Code Implementation details

-> Implemented backend service calls with **REST** services and captured the web services responses in **RxJava** callbacks.

-> Implemented complete code with **Kotlin**

-> Implemented UI with **ConstraintLayout** , which allows it to fit for all kinds of screen densities.

-> Used **RecyclerViews** to render the list items.

-> Implemented **Picasa** library to render images in Application.

-> Implemented functionality with all **Fragments** to ease screen navigation , view lifecycle management and maintain the top Toolbar and bottom tabs across all screens.

-> Implemented back **navigation button** in Restaurant details screen using Android default icon provider not with any explicit image.

-> Restricted **Google Maps Api key** , which will work only for this project with package name.

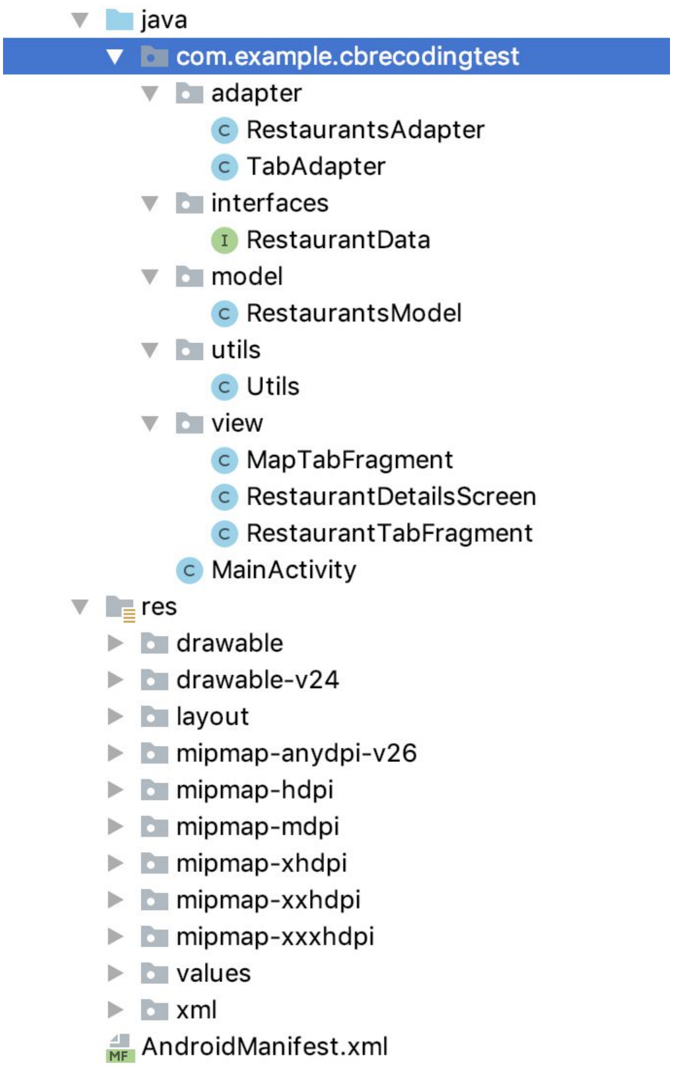
-> Given zoom in Map to properly display the map markers otherwise all markers will overlap and show as a single marker .

```
mMap.animateCamera( CameraUpdateFactory.zoomTo( 11.0f ) );
```

-> Created separate Utils class , which holds all utility data needed for the complete project.

-> All code parts are divided into separate independent modules to allow further extension of feature implementation.

-> Divided the project structure into multiple parts allows easy understanding of code , extendability , reusability , scalability .



```
▼ java
  ▼ com.example.cbrecodingtest
    ▼ adapter
        RestaurantsAdapter
        TabAdapter
    ▼ interfaces
        RestaurantData
    ▼ model
        RestaurantsModel
    ▼ utils
        Utils
    ▼ view
        MapTabFragment
        RestaurantDetailsScreen
        RestaurantTabFragment
      MainActivity
▼ res
  ▶ drawable
  ▶ drawable-v24
  ▶ layout
  ▶ mipmap-anydpi-v26
  ▶ mipmap-hdpi
  ▶ mipmap-mdpi
  ▶ mipmap-xhdpi
  ▶ mipmap-xxhdpi
  ▶ mipmap-xxxhdpi
  ▶ values
  ▶ xml
  AndroidManifest.xml
```

-> Included all dimensions into **dimens.xml**

-> Separated all colors into **colors.xml**