A Project Report on

“WAKE N BAKE”

Submitted by: -

Bhuta Nachiket Mehul (1598004)

Chandura Dishant Tushar (1598005)

Desai Dhairya Rahul (1598006)

Guided By: -

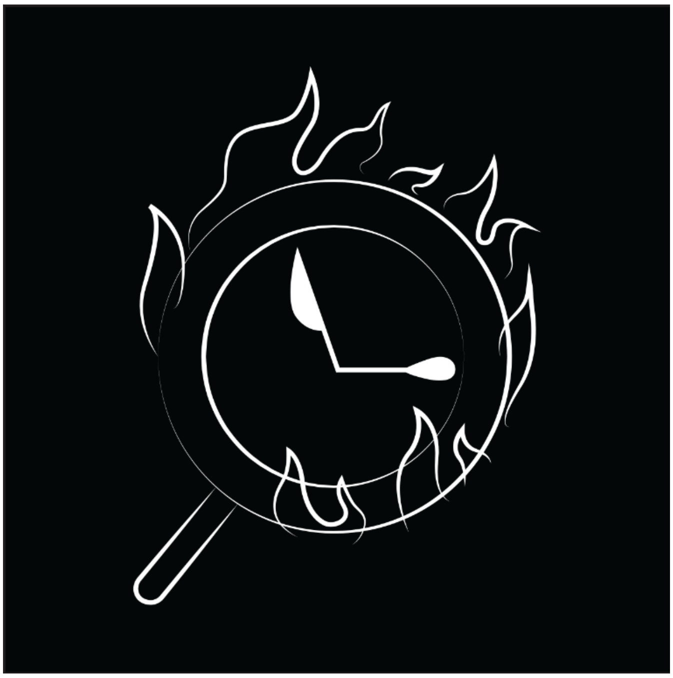
Mr. Manish R. Solanki

Term – December’17 to April’18

Department of Information Technology

SVKM’s Shri Bhagubhai Mafatlal Polytechnic

Irla, N.R.G Marg, Vile Parle (W), Mumbai – 400056.



Wake N Bake

INDEX

LIST OF FIGURES

ACKNOWLEDGEMENT

We take immense pleasure in thanking, Professor. UMESH KANTUTE, Principal of Shri Bhagubhai Mafatlal Polytechnic for having permitted us to carry out this project work.

We wish to express our deep sense of gratitude to my Internal Guide, Mr. MANISH SOLANKI for his able guidance and useful suggestions, which helped us in completing the project work, in time. We would like to thank him for his very valuable assistance in the project work. Words are inadequate in offering my thanks the entire staff of I.T/C.S.E for providing us with all the amenities and facilities.

The success of this project would not have been possible without the constant encouragement, advice and support from a vast number of people. Ideas, they say often remain ideas. Very few of them, if find support are turned into actual working models. And the rest of them are just forgotten and remain what they started off as ideas. This idea as well would have remained, but for one person, The real debt of gratitude that we owe to him, without whom this project would not have seen light of the day .We, the members of the team, who developed “WAKE N BAKE” are very delighted to take the opportunity to acknowledge whole heartedly the innumerable guidance & support extended to us by our guide.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved parents for their blessings, our friends/classmates for their help and wishes for the successful completion of this project.

1. PROBLEM STATEMENT

Students and teenagers usually study at night and then when they are hungry, they have no idea where to go to have some food. The only option they have is to have to have some food available in the house. Sometimes the family members are out of town, they require food at night. But this application solves this problem since it will provide the user to contact some food joints.

Another problem that is solved by this application is that all the students living in hostels, PG or those living alone who are all tired of eating fast foods or junk food and cannot cook food on their own.

Students can contact local home-made food vendors. Home-made food vendors can also register themselves on this application. They can enter the details of the time they will be available. They can also upload the menu of the foods they make and the phone number of the home made food vendors will be provided to the user location.

2. ABSTRACT

Mobile applications can be one of the best ways to keep consumers engaged with a brand as they are on the move. With the increase in demand for smartphones and efficiency of wireless networks, the demand for mobile applications has increased incredibly. Android is one of the most popular open source platforms that offers the developer's full access to the framework API's so as to build innovative applications.

This project is focused on providing users with an interface to search for fast food joints/restaurants at midnight or during the time between 1-6 am. The application has a feature to provide local food vendors with a way to connect with customers in their region or in their area. Also the user will be able to browse through the photos, reviews and ratings and decide where he/she wants to eat and also the user can use the map feature to guide themselves there.

3. INTRODUCTION

The era of mobile technology opens the windows to the android app. The websites are diminishing and the mobile phones are emerging. It’s the time to change from conventional websites to apps, which has become the part of our daily routine. We are introducing the ‘WAKE N BAKE’ to provide people with different vendors and food joints at different time and providing the GPS location of each food joint and vendors as people from Mumbai are very foody and are always looking for new good food. Our application tries to connect to the people and provide them basic details like phone numbers and GPS location.

This application provides many good features to many people like firstly who are new to the city and don’t know about good food joints, so by using one day module of our application, they can find different good food joints. Then many people study or work at night, some need food and some beverages like tea and coffee and are unaware or are really bored of the same places, they can use our night module to find the new places. Then the people staying in the city in hostels or as Paying Guests in some places, they might be missing their mom’s food or are bored of the same food , they can use our tiffin module in which they are provided with details of different house wives who prepare homely food. In this way, this application connects to different people and help them use this application as daily necessity or sometimes to change their routine.

4. HARDWARE REQUIREMENTS

1. Mobile Phone

- The mobile phone should be an Android powered phone.
- The mobile phone should have atleast Android API 19 (Kitkat).
- The mobile phone should have an Internet Connection.

2. For Development,

For Windows

- Microsoft® Windows® 7/8/10 (32- or 64-bit).
- 3 GB RAM minimum.
- 2 GB of available disk space minimum.
- 1280 x 800 minimum screen resolution.

5. SOFTWARE REQUIREMENTS

1. Android Studio

It will be used to design the layout of all the user-interface screen. And it will be used to connect to PHP. (PHP file will be used to connect and retrieve data from database) and Java will be used to develop the application for passing different values between different activities and intents.

2. XAMPP

XAMPP is the most popular PHP development environment. XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

3. JSON

JSON (JavaScript Object Notation) is a lightweight text-based open standard designed for human-readable data interchange and it is used in this application to send data from Android device to PHP Script. When the application is executed, it connects the device to the PHP script on the server. PHP script fetches the response data which is encoded to JSON format and then sent back to the device. The data is parsed and displayed according to the requirement.

4. Firebase

We will use Firebase to authenticate each user and stores the location of different food joints. Firebase will be even used for analytics. Firebase will be even used to provide notifications on the user’s phone.

We used Firebase Authentication (Google Sign-In, One-Time Password, Email), database and storage.

5. Postman

Postman is a powerful HTTP client for testing web services. Created by Abhinav Asthana, a programmer and designer based in Bangalore, India, Postman makes it easy to test, develop and document APIs by allowing users to quickly put together both simple and complex HTTP requests. Postman is available as both a Google Chrome Packaged App and a Google Chrome in-browser app. The packaged app version includes advanced features such as OAuth 2.0 support and bulk uploading/importing that are not available in the in-browser version. The in-browser version includes a few features, such as session cookies support, that are not yet available in the packaged app version. At publication time, the Postman REST Client is one of the highest-rated productivity apps in the Chrome Web Store, with more than 348,000 unique users (for both versions), and more than 63,000 collections shared via Postman (more on that later). This post is an in-depth review of the Postman Google Chrome Packaged App highlighting the key features that make Postman a must-have tool for API developers and consumers at all levels of experience.

6. Sublime Text

Sublime Text is a superfast and feature packed text and development editor. If you are going to be coding regularly you want to try this amazing editor (IDE). Following some of the great features that make Sublime Text stand out from other code editors:

- **Multiple cursors**: Once you have discovered multiple cursors you won't want to work without them anymore. As the name suggests they let you write or edit in multiple places in a document at the same time.
- **Vintage mode**: Vim keyboard shortcuts will work just like in the original Vim editor. To use them, all you need to do is to enable vintage mode.
- **Lightning fast**: This is the fastest code editor you will find right now.
- **Command pallet**: A great feature that allows you to reach about all functions of the editor via the keyboard. You will hardly use your mouse and thus code more efficiently.
- **Plugin collection**: A hugely active community creates plugins for almost any task in Sublime Text. This includes syntax highlighting and code snippets for a large number of languages, for example JavaScript, PHP, CSS, HTML, Python, LESS, XML and C++ to name just a few.
- **Package control**: This add-on lets you install plugins within seconds directly from the editor.

7. Git

Git is a mature, actively maintained open source project originally developed in 2005 by Linus Torvalds, the famous creator of the Linux operating system kernel. A staggering number of software projects rely on Git for version control, including commercial projects as well as open source. Developers who have worked with Git are well represented in the pool of available software development talent and it works well on a wide range of operating systems and IDEs (Integrated Development Environments).

6. GANTT CHART

Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Planning	🟦															
Requirement Analysis		⬛														
Literature Survey			🟩													
Learning new technologies				🟪	🟪											
Designing						🟥	🟥									
Coding & Testing								🟦	🟦	🟦	🟦	🟦	🟦	🟦		
System Testing & Error debugging															🟩	
Project Report																🟪

7. LITERATURE SURVEY

7.1 INTRODUCTION TO ANDROID OS

Android is a software package and Linux based operating system for mobile devices such as tablet computers and smartphones. Android is a software bunch comprising not only operating system but also middleware and key applications. Android Inc. was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick Sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version of Android. The goal of android project is to create a successful real-world product that improves the mobile experience for end users. Android is a powerful OS supporting a large number of applications in smartphones. These applications make life more comfortable and advanced for the users. Hardware that support Android are mainly based on ARM architecture platform.

