener {
        @BindView(R.id.mealThumb)
        ImageView mealThumb;
        @BindView(R.id.mealName)
        TextView mealName;
        RecyclerViewHolder(@NonNull View itemView) {
            super(itemView);
            ButterKnife.bind(this, itemView);
            itemView.setOnClickListener(this);
        }

        @Override
        public void onClick(View v) {
            clickListener.onClick(v, getAdapterPosition());
        }
    }

    public void setOnItemClickListener(ClickListener clickListener) {
        RecyclerViewMealByCategory.clickListener = clickListener;
    }

    public interface ClickListener {
        void onClick(View view, int position);
    }
}

```

## ViewPagerCatgoryAdapter.java

```

package com.akash.chefhour.adapter;

import android.os.Bundle;

```

```

import android.support.annotation.Nullable;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;

import com.akash.chefhour.model.Categories;
import com.akash.chefhour.view.category.CategoryFragment;

import java.util.List;

public class ViewPagerCategoryAdapter extends FragmentPagerAdapter {

    private List<Categories.Category> categories;

    public ViewPagerCategoryAdapter(FragmentManager fm, List<Categories.Category>
categories) {
        super(fm);
        this.categories = categories;
    }

    @Override
    public Fragment getItem(int i) {
        CategoryFragment fragment = new CategoryFragment();
        Bundle args = new Bundle();
        args.putString("EXTRA_DATA_NAME", categories.get(i).getStrCategory());
        args.putString("EXTRA_DATA_DESC",
categories.get(i).getStrCategoryDescription());
        args.putString("EXTRA_DATA_IMAGE", categories.get(i).getStrCategoryThumb());
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public int getCount() {
        return categories.size();
    }

    @Nullable
    @Override
    public CharSequence getPageTitle(int position) {
        return categories.get(position).getStrCategory();
    }
}

```

## ViewPagerHeaderAdapter.java

```

package com.akash.chefhour.adapter;

```

```

import android.content.Context;
import android.support.annotation.NonNull;
import android.support.v4.view.PagerAdapter;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.akash.chefhour.R;
import com.akash.chefhour.model.Meals;
import com.squareup.picasso.Picasso;

import java.util.List;

public class ViewPagerHeaderAdapter extends PagerAdapter {

    private List<Meals.Meal> meals;
    private Context context;
    private static ClickListener clickListener;

    public ViewPagerHeaderAdapter(List<Meals.Meal> meals, Context context) {
        this.meals = meals;
        this.context = context;
    }

    public void setOnItemClickListener(ClickListener clickListener) {
        ViewPagerHeaderAdapter.clickListener = clickListener;
    }

    @Override
    public int getCount() {
        return meals.size();
    }

    @Override
    public boolean isViewFromObject(@NonNull View view, @NonNull Object o) {
        return view.equals(o);
    }

    @NonNull
    @Override
    public Object instantiateItem(@NonNull ViewGroup container, int position) {
        View view = LayoutInflater.from(context).inflate(
            R.layout.item_view_pager_header,
            container,
            false
        );

        ImageView mealThumb = view.findViewById(R.id.mealThumb);

```

```

        TextView mealName = view.findViewById(R.id.mealName);

        String strMealThumb = meals.get(position).getStrMealThumb();
        Picasso.get().load(strMealThumb).into(mealThumb);

        String strMealName = meals.get(position).getStrMeal();
        mealName.setText(strMealName);

        view.setOnClickListener(v -> clickListener.onClick(v, position));

        container.addView(view, 0);
        return view;
    }

    @Override
    public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object
object) {
        container.removeView((View) object);
    }

    public interface ClickListener {
        void onClick(View v, int position);
    }
}

```

## API Java Files

### FoodApi.java

```
package com.akash.chefhour.api;

import com.akash.chefhour.model.Categories;
import com.akash.chefhour.model.Meals;

import retrofit2.Call;
import retrofit2.http.GET;
import retrofit2.http.Query;

public interface FoodApi {

    @GET("latest.php")
    Call<Meals> getMeal();