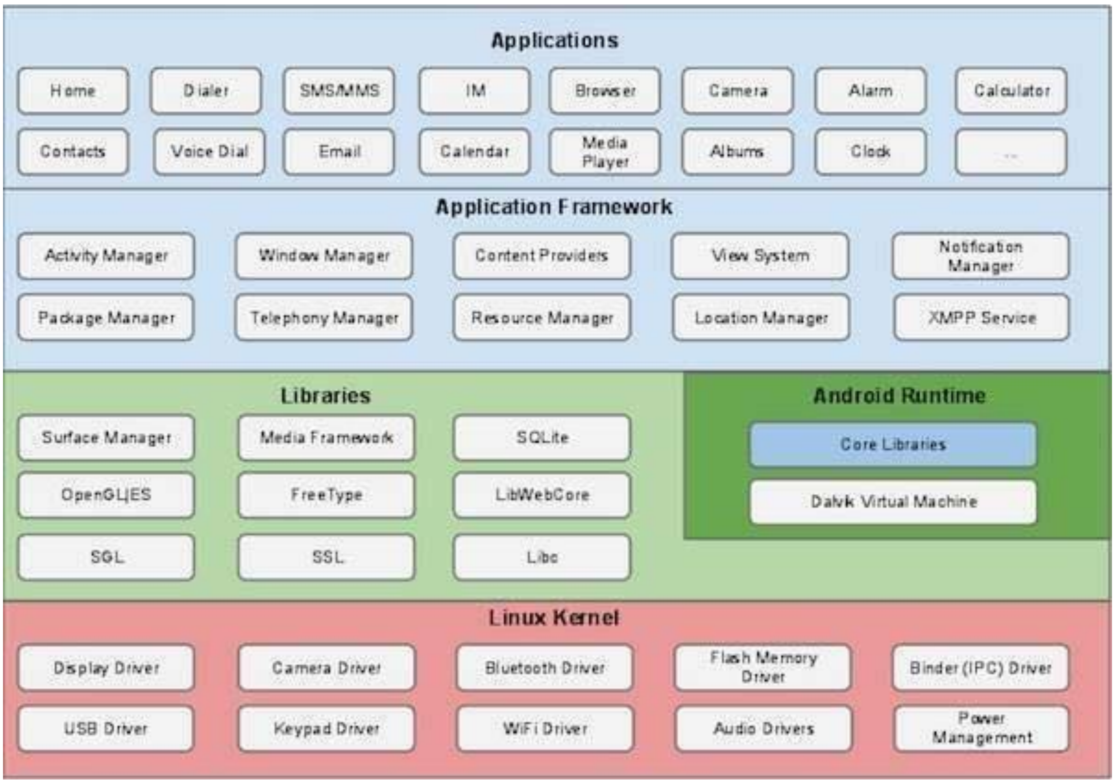


Figure 1: Architecture of Android OS

Link:https://www.tutorialspoint.com/android/android_architecture.htm

7.1.1 KEY FEATURES OF ANDROID OS

Android is a powerful operating system competing with Apple's iOS and supports great features. The following are some of the features:-

- **Beautiful UI:** Android OS basic screen provides a beautiful and intuitive user interface.
- **Connectivity:** GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
- **Storage:** SQLite, a lightweight relational database, is used for data storage purposes.
- **Media support:** H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg-Vorbis, WAV, JPEG, PNG, GIF, and BMP.
- **Messaging:** SMS and MMS.
- **Web browser:** Based on the open-source Web Kit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
- **Multi-touch:** Android has native support for multi-touch which was initially made available in handsets.
- **Multi-tasking:** User can jump from one task to another and same time various application can run simultaneously.
- **Resizable widgets:** Widgets are resizable, so users can expand them to show more content or shrink them to save space.
- **Multi-Language:** Supports single direction and bi-directional text.
- **GCM:** Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.
- **Wi-Fi Direct:** A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
- **Android Beam:** A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

7.2 ANDROID APPLICATIONS

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**, **SlideME**, **Opera Mobile Store**, **Mobango**, **F-droid** and the **Amazon AppStore**. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

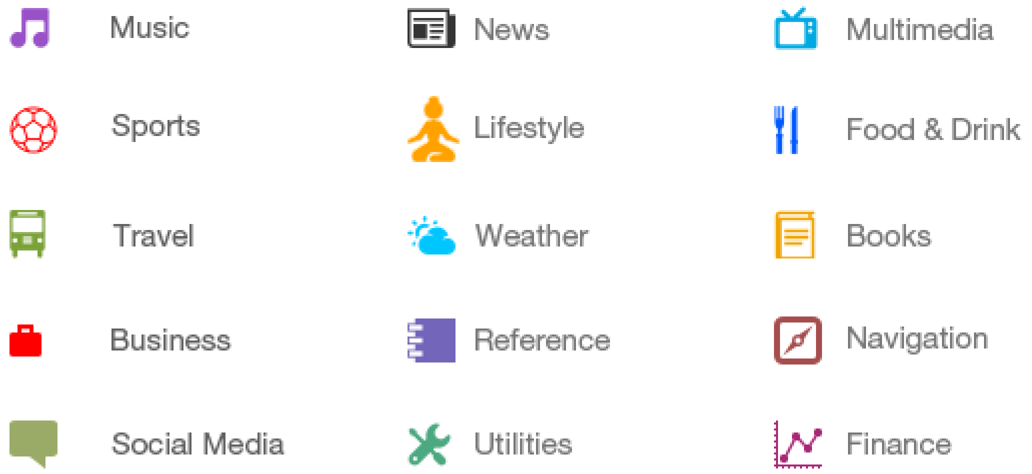


Figure 2: Categories of Android Applications

Link: https://www.tutorialspoint.com/android/android_overview.htm

7.2.1 COMPONENTS OF ANDROID APPLICATION

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file **AndroidManifest.xml** that describes each component of the application and how they interact.

There are following four main components that can be used within an Android application:

1. **Activities**: They dictate the UI and handle the user interaction to the smart phone screen.
2. **Services**: They handle background processing associated with an application.
3. **Broadcast Receivers**: They handle communication between Android OS and applications.
4. **Content Providers**: They handle data and database management issues.

7.3 PHP

The PHP Hypertext Pre-processor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. MYSQL can be used as a database at the webserver and PHP is used to fetch data from the database. Our application will communicate with the PHP page with necessary parameters and PHP will contact MYSQL database and will fetch the result and return the results to us.

7.4 FIREBASE

Firebase is a development platform for mobile and web applications. It provides tools and infrastructure to enable a developer to deliver functionality across multiple platforms more easily and efficiently. The product range mainly addresses application developers and vendors who can use a SDK to address various APIs, and thus the features offered by Firebase, and use them in their applications.

7.4.1 PRODUCTS & SERVICES BY FIREBASE

Firebase offers as a platform various, partly free, services and tools. In the following the individual offers are listed and described:

1. **Analytics**

   Analytics is Firebase's flagship product and offers unlimited analytics solutions for free. Unlimited in this case means that the solutions provided by Firebase can collect and manage an unlimited amount of Analytics data. There are 500 event types with up to 25 attributes available. The collected data can be evaluated in dashboards and various graphics and thus provide information about user behaviour. In addition, an export of the data to the Google Cloud Platform service Big Query is possible to run own evaluations on the data. The use of Big Query is not free and is not included in Firebase, but must be billed separately.

2. **Cloud Messaging**

   Formerly known as Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM for short) provides a cross-platform way to deliver and receive notifications for android, iOS, and the Web using JavaScript.

3. **Authentication**

   Firebase Authentication provides a cross-platform system for different authentication options. Included are the registration with Facebook, Twitter, Google, GitHub as well as username and password. The integration into an existing authentication system should be possible without any problems.

4. **Real-time database**

   With the NoSQL real-time database, data can be saved as JSON and synchronized across multiple devices. Access is also possible when the device is currently offline.

5. **File storage**

   A cloud-based storage for user-generated data. The data is stored in Google Cloud Storage.

6. **Hosting**

   With hosting, Firebase provides a content delivery network (CDN) available for web platforms to deliver resources faster to users (e.g. images or JavaScript code).

7. **Test Lab**

A test platform for Android apps.

8. **Crash Reports**

Firebase's Crash Reporting service provides the ability to gather detailed information if the app crashes due to an unexpected error on a device or if the system stops responding when it stops responding.

9. **Notifications**

Send notifications as a method to increase user interaction with an app.

10. **Remote configuration**

Change configuration parameters of the app without having to create a new release and distribute it through the Google Play Store.

11. **App Indexing**

Helps to make the content of the app index able in Google Search, thus increasing user interaction and finding app content.

12. **Dynamic links**

Dynamic links are ultimately the same as deep links to specific content in the app. The difference is that these dynamic links also survive the app's installation process, leading new users to the content they wanted to see before installing the app.

13. **Invitations**

An out-of-the-box solution to allow existing users to share an app and invite friends or acquaintances to the app.

7.5 VERSION CONTROL SYSTEM (VCS)

Version control systems are a category of software tools that help a software team manage changes to source code over time. Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code to help fix the mistake while minimizing disruption to all team members.

For most software teams, the source code is a repository of the invaluable knowledge and understanding about the problem domain that the developers have collected and refined through careful effort. Version control protects source code from both catastrophe and the casual degradation of human error and unintended consequences.

Software developers working in teams are continually writing new source code and changing existing source code. The code for a project, app or software component is typically organized in a folder structure or "file tree". One developer on the team may be working on a new feature while another developer fixes an unrelated bug by changing code, each developer may make their changes in several parts of the file tree.

Version control helps teams solve these kinds of problems, tracking every individual change by each contributor and helping prevent concurrent work from conflicting. Changes made in one part of the software can be incompatible with those made by another developer working at the same time. This problem should be discovered and solved in an orderly manner without blocking the work of the rest of the team. Further, in all software development, any change can introduce new bugs on its own and new software can't be trusted until it's tested. So testing and development proceed together until a new version is ready.

7.5.1 VERSION CONTROL SYSTEM USING GIT

Git is the best choice for most software teams today. While every team is different and should do their own analysis, here are the main reasons why version control with Git is preferred over alternatives:

1. **Git is good**

   Git has the functionality, performance, security and flexibility that most teams and individual developers need. These attributes of Git are detailed above. In side-by-side comparisons with most other alternatives, many teams find that Git is very favourable.

2. **Git is a de facto standard**
   Git is the most broadly adopted tool of its kind. This is makes Git attractive for the following reasons. Vast numbers of developers already have Git experience and a significant proportion of college graduates may have experience with only Git. While some organizations may need to climb the learning curve when migrating to Git from another version control system, many of their existing and future developers do not need to be trained on Git. In addition to the benefits of a large talent pool, the predominance of Git also means that many third party software tools and services are already integrated with Git including IDEs, and our own tools like DVCS desktop client Source tree, issue and project tracking software, Jira, and code hosting service, Bit bucket.

3. **Git is a quality open source project**

   Git is a very well supported open source project with over a decade of solid stewardship. The project maintainers have shown balanced judgment and a mature approach to meeting the long term needs of its users with regular releases that improve usability and functionality. The quality of the open source software is easily scrutinized and countless businesses rely heavily on that quality.

Git enjoys great community support and a vast user base. Documentation is excellent and plentiful, including books, tutorials and dedicated web sites. There are also podcasts and video tutorials.

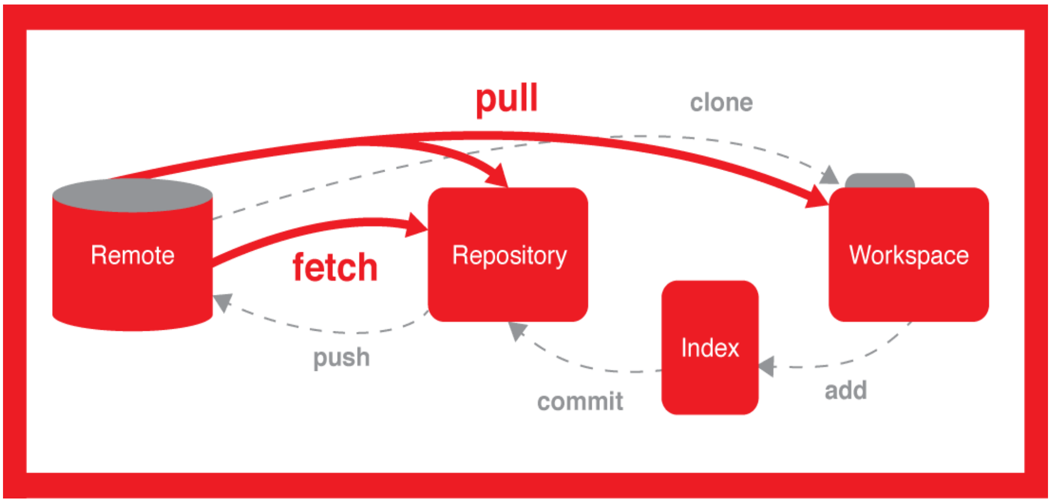


Figure 3 : Basic Architecture of Git

Link: https://foxutech.com/git-interview-question-and-answers/

7.6 GITHUB

GitHub is the single largest host for Git repositories, and is the central point of collaboration for millions of developers and projects. GitHub is a Web-based Git version control repository hosting service. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

7.7 GOOGLE APIS

Google APIs is a set of application programming interfaces (APIs) developed by Google which allow communication with Google Services and their integration to other services. Third-party apps can use these APIs to take advantage of or extend the functionality of the existing services. The APIs provide functionality like analytics, machine learning as a service (the Prediction API) or access to user data (when permission to read the data is given). Another important example is an embedded Google map on a website, which can be achieved using the Static maps API, Places API or Google Earth API.

7.7.1 GOOGLE PLAY SERVICES

With Google Play services, applications can take advantage of the latest, Google-powered features such as Maps, Google+, and more, with automatic platform updates distributed as an APK through the Google Play store. This makes it faster for your users to receive updates and easier for you to integrate the newest that Google has to offer.

Google Play services gives you the freedom to use the newest APIs for popular Google services without worrying about device support. Updates to Google Play services are distributed automatically by the Google Play Store and new versions of the client library are delivered through the Android SDK Manager. This makes it easy for you to focus on what's important: your users' experience.

7.7.2 GOOGLE MAPS API

Build full-featured Android apps for users. Google Maps APIs for Android are available via Google Play services so your app can be location-aware, include data-rich maps, find relevant places nearby and more. Add maps to your Android app. Integrate base maps, 3D buildings, indoor floor plans, Street View and Satellite imagery, custom markers and more.

7.7.3 GOOGLE PLACES API

With the Google Maps Android API, maps can be added based on Google Maps data to your application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. You can also use API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map.

7.8 RESTFUL API

A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

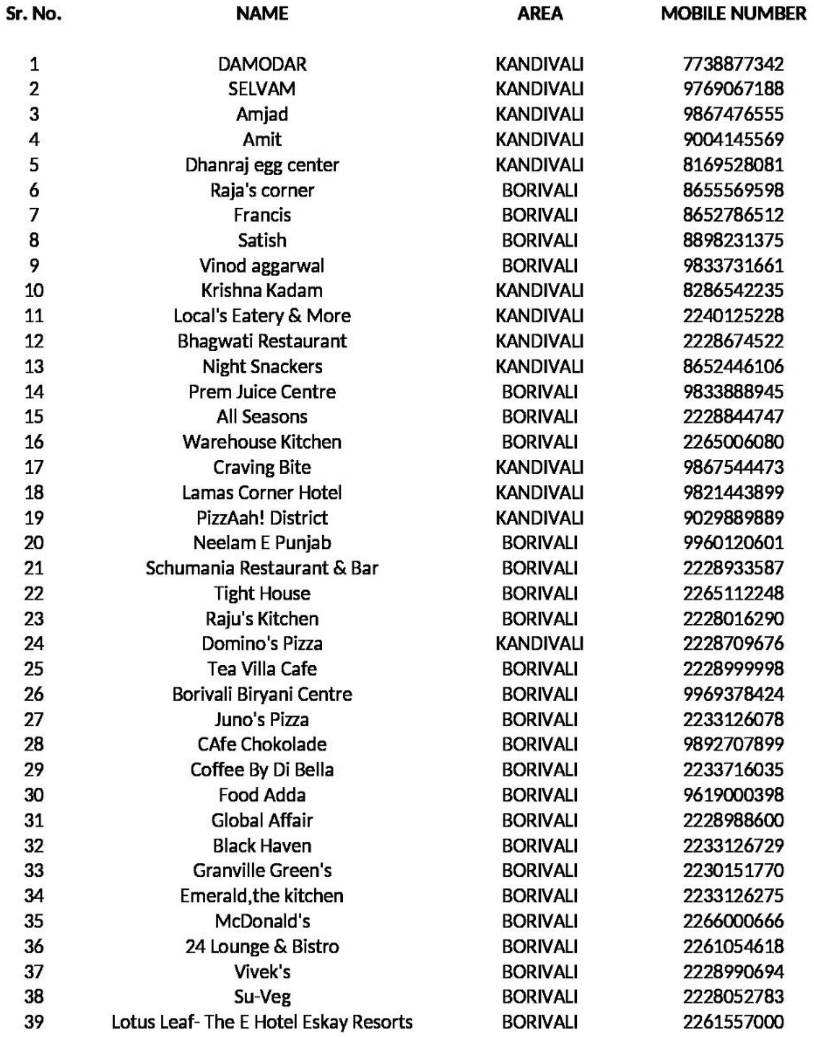
It also referred to as a RESTful web service- is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development. REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST leverages less bandwidth, making it more suitable for internet usage.

An API for a website is code that allows two software programs to communicate with each another. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.

The REST used by browsers can be thought of as the language of the internet. With cloud use on the rise, APIs are emerging to expose web services. REST is a logical choice for building APIs that allow users to connect and interact with cloud services. RESTful APIs are used by such sites as Amazon, Google, LinkedIn and Twitter.