    @GET("categories.php")
    Call<Categories> getCategories();

    @GET("filter.php")
    Call<Meals> getMealByCategory(@Query("c") String category);

    @GET("search.php")
    Call<Meals> getMealByName(@Query("s") String mealName);
}
```

### FoodClient.java

```
package com.akash.chefhour.api;

import java.util.concurrent.TimeUnit;

import okhttp3.Interceptor;
import okhttp3.OkHttpClient;
import okhttp3.logging.HttpLoggingInterceptor;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class FoodClient {

    private static final String BASE_URL = "https://www.themealdb.com/api/json/v1/1/";

    public static Retrofit getFoodClient() {
        return new Retrofit.Builder().baseUrl(BASE_URL)
            .client(provideOkHttp())
    }
```



```

        .addConverterFactory(GsonConverterFactory.create())
        .build();
    }

    private static Interceptor provideLoggingInterceptor() {
        return new
        HttpLoggingInterceptor().setLevel(HttpLoggingInterceptor.Level.BODY);
    }

    private static OkHttpClient provideOkHttp() {
        return new OkHttpClient.Builder()
            .connectTimeout(30, TimeUnit.SECONDS)
            .writeTimeout(30, TimeUnit.SECONDS)
            .readTimeout(30, TimeUnit.SECONDS)
            .addNetworkInterceptor(provideLoggingInterceptor())
            .build();
    }
}

```

# **UNIT VIII**

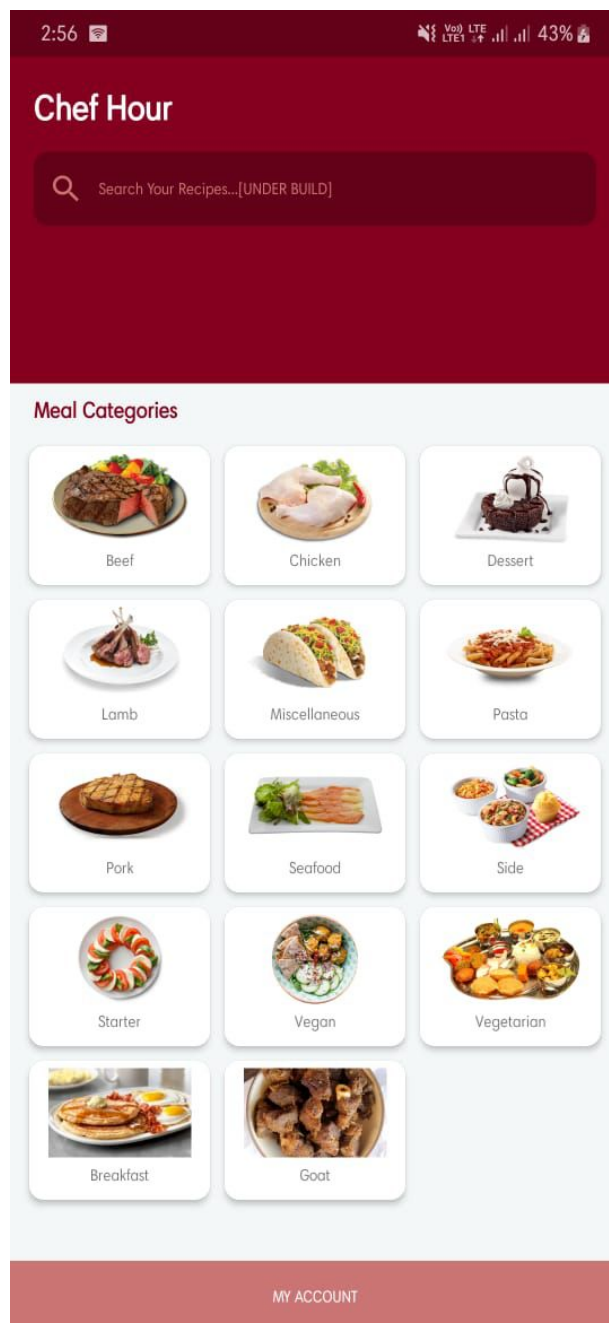