7.9 SURVEY OF FOOD JOINTS

We had done a survey of the food joints which are available at night between 12am to 6am. The list of food joints are as follows: -



Sr. No.	NAME	AREA	MOBILE NUMBER
1	DAMODAR	KANDIVALI	7738877342
2	SELVAM	KANDIVALI	9769067188
3	Amjad	KANDIVALI	9867476555
4	Amit	KANDIVALI	9004145569
5	Dhanraj egg center	KANDIVALI	8169528081
6	Raja's corner	BORIVALI	8655569598
7	Francis	BORIVALI	8652786512
8	Satish	BORIVALI	8898231375
9	Vinod aggarwal	BORIVALI	9833731661
10	Krishna Kadam	KANDIVALI	8286542235
11	Local's Eatery & More	KANDIVALI	2240125228
12	Bhagwati Restaurant	KANDIVALI	2228674522
13	Night Snackers	KANDIVALI	8652446106
14	Prem Juice Centre	BORIVALI	9833888945
15	All Seasons	BORIVALI	2228844747
16	Warehouse Kitchen	BORIVALI	2265006080
17	Craving Bite	KANDIVALI	9867544473
18	Lamas Corner Hotel	KANDIVALI	9821443899
19	PizzAah! District	KANDIVALI	9029889889
20	Neelam E Punjab	BORIVALI	9960120601
21	Schumania Restaurant & Bar	BORIVALI	2228933587
22	Tight House	BORIVALI	2265112248
23	Raju's Kitchen	BORIVALI	2228016290
24	Domino's Pizza	KANDIVALI	2228709676
25	Tea Villa Cafe	BORIVALI	2228999998
26	Borivali Biryani Centre	BORIVALI	9969378424
27	Juno's Pizza	BORIVALI	2233126078
28	CAfe Chokolade	BORIVALI	9892707899
29	Coffee By Di Bella	BORIVALI	2233716035
30	Food Adda	BORIVALI	9619000398
31	Global Affair	BORIVALI	2228988600
32	Black Haven	BORIVALI	2233126729
33	Granville Green's	BORIVALI	2230151770
34	Emerald,the kitchen	BORIVALI	2233126275
35	McDonald's	BORIVALI	2266000666
36	24 Lounge & Bistro	BORIVALI	2261054618
37	Vivek's	BORIVALI	2228990694
38	Su-Veg	BORIVALI	2228052783
39	Lotus Leaf- The E Hotel Eskay Resorts	BORIVALI	2261557000

Figure 4 : List of Food Joints

\

8. SYSTEM DESIGN

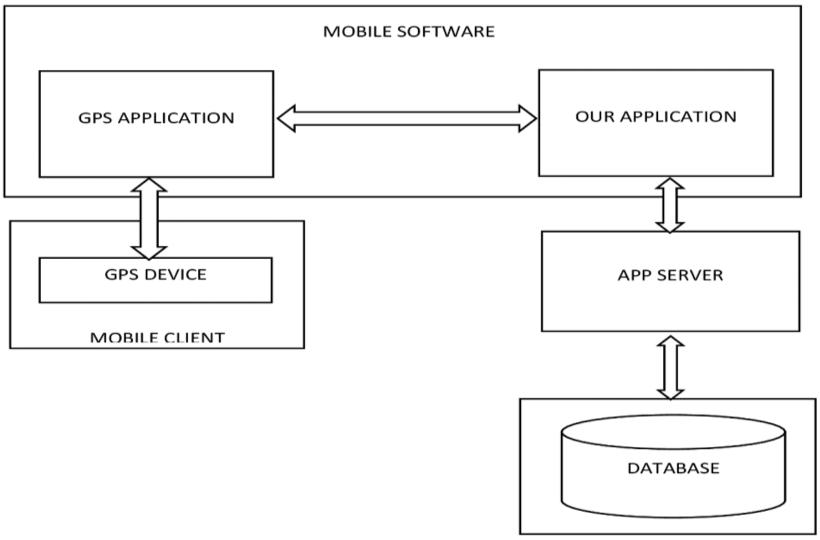
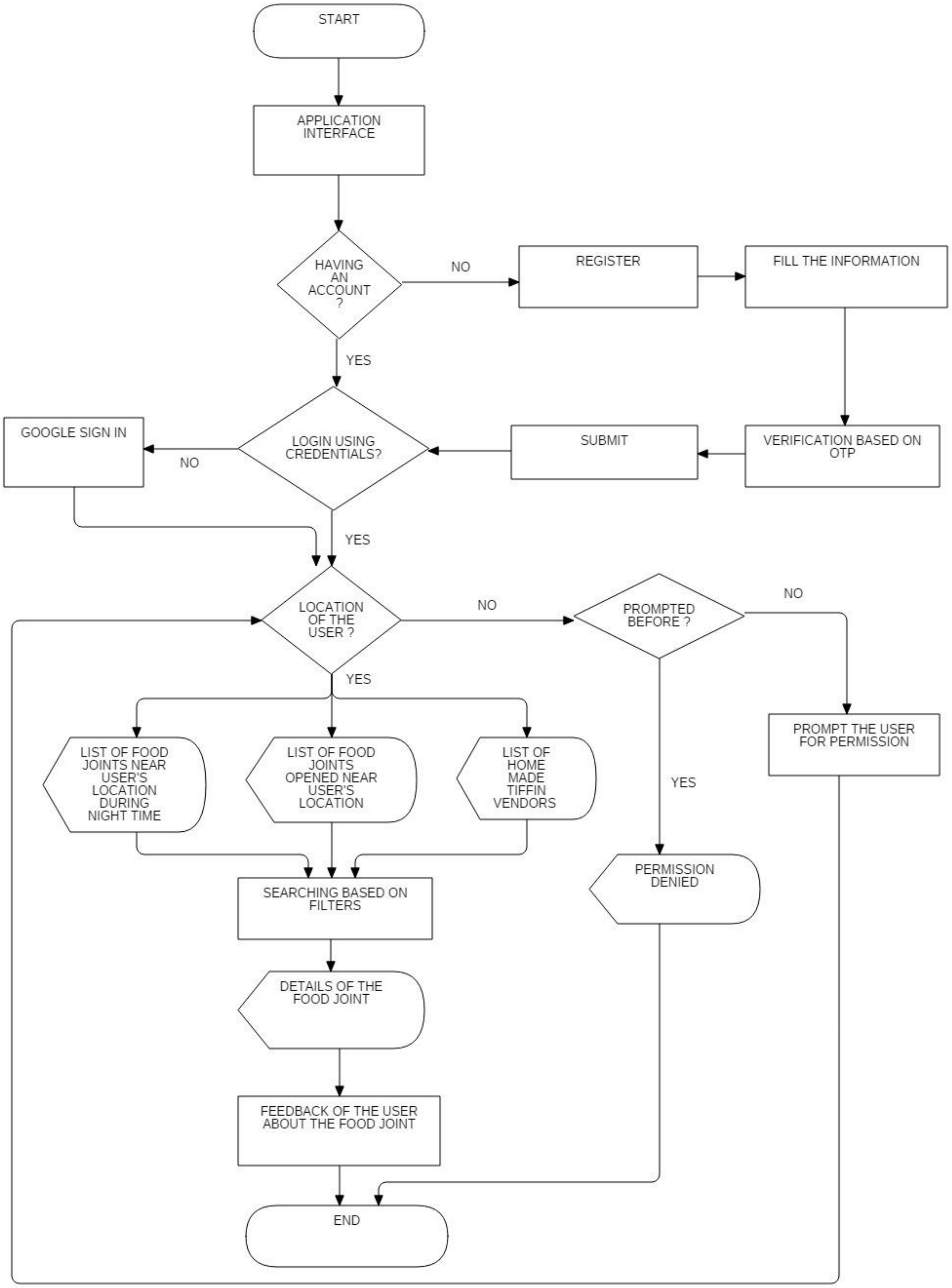


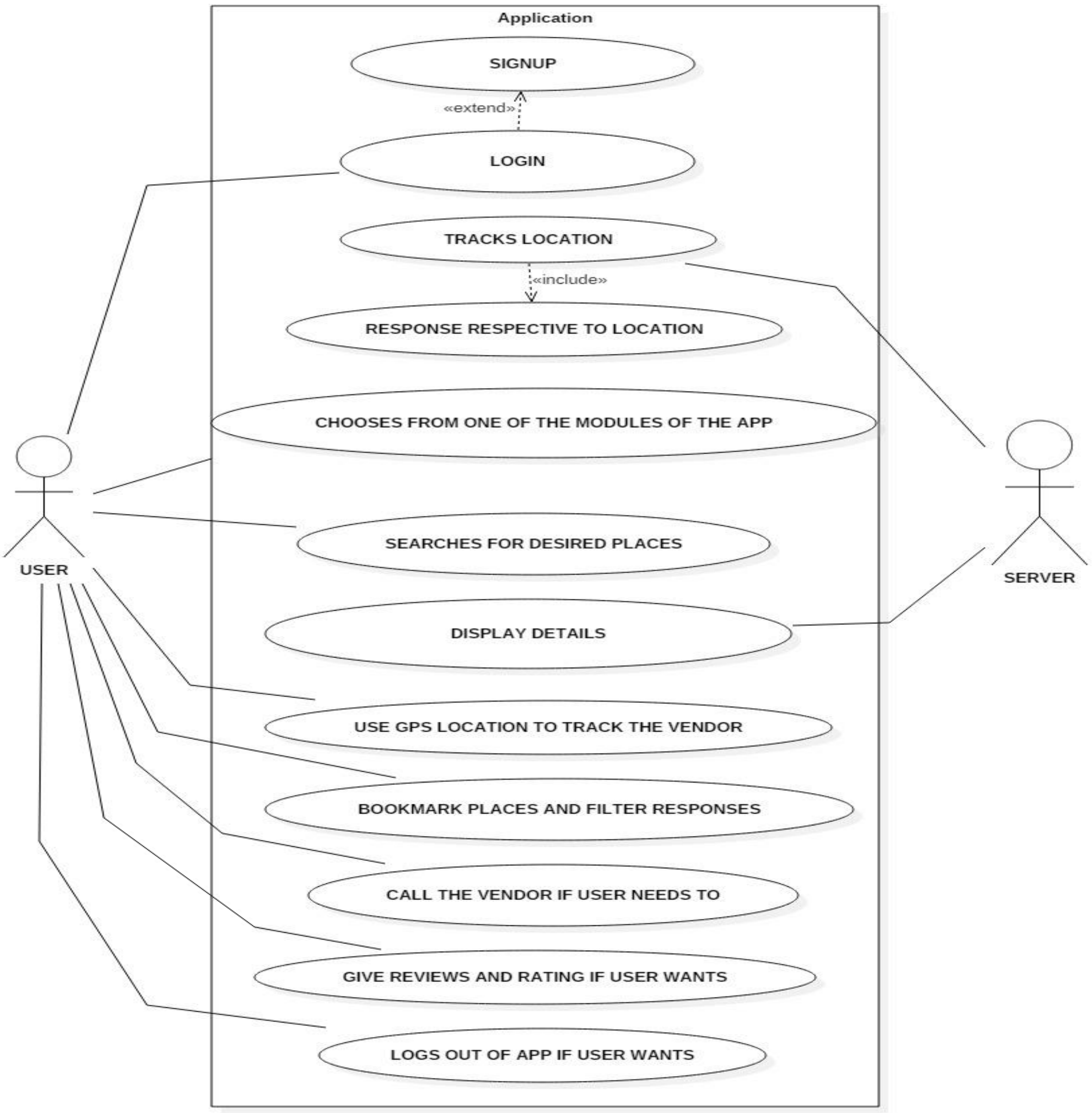
Figure 5 : Basic Architecture of the Application

- The user will have to register himself/herself with email and phone number to the application. A two-step authentication will be done using One Time Password (OTP) and the OTP will be sent to the phone number of the user. The OTP will be generated and the details of the user will be stored in the database at the server side.
- The user’s location will be detected by Google Maps API and it will be sent to server. The food joints located near the user’s location will be detected and the location of the food joint will be responded to the application. The user will have to choose the type of cuisine and based on that, the nearest food joints will be selected.
- The location of the food joint will be displayed to the user in the application. When the user will click on the location of the food joint, he/she will be redirected to the Google Maps.

8.1 FLOWCHART

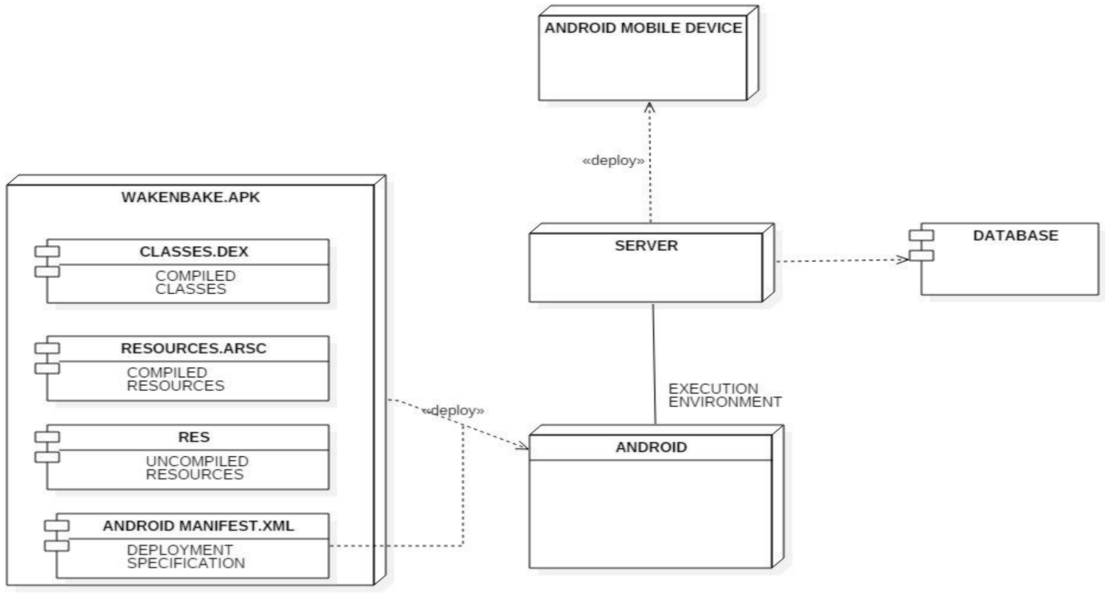

START
APPLICATION INTERFACE
HAVING AN ACCOUNT ?
NO
REGISTER
FILL THE INFORMATION
YES
GOOGLE SIGN IN
NO
LOGIN USING CREDENTIALS?
SUBMIT
VERIFICATION BASED ON OTP
YES
LOCATION OF THE USER ?
NO
PROMPTED BEFORE ?
NO
YES
LIST OF FOOD JOINTS NEAR USER'S LOCATION DURING NIGHT TIME
LIST OF FOOD JOINTS OPENED NEAR USER'S LOCATION
LIST OF HOME MADE TIFFIN VENDORS
YES
PROMPT THE USER FOR PERMISSION
SEARCHING BASED ON FILTERS
PERMISSION DENIED
DETAILS OF THE FOOD JOINT
FEEDBACK OF THE USER ABOUT THE FOOD JOINT
END

8.2 USE CASE DIAGRAM


Application
SIGNUP
«extend»
LOGIN
TRACKS LOCATION
«include»
RESPONSE RESPECTIVE TO LOCATION
CHOOSES FROM ONE OF THE MODULES OF THE APP
SEARCHES FOR DESIRED PLACES
DISPLAY DETAILS
USE GPS LOCATION TO TRACK THE VENDOR
BOOKMARK PLACES AND FILTER RESPONSES
CALL THE VENDOR IF USER NEEDS TO
GIVE REVIEWS AND RATING IF USER WANTS
LOGS OUT OF APP IF USER WANTS
USER
SERVER

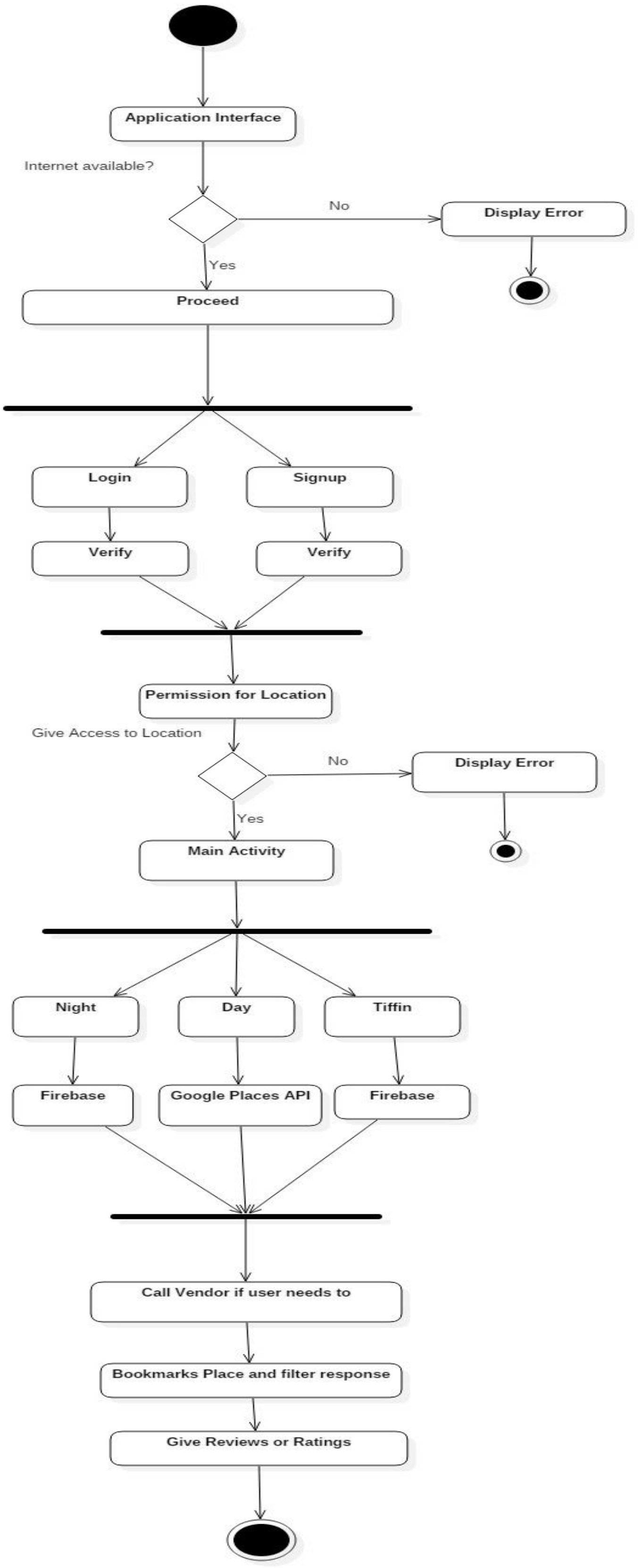
- A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.
- This use case describes the overall structure of our application with different interactive functions of the user and use cases of the diagram.
- Here the application provides functionalities like login, signup, different modules, map view, ratings, review and this diagram shows somewhat exact replication of interaction between users and use cases.

8.3 DEPLOYMENT DIAGRAM



- A deployment diagram in the Unified Modelling Language models the physical deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).
- Here in our diagram the different hardware of the application of the android studio and shows communication between android mobile device and server (We have used a web server as well as used XAMPP application server to run it locally) via android studio development.

8.4 ACTIVITY DIAGRAM



- Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
- The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.
- This activity diagram describes the overall operations/activities of our application.
- The flow of our activity diagram is sequential.
- It has many different operations like login, signup, reviews, etc.
- It deals with all flow control with different elements such as fork, join, etc.

9. LIBRARIES

9.1 Android Support Library

When developing apps that support multiple API versions, developers may want a standard way to provide newer features on earlier versions of Android or gracefully fall back to equivalent functionality. Rather than building code to handle earlier versions of the platform, developers can leverage these libraries to provide that compatibility layer. In addition, the Support Libraries provide additional convenience classes and features not available in the standard Framework API for easier development and support across more devices.

Library: com.android.support:support-core-utils:26.1.0

9.2 Room Persistence Library

The Room persistence library provides an abstraction layer over SQLite to allow fluent database access while harnessing the full power of SQLite. The library helps to create a cache of app's data on a device that's running your app. This cache, which serves as app's single source of truth, allows users to view a consistent copy of key information within your app, regardless of whether users have an internet connection.

Library: android.arch.persistence.room:runtime:1.0.0
android.arch.persistence.room:compiler:1.0.0

9.3 FirebaseUI Auth Library

FirebaseUI is an open-source library that offers simple, customizable UI bindings on top of the core Firebase SDKs. It aims to eliminate boilerplate code and promote best practices (both user experience and security) for authentication. A simple API is provided for drop-in user authentication which handles the flow of signing in users with email addresses and passwords, phone numbers, and federated identity providers such as Google Sign-In, and Facebook Login. It is built on top of Firebase Auth.

Library: com.firebaseui:firebase-ui-auth:3.3.0

9.4 FirebaseUI Database Library

FirebaseUI makes it simple to bind data from the Firebase Realtime Database to your app's UI.

Library: com.firebaseui:firebase-ui-database:3.3.0

9.5 Material Search Bar

Library: com.github.mancj: MaterialSearchBar: 0.7.6
Name: mancj
Use: Search Bar
Contributors:-

- mancj
- GitUser4732
- RuijiePan
- Pavel163
- Vbauer

Link: https://github.com/mancj/MaterialSearchBar

9.6 Pin View

Library: com.github.GoodieBag: Pinview: v1.3
Name: GoodieBag/Pinview
Use: OTP View
Contributors:

- GrumpyKai
- KoderKrishanu
- Arberg
- Koushik1994

Link: https://github.com/GoodieBag/Pinview

9.7 Simple Rating Bar

Library: com.github.ome450901: SimpleRatingBar: 1.4.1
Name: ome450901
Use: Rating Bar
Contributors:

- ome450901
- ANPez
- Nappannda
- harshsharma94
- desmese
- zhangruize

Link: https://github.com/ome450901/SimpleRatingBar

9.8 Smart Tab Layout

Library: com.ogaclejapan.smarttablayout:library:1.6.1@aar
Name: ogcalejapan
Use: Login, Signup tab and Homepage Tabs

Contributors:

- satorufujiwara
- Numan1617
- shaunkawano
- ffts
- EzimetYusup
- ralexey
- ataulm
- songzhiyong
- magiepooh
- nshmura
- cosicLink

Link: https://github.com/ogaclejapan/SmartTabLayout

9.9 Android Ripple Background

Library: com.skyfishjy.ripplebackground: library: 1.0.1
Name: skyfishjy
Use: Ripple Effect in background
Contributors:

- skyfishjy
- xujinyang

Link: https://github.com/skyfishjy/android-ripple-background

9.10 Retrofit

Retrofit is a type-safe HTTP client for Android and Java. Retrofit makes it easy to connect to a REST web service by translating the API into Java interfaces. In this tutorial, I'll show you how to use one of most popular and often-recommended HTTP libraries available for Android.
This powerful library makes it easy to consume JSON or XML data which is then parsed into Plain Old Java Objects (POJOs). GET, POST, PUT, PATCH, and DELETE requests can all be executed.

Library: com.squareup.retrofit2: retrofit: 2.4.0
Name: Retrofit
Use: To make API calls to the web server
Contributors: 119 contributors
Link: https://github.com/square/retrofit

9.11 Picasso

Picasso allows for hassle-free image loading in your application—often in one line of code!

Library:
Name: Picasso
Use: To download images through URL

Contributors: 85 contributors
Link: https://github.com/square/picasso

9.12 RecyclerView Library

The RecyclerView is a new ViewGroup that is prepared to render any adapter-based view in a similar way. It is supossed to be the successor of ListView and GridView, and it can be found in the latest support-v7 version.

Library: com.android.support:recyclerview-v7:27.1.1

10. SOURCE CODE

10.1 Android Manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="nshmadhani.com.wakenbake">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.CALL_PHONE" />

    <uses-feature android:name="android.hardware.location.gps" />

    <application
        android:name=".Models.WakeNBake"
        android:allowBackup="true"
        android:icon="@mipmap/app_icon"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.NoActionBar">
        <activity android:name=".Activities.SplashScreenActivity">
            <intent-filter>
                <category android:name="android.intent.category.LAUNCHER"
/>

                <action android:name="android.intent.action.MAIN" />
            </intent-filter>
        </activity>
        <activity android:name=".Activities.LoginActivity" />
        <activity android:name=".Activities.OtpActivity" />
        <activity android:name=".Activities.LocationActivity" />

        <!-- Google Maps -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".Activities.NavigationActivity"
            android:label="@string/title_activity_navigation"
            android:theme="@style/Theme.AppCompat.NoActionBar" />
        <activity
            android:name=".Activities.GooglePlacesActivity"
            android:label="@string/title_activity_place"
            android:theme="@style/Theme.AppCompat.NoActionBar" />

        <activity android:name=".Activities.BookmarkActivity" />
        <activity
            android:name=".Activities.FirebasePlaceActivity"
            android:label="@string/title_activity_firebase_place"
            android:theme="@style/Theme.AppCompat.NoActionBar" />
        <activity
            android:name=".Activities.TiffinPlacesActivity"
            android:label="@string/title_activity_tiffin_places"
            android:theme="@style/Theme.AppCompat.NoActionBar" />
    </application>

</manifest>
```

10.2 Layouts

activity_splash_screen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".Activities.SplashScreenActivity">

    <ImageView
        android:id="@+id/splashScreenImage"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scaleType="centerCrop" />

</LinearLayout>
```

activity_firebase_place.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="nshmadhani.com.wakenbake.Activities.FirebasePlaceActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="@dimen/app_bar_height"
        android:fitsSystemWindows="true"
        android:theme="@style/AppBarOverlay">

        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/toolbar_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed"
            app:toolbarId="@+id/toolbar">

            <ImageView
                android:id="@+id/firebasePlaceImageView"
                android:layout_width="match_parent"
                android:layout_height="match_parent" />

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

    <include layout="@layout/content_firebase_place" />

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

activity_location.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:background="@color/black"
    tools:context=".Activities.LocationActivity">


    <com.skyfishjy.library.RippleBackground
        android:id="@+id/content"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:rb_color="@color/white"
        app:rb_radius="32dp"
        app:rb_rippleAmount="4"
        app:rb_duration="3000"
        app:rb_scale="6">

        <ImageView
            android:id="@+id/locationIcon"
            android:layout_width="64dp"
            android:layout_height="64dp"
            android:layout_centerInParent="true"
            android:src="@drawable/location_pin_icon"/>

        <TextView
            android:id = "@+id/gettingLocation"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Getting Your Location...."
            android:textSize="24sp"
            android:textColor="@color/white"
            android:layout_marginTop="510dp"
            android:layout_marginStart="70dp"
            android:layout_gravity="center" />
    </com.skyfishjy.library.RippleBackground>

</LinearLayout>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="100"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:background="@string/mainAppThemeColor"
        android:layout_weight="30"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="WAKENBAKE"
            android:gravity="center"
            android:textSize="50sp"
            android:textColor="@color/white"
            android:textStyle="bold|italic" />
    </LinearLayout>
    <LinearLayout
        android:id="@+id/tabs_layout"
        android:orientation="vertical"
        android:layout_weight="70"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:background="@color/white">

        <com.ogaclejapan.smarttablayout.SmartTabLayout
            android:id="@+id/viewpagertab"
            android:layout_width="match_parent"
            android:layout_height="48dp"
            app:stl_indicatorAlwaysInCenter="false"
            app:stl_indicatorWithoutPadding="false"
            app:stl_indicatorInFront="false"
            app:stl_indicatorInterpolation="smart"
            app:stl_indicatorGravity="bottom"
            app:stl_indicatorColor="@color/white"
            app:stl_indicatorThickness="4dp"
            app:stl_indicatorWidth="auto"
            app:stl_indicatorCornerRadius="2dp"
            app:stl_overlineColor="@color/white"
            app:stl_overlineThickness="0dp"
            app:stl_underlineColor="@color/white"
            app:stl_underlineThickness="1dp"
            app:stl_dividerColor="@color/white"
            app:stl_dividerThickness="1dp"
            android:background="@string/mainAppThemeColor"
            app:stl_defaultTabTextAllCaps="true"
            app:stl_defaultTabTextColor="@color/white"
            app:stl_defaultTabTextSize="12sp"
            app:stl_defaultTabTextHorizontalPadding="16dp"
            app:stl_defaultTabTextMinWidth="0dp"
            app:stl_distributeEvenly="true"
            app:stl_clickable="true"
            app:stl_titleOffset="24dp"
            app:stl_drawDecorationAfterTab="false" />

        <android.support.v4.view.ViewPager
            android:id="@+id/view_pager"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />
    </LinearLayout>
</LinearLayout>
```

activity_otp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:background="#FFFFFF"
    tools:context=".Activities.OtpActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:background="@string/mainAppThemeColor"
        android:layout_weight="30"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="WAKENBAKE"
            android:gravity="center"
            android:textSize="50sp"
            android:textColor="@color/white"
            android:textStyle="bold|italic" />
    </LinearLayout>

    <LinearLayout
        android:id="@+id/tabs_layout"
        android:orientation="vertical"
        android:layout_weight="70"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:background="@color/white">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Verification"
            android:textColor="@string/mainAppThemeColor"
            android:textSize="24sp"
            android:textStyle="bold"
            android:layout_marginTop="35dp"
            android:layout_marginStart="40dp"/>

        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="45dp"
            android:textColorHint="@color/grey"
            android:textColor="@color/black"
            android:layout_marginTop="25dp"
            android:paddingStart="40dp"
            android:paddingEnd="40dp">
            <EditText
                android:id="@+id/otpPhoneNumber"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:textSize="15sp"
                android:hint="Mobile Number"
                android:textColor="@color/black"
                android:paddingTop="5dp"
                android:inputType="number"
                android:paddingBottom="4dp"
                android:background="@drawable/selector_bg_edit_2"/>
        </android.support.design.widget.TextInputLayout>

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <TextView
                android:id="@+id/otpTextView"
                android:layout_centerVertical="true"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="40dp"
                android:text="OTP"
                android:textColor="#000000"
                android:textSize="24sp"
                android:layout_marginTop="35dp"
                android:textStyle="bold" />


        </RelativeLayout>

        <LinearLayout
            android:layout_marginTop="10dp"
            android:paddingStart="20dp"
            android:paddingEnd="20dp"
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="50dp">
            <com.goodiebag.pinview.Pinview
                android:id="@+id/otpPinView"
                app:pinBackground="@drawable/otpview"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                app:pinWidth="30dp"
                app:pinHeight="40dp"
```

```
                app:pinLength="6"
                app:cursorVisible="false"
                app:forceKeyboard="true"
                app:hint="0"
                app:inputType="text"
                app:password="false"/>


        </LinearLayout>

        <Button
            android:layout_marginTop="45dp"
            android:layout_gravity="center_horizontal"
            android:id="@+id/confirmButton"
            android:backgroundTint="@string/mainAppThemeColor"
            android:background="@drawable/button_shape_white_layout"
            android:stateListAnimator="@drawable/state_list_animator_z"
            android:layout_width="150dp"
            android:layout_height="50dp"
            android:text="Signup"
            android:elevation="8dp"
            android:layout_marginBottom="3dp"
            android:textColor="@color/white">
        </Button>

    </LinearLayout>

</LinearLayout>
```

activity_place.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=".Activities.GooglePlacesActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="@dimen/app_bar_height"
        android:fitsSystemWindows="true"
        android:theme="@style/AppBarOverlay">

        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/toolbar_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed"
            app:toolbarId="@+id/toolbar">

            <ViewFlipper
                android:id="@+id/image_flipper"
                android:layout_width="match_parent"
                android:layout_height="match_parent" />

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

    <include layout="@layout/content_place" />


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

activity_tiffin_places.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="nshmadhani.com.wakenbake.Activities.TiffinPlacesActivity">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="@dimen/app_bar_height"
        android:fitsSystemWindows="true"
        android:theme="@style/AppBarOverlay">

        <android.support.design.widget.CollapsingToolbarLayout
            android:id="@+id/toolbar_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed"
            app:toolbarId="@+id/toolbar">

            <ImageView
                android:id="@+id/tiffinPlaceImageView"
                android:layout_width="match_parent"
                android:layout_height="match_parent" />


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

    <include layout="@layout/content_tiffin_places" />

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

app_bar.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"
    tools:context=".Activities.NavigationActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:theme="@style/AppTheme.AppBarOverlay">

        <com.mancj.materialsearchbar.MaterialSearchBar
            style="@style/MaterialSearchBarLight"
            app:mt_hint="Search"
            app:mt_textColor="@color/white"
            app:mt_hintColor="@color/white"
            app:mt_placeholder="WAKENBAKE"
            app:mt_placeholderColor="@color/white"
            app:mt_searchBarColor="@string/mainAppThemeColor"
            app:mt_searchIconTint="@color/white"
            app:mt_maxSuggestionsCount="10"
            app:mt_navIconEnabled="true"
            app:mt_clearIconTint="@color/white"
            app:mt_backIconTint="@color/white"
            app:mt_navIconTint="@color/white"
            app:mt_navIconUseTint="@color/white"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/searchBar" />

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

    <include layout="@layout/content" />

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

bookmark_cardlayout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_margin="4dp">

    <android.support.v7.widget.CardView
        android:id="@+id/bookmark_card_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:elevation="5dp"
        android:backgroundTint="@color/colorAccent">

        <RelativeLayout
            android:id="@+id/bookmark_placeLinearLayout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
            <ImageView
                android:id="@+id/bookmark_placeImage"
                android:layout_width="match_parent"
                android:layout_height="180dp"
                android:scaleType="centerCrop"
                android:src="@mipmap/ic_launcher"
                android:contentDescription="Image" />

            <TextView
                android:id="@+id/bookmark_placeName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@id/placeImage"
                android:layout_margin="5dp"
                android:textSize="20sp"
                android:textColor="@color/purple"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/bookmark_time_stamp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_alignParentRight="true"
                android:gravity="right"
                android:textSize="14sp"
                android:layout_marginTop="220dp"
                android:layout_marginBottom="12dp"
                android:layout_marginRight="4dp"
                android:textColor="@color/black"
                android:text="2 hours ago"/>
        </RelativeLayout>
    </android.support.v7.widget.CardView>

</LinearLayout>
```

content.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:background="@string/mainAppThemeColor"
    android:orientation="vertical"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".Activities.NavigationActivity"
    tools:showIn="@layout/app_bar">

    <com.ogaclejapan.smarttablayout.SmartTabLayout
        android:id="@+id/navviewpagertab"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:background="@string/mainAppThemeColor"
        app:stl_indicatorAlwaysInCenter="false"
        app:stl_indicatorWithoutPadding="false"
        app:stl_indicatorInFront="false"
        app:stl_indicatorInterpolation="smart"
        app:stl_indicatorGravity="bottom"
        app:stl_indicatorColor="@color/white"
        app:stl_indicatorThickness="4dp"
        app:stl_indicatorWidth="auto"
        app:stl_indicatorCornerRadius="2dp"
        app:stl_overlineColor="@color/white"
        app:stl_overlineThickness="0dp"
        app:stl_underlineColor="@color/white"
        app:stl_underlineThickness="1dp"
        app:stl_dividerColor="@color/white"
        app:stl_dividerThickness="1dp"
        app:stl_defaultTabTextAllCaps="true"
        app:stl_defaultTabTextColor="@color/white"
        app:stl_defaultTabTextSize="12sp"
        app:stl_defaultTabTextHorizontalPadding="16dp"
        app:stl_defaultTabTextMinWidth="0dp"
        app:stl_distributeEvenly="true"
        app:stl_clickable="true"
        app:stl_titleOffset="24dp"
        app:stl_drawDecorationAfterTab="false" />

    <android.support.v4.view.ViewPager
        android:id="@+id/nav_view_pager"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>
```

content_firebase_place.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.NestedScrollView
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:background="@color/white"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"

tools:context="nshmadhani.com.wakenbake.Activities.FirebasePlaceActivity"
    tools:showIn="@layout/activity_firebase_place">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center_horizontal"
        android:orientation="vertical">

        <TextView
            android:id="@+id/night_placeNameTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_margin="5dp"
            android:text="Place Name"
            android:textColor="@color/black"
            android:textSize="22sp"
            android:textStyle="bold">
        </TextView>
        <com.willy.ratingbar.ScaleRatingBar
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:id="@+id/night_simpleRatingBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            app:srb_numStars="5"
            app:srb_minimumStars="0"
            app:srb_starWidth="30dp"
            app:srb_starHeight="30dp"
            app:srb_starPadding="10dp"
            app:srb_stepSize="0.5"
            app:srb_isIndicator="false"
            app:srb_clickable="true"
            app:srb_scrollable="false"
            app:srb_clearRatingEnabled="true">
        </com.willy.ratingbar.ScaleRatingBar>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">

            <android.support.design.widget.FloatingActionButton
                android:id="@+id/night_callButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="@dimen/fab_margin"
                android:src="@drawable/ic_dialog_phone" />

            <android.support.design.widget.FloatingActionButton
                android:id="@+id/night_bookmarkButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="@dimen/fab_margin"
                android:src="@drawable/ic_action_bookmark" />

            <android.support.design.widget.FloatingActionButton
                android:id="@+id/night_mapsButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="@dimen/fab_margin"
                android:src="@drawable/ic_action_maps" />

        </LinearLayout>

        <FrameLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:orientation="vertical">

                    <TextView
                        android:id="@+id/night_address"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginTop="20dp"
                        android:layout_marginStart="30dp"
                        android:text="Address:"
                        android:textColor="@color/black"
                        android:textSize="18sp"
                        android:textStyle="bold" />

                    <TextView
                        android:id="@+id/night_foodType"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginTop="30dp"
                        android:layout_marginStart="30dp"
                        android:text="Cuisine:"
                        android:textColor="@color/black"
                        android:textSize="18sp"
                        android:textStyle="bold" />

                    <TextView
                        android:id="@+id/night_time"
                        android:layout_marginTop="30dp"
                        android:layout_marginBottom="30dp"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="30dp"
                        android:text="Time:"
                        android:textColor="@color/black"
                        android:textSize="18sp"
                        android:textStyle="bold" />
                </LinearLayout>

        </FrameLayout>

        <TextView
            android:id="@+id/reviewHeading"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="35dp"
            android:layout_marginStart="40dp"
            android:text="Reviews"
            android:textColor="@string/mainAppThemeColor"
            android:textSize="24sp"
            android:textStyle="bold" />

        <android.support.v7.widget.RecyclerView
            android:id="@+id/mReviewRecyclerView"
            android:layout_width="match_parent"
            android:layout_marginStart="40dp"
            android:layout_marginTop="5dp"
            android:layout_height="wrap_content" />

        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColorHint="@color/grey"
            android:textColor="@color/black"
            android:layout_marginTop="25dp"
            android:paddingStart="20dp"
            android:paddingEnd="20dp">
```

```
            <EditText
                android:id="@+id/reviewEditText"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Enter Review"
                android:textSize="15sp"
                android:textColor="@color/black"
                android:paddingTop="5dp"
                android:gravity="top|left"
                android:inputType="textMultiLine"
                android:singleLine="false"
                android:paddingBottom="4dp"
                android:background="@drawable/selector_bg_edit_2"/>

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

        <Button
            android:id="@+id/submitButton"
            android:layout_width="150dp"
            android:layout_height="50dp"
            android:layout_marginTop="45dp"
            android:layout_gravity="center_horizontal"
            android:backgroundTint="@string/mainAppThemeColor"
            android:background="@drawable/button_shape_white_layout"
            android:stateListAnimator="@drawable/state_list_animator_z"
            android:text="Submit"
            android:elevation="8dp"
            android:layout_marginBottom="3dp"
            android:textColor="@color/white">
        </Button>

        <fragment
xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:id="@+id/night_map"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:layout_gravity="center|end"
            android:layout_marginTop="200dp"
            android:layout_margin="10dp"
            android:name="com.google.android.gms.maps.SupportMapFragment"
            tools:context=".Activities.GooglePlacesActivity" />

    </LinearLayout>
</android.support.v4.widget.NestedScrollView>
```

content_place.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.NestedScrollView
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:background="@color/white"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context=".Activities.GooglePlacesActivity"
    tools:showIn="@layout/activity_place">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_gravity="center_horizontal">

        <TextView
            android:id="@+id/placeNameTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Place Name"
            android:textColor="@color/black"
            android:textSize="22sp"
            android:textStyle="bold"
            android:layout_margin="5dp"
            android:layout_gravity="center_horizontal">
        </TextView>

        <com.willy.ratingbar.ScaleRatingBar
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:id="@+id/simpleRatingBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            app:srb_numStars="5"
            app:srb_minimumStars="0"
            app:srb_starWidth="30dp"
            app:srb_starHeight="30dp"
            app:srb_starPadding="10dp"
            app:srb_stepSize="0.5"
            app:srb_isIndicator="false"
            app:srb_clickable="false"
            app:srb_scrollable="false"
            app:srb_clearRatingEnabled="true">
        </com.willy.ratingbar.ScaleRatingBar>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">


    <android.support.design.widget.FloatingActionButton
        android:id="@+id/callButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/fab_margin"
        android:src="@drawable/ic_dialog_phone" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/bookmarkButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/fab_margin"
        android:src="@drawable/ic_action_bookmark" />

    <android.support.design.widget.FloatingActionButton
        android:id="@+id/mapsButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/fab_margin"
        android:src="@drawable/ic_action_maps" />

    </LinearLayout>

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="vertical">

                <TextView
                    android:id="@+id/address"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="20dp"
                    android:layout_marginStart="30dp"
                    android:text="Address:"
                    android:textColor="@color/black"
                    android:textSize="18sp"
                    android:textStyle="bold" />

                <TextView
                    android:id="@+id/website"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="30dp"
                    android:layout_marginStart="30dp"
                    android:text="Website:"
                    android:textColor="@color/black"
                    android:textSize="18sp"
                    android:textStyle="bold" />


                <TextView
                    android:id="@+id/time"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_marginTop="30dp"
                    android:layout_marginBottom="30dp"
                    android:layout_marginStart="30dp"
                    android:text="Time - 11:00AM - 11:00PM"
                    android:textColor="@color/black"
                    android:textSize="18sp"
                    android:textStyle="bold" />
            </LinearLayout>
    </FrameLayout>

    <fragment xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="200dp"
        android:layout_gravity="center|end"
        android:layout_marginTop="200dp"
        android:layout_margin="5dp"
        tools:context=".Activities.GooglePlacesActivity"
        android:name="com.google.android.gms.maps.SupportMapFragment"/>

    </LinearLayout>
</android.support.v4.widget.NestedScrollView>
```

content_tiffin_places.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.NestedScrollView
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:background="@color/white"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"

tools:context="nshmadhani.com.wakenbake.Activities.TiffinPlacesActivity"
    tools:showIn="@layout/activity_tiffin_places">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_gravity="center_horizontal">

        <TextView
            android:id="@+id/tiffin_placeNameTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Place Name"
            android:textColor="@color/black"
            android:textSize="22sp"
            android:textStyle="bold"
            android:layout_gravity="center_horizontal"
            android:layout_margin="5dp">
        </TextView>

        <com.willy.ratingbar.ScaleRatingBar
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:id="@+id/tiffin_simpleRatingBar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            app:srb_numStars="5"
            app:srb_minimumStars="0"
            app:srb_starWidth="30dp"
            app:srb_starHeight="30dp"
            app:srb_starPadding="10dp"
            app:srb_stepSize="0.5"
            app:srb_isIndicator="false"
            app:srb_clickable="true"
            app:srb_scrollable="false"
            app:srb_clearRatingEnabled="true">
        </com.willy.ratingbar.ScaleRatingBar>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:orientation="horizontal">

            <android.support.design.widget.FloatingActionButton
                android:id="@+id/tiffin_callButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="@dimen/fab_margin"
                android:src="@drawable/ic_dialog_phone" />

            <android.support.design.widget.FloatingActionButton
                android:id="@+id/tiffin_bookmarkButton"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_margin="@dimen/fab_margin"
                android:src="@drawable/ic_action_bookmark" />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <TextView
                android:id="@+id/tiffin_foodType"
                android:layout_marginTop="30dp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginStart="30dp"
                android:text="Cuisine:"
                android:textColor="@color/white"
                android:textSize="18sp"
                android:textStyle="bold" />
        </LinearLayout>
        <TextView
            android:id="@+id/reviewHeading"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="35dp"
            android:text="Reviews"
            android:textColor="@string/mainAppThemeColor"
            android:textSize="24sp"
            android:textStyle="bold" />

        <android.support.v7.widget.RecyclerView
            android:id="@+id/mReviewRecyclerView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textColorHint="@color/grey"
            android:textColor="@color/black"
            android:layout_marginTop="25dp"
            android:paddingStart="20dp"
            android:paddingEnd="20dp">

            <EditText
                android:id="@+id/reviewEditText"
```

```
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Enter Review"
                android:textSize="15sp"
                android:textColor="@color/black"
                android:paddingTop="5dp"
                android:gravity="top|left"
                android:inputType="textMultiLine"
                android:singleLine="false"
                android:paddingBottom="4dp"
                android:background="@drawable/selector_bg_edit_2"/>

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

        <Button
            android:id="@+id/submitButton"
            android:layout_width="150dp"
            android:layout_height="50dp"
            android:layout_marginTop="45dp"
            android:layout_gravity="center_horizontal"
            android:backgroundTint="@string/mainAppThemeColor"
            android:background="@drawable/button_shape_white_layout"
            android:stateListAnimator="@drawable/state_list_animator_z"
            android:text="Submit"
            android:elevation="8dp"
            android:layout_marginBottom="3dp"
            android:textColor="@color/white">
        </Button>
    </LinearLayout>
</android.support.v4.widget.NestedScrollView>
```

firebase_places_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/firebasePlacesRecyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

google_places_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/googlePlacesFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

list_places.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:background="#fff"
    android:layout_height="wrap_content">

    <android.support.v7.widget.CardView
        android:id="@+id/list_card_view"
        android:layout_width="match_parent"
        android:layout_height="90dp"
        android:layout_gravity="center"
        android:layout_margin="10dp"
        android:background="#fff"
        android:elevation="3dp">

        <LinearLayout
            android:id="