## **RESULT, STORYBOARD & FUTURE ADDITIONS**

### **Conclusions**


This application is extremely handy and useful for cooking a variety of recipes with minimum search effort from the internet. Internet recipes may appeal to one but not to another, some people like experimenting and trying their own recipe. This app also provides the ability for the user to create a recipe and save it for later reference. It will help people to save their time and energy in finding recipes for daily routine as well as for special occasions. And since this is a mobile application, users have the luxury to check for recipes wherever they are and save them for later. The application can be used by a broad range of users which may include parents trying to cook new recipes for kids, people who are fond of desserts, restaurant owners trying to add new items to their menu and for regular cooking. It will help to make the lives of people simpler

# STORYBOARD

## Home page



## User Registration



A screenshot of a mobile application interface for user registration. The background is a textured, reddish-pink watercolor. At the top, a dark red status bar shows the time 2:59, signal strength, LTE, and 43% battery. Below the status bar, the title "User Registration" is centered in a white sans-serif font. The form consists of three input fields with thin white borders. The first field contains the name "Akash". The second field contains the email address "akashravi2209@gmail.com". The third field contains a password represented by six dots. Below the input fields is a light gray rectangular button with the text "REGISTER FROM HERE" in a small, dark gray, all-caps sans-serif font. The bottom half of the screen is mostly obscured by the textured background.

2:59

User Registration

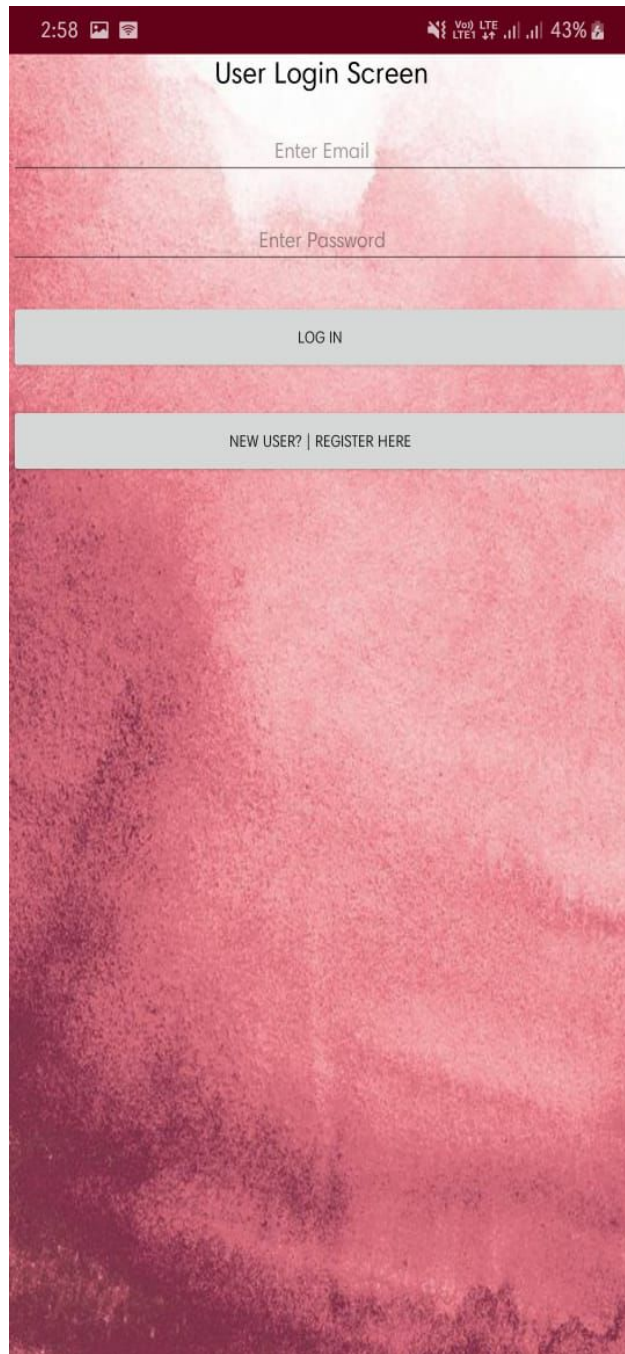
Akash

akashravi2209@gmail.com

.....

REGISTER FROM HERE

## Login Page



The image shows a mobile application interface for a login screen. At the top, there is a dark red status bar with the time 2:58, signal strength icons, and a 43% battery level. Below the status bar, the title "User Login Screen" is centered in a white font. The background of the app is a light pink with a subtle, abstract pattern. There are two input fields: "Enter Email" and "Enter Password", both with light gray borders and placeholder text. Below the password field is a "LOG IN" button with a light gray background and dark text. At the bottom, there is a "NEW USER? | REGISTER HERE" link with a light gray background and dark text.

2:58

User Login Screen

Enter Email

Enter Password

LOG IN

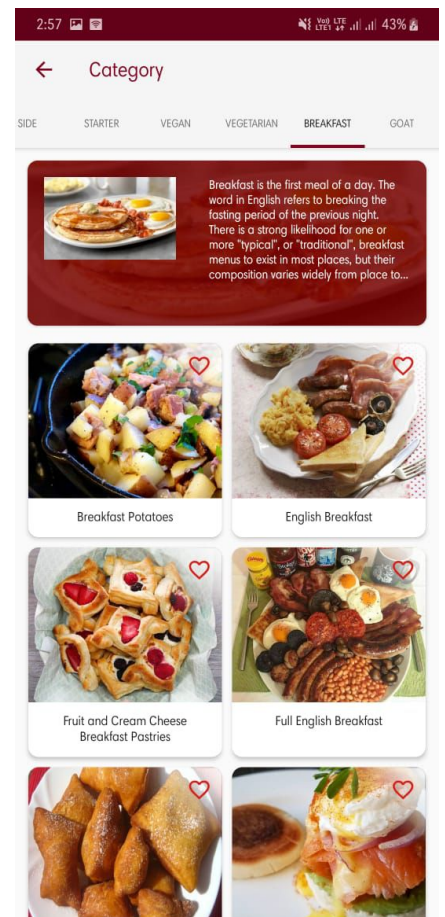
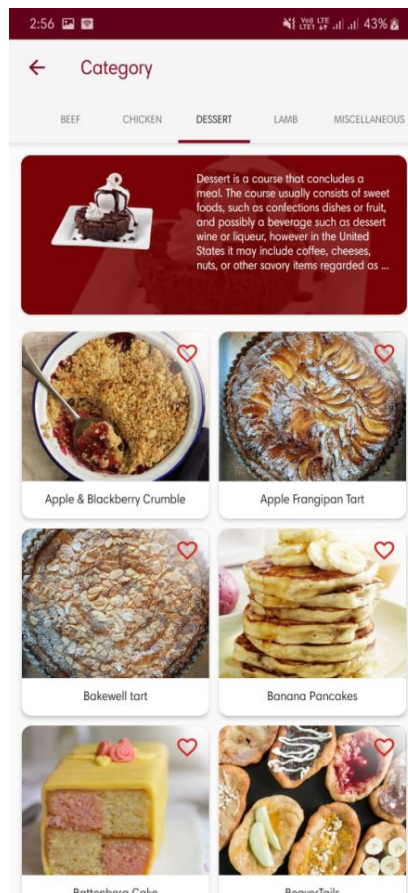
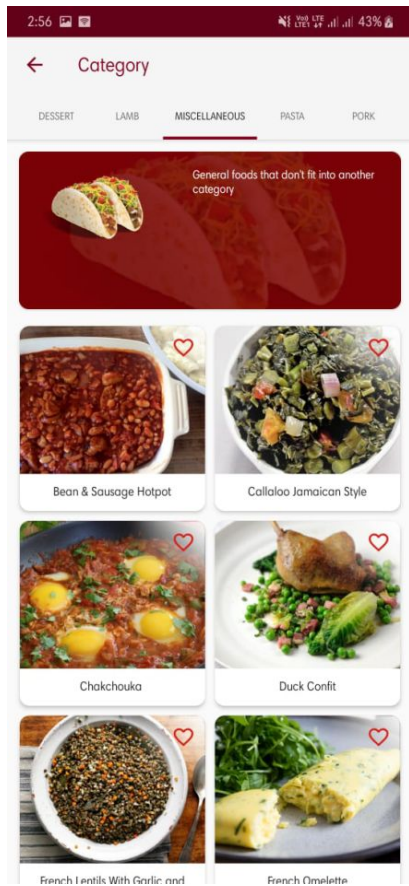
NEW USER? | REGISTER HERE




## Successful login








## Choosing different categories



2:57




Vol: LTE
LTE
43%



# Lasagne

Category  
Pasta

Country  
Italian

## INSTRUCTIONS

Heat the oil in a large saucepan. Use kitchen scissors to snip the bacon into small pieces, or use a sharp knife to chop it on a chopping board. Add the bacon to the pan and cook for just a few mins until starting to turn golden. Add the onion, celery and carrot, and cook over a medium heat for 5 mins, stirring occasionally, until softened. Add the garlic and cook for 1 min, then tip in the mince and cook, stirring and breaking it up with a wooden spoon, for about 6 mins until browned all over.

Stir in the tomato purée and cook for 1 min, mixing in well with the beef and vegetables. Tip in the chopped tomatoes. Fill each can half full with water to rinse out any tomatoes left in the can, and add to the pan. Add the honey and season to taste. Simmer for 20 mins.

Heat oven to 200C/180C fan/gas 6. To assemble the lasagne, ladle a little of the ragu sauce into the bottom of the roasting tin or casserole dish, spreading the sauce all over the base. Place 2 sheets of lasagne on top of the sauce overlapping to make it fit, then repeat with more sauce and another layer of pasta. Repeat with a further 2 layers of sauce and pasta, finishing with a layer of pasta.

Put the crème fraîche in a bowl and mix with 2 tbsp water to loosen it and make a smooth pourable sauce. Pour this over the top of the pasta, then top with the mozzarella. Sprinkle Parmesan over the top and bake for 25–30 mins until golden and bubbling. Serve scattered with basil, if you like.

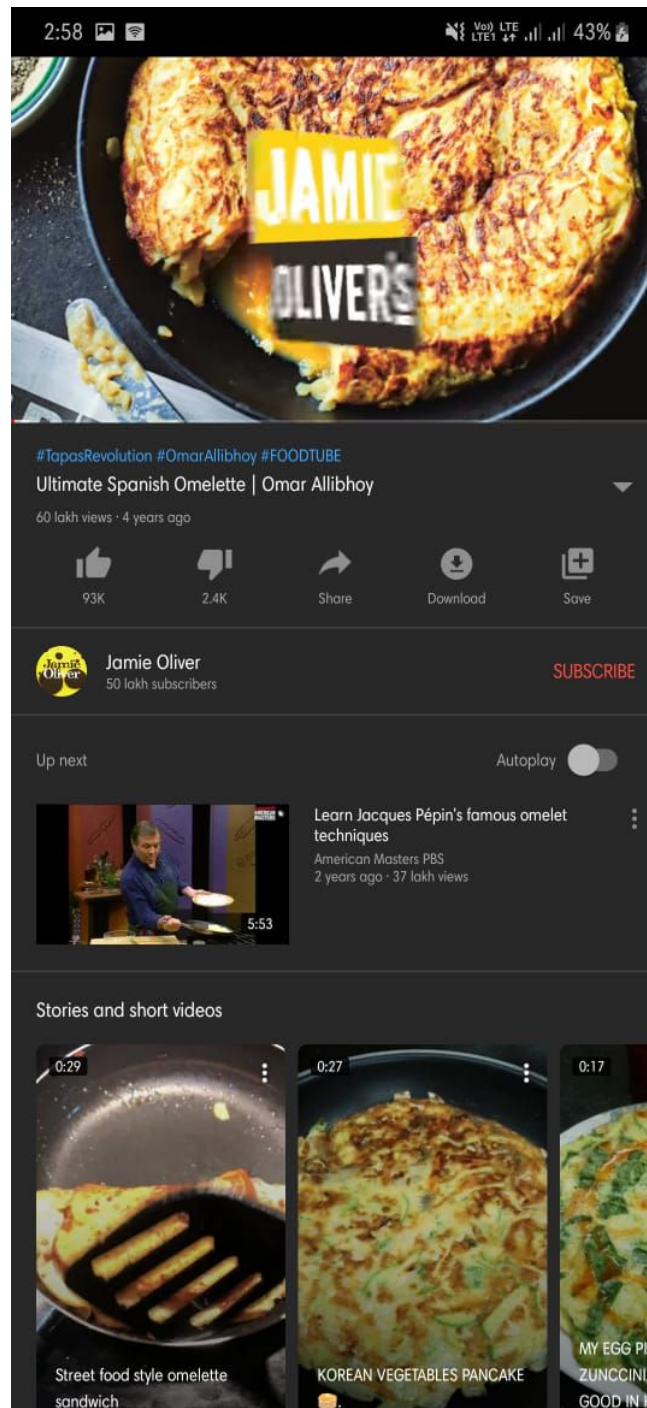
## INGREDIENTS

- Olive Oil : 1 tblsp
- Bacon : 2
- Onion : 1 finely chopped
- Celery : 1 stick
- Carrot : 1
- Garlic : 2 cloves
- Beef Mince : 400g
- Tomato Purée : 1 x 400g can
- Chopped Tomatoes : 2 x 400g cans
- Water : 2 x 400g cans
- Honey : 1 tsp
- Salt : to taste
- Lasagne Sheets : 10
- Mozzarella : 100g
- Parmesan : 50g
- Basil : to garnish

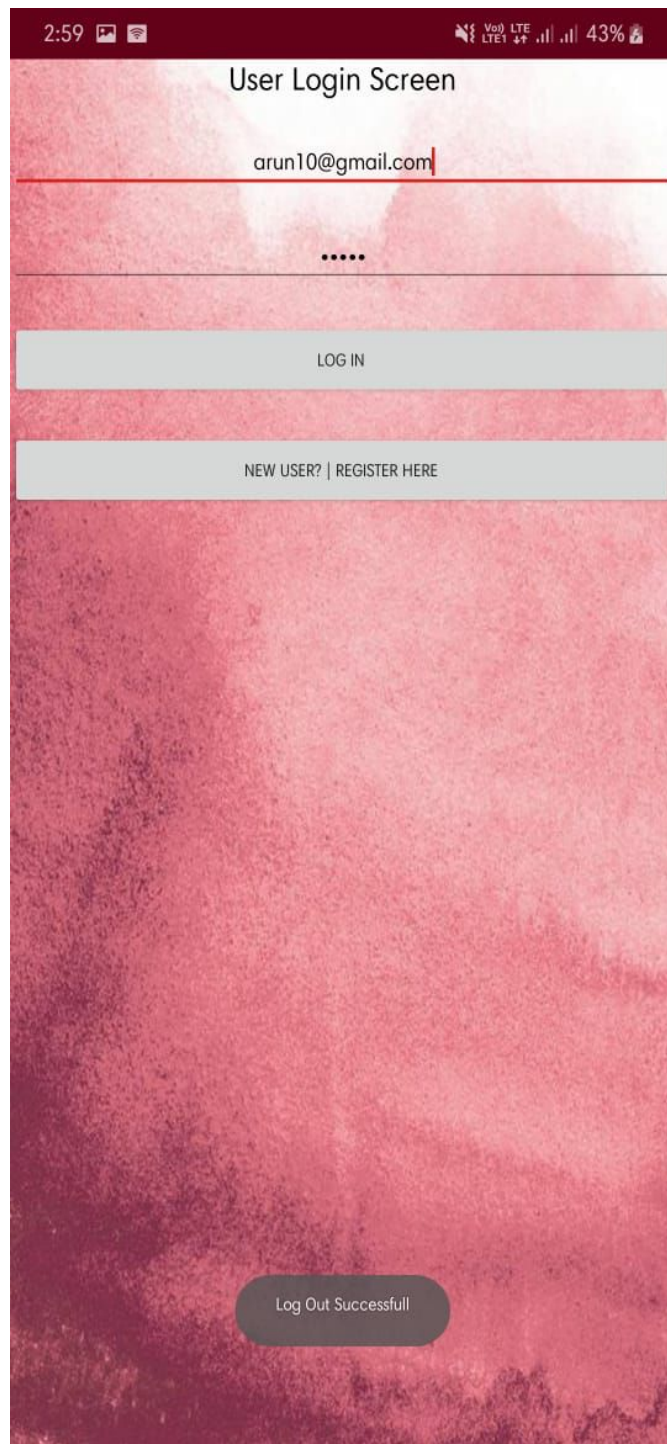




## On clicking Youtube option



## Logout



# FUTURE ADDITIONS

**The recipe search can be done in different ways:**

- Search by Title and Ingredients
- Search on Recipe ratings
- Search Trending Recipes in different categories
- Search by category

## **MyRecipeList**

Users have the option to manage the cookbook by creating recipes, viewing recipes and deleting recipes.

### ***Create Recipe:***

User can create and manage his personal cookbook on phone, tablet. When user clicks personal cookbook image button on home page, user gets navigated to new screen. This new screen provides three image button for creating, viewing and deleting recipe respectively. On clicking, click image to add recipes button on personal cookbook screen, new activity screen template is displayed. Users can create recipes by filling details in template fields and clicking on the save button. When the user clicks the save button, a new row gets added in the database for the created recipe.

### ***View and Modify Recipes:***

Application save recipes added by user in database providing capability to view saved recipes when required. On clicking, view recipes imagebutton on personal recipe screen, list of user created recipes are displayed with recipe image and recipe title.

If a recipe image is not provided by the user during creation of recipe default image gets displayed with message no image available. Users can select a recipe from the list by clicking on name. Upon clicking the recipe title, time required for cooking, ingredients needed, cooking directions and recipe image are displayed. You can edit these.

### ***Delete Recipes:***

User has a delete option in the personal recipe screen. Users can select a recipe from the list, on selection alertdialog box is displayed with message deleting recipe. User can select yes button for deletion, and no to cancel delete operation. When user clicks yes, recipe gets deleted and page is refreshed. Upon clicking no, user is navigated to recipe list screen

We have planned to include a Restaurant locator as well in the Application. This is for the people to see the restaurants in their cities and rate them. It can be also used in knowing how good the restaurant is, as well as how much is the expense, etc. Further we hope to integrate a food delivery system into this App. Hence making the App a complete place for a foody to enjoy the feast.

# UNIT XI

## REFERENCES

- [\*http://github.com/facebook/shimmer-android\*](http://github.com/facebook/shimmer-android)
- [\*https://www.tutorialspoint.com/sqlite/sqlite\\_overview.htm\*](https://www.tutorialspoint.com/sqlite/sqlite_overview.htm)
- [\*https://www.sqlite.org/index.html\*](https://www.sqlite.org/index.html)
- [\*https://github.com/haerulmuttaqin/FoodsApp-starting-code\*](https://github.com/haerulmuttaqin/FoodsApp-starting-code)
- [\*https://square.github.io/retrofit/\*](https://square.github.io/retrofit/)
- [\*http://www.jsonschema2pojo.org/\*](http://www.jsonschema2pojo.org/)
- [\*https://www.themealdb.com/api.php\*](https://www.themealdb.com/api.php)
- [\*https://www.themealdb.com/api/json/v2/1/latest.php\*](https://www.themealdb.com/api/json/v2/1/latest.php)
- [\*https://abhiandroid.com/\*](https://abhiandroid.com/)
- [\*https://www.youtube.com/watch?v=njTHtyzaBug\*](https://www.youtube.com/watch?v=njTHtyzaBug)
- [\*https://www.tutorialspoint.com/android/android\\_login\\_screen.htm\*](https://www.tutorialspoint.com/android/android_login_screen.htm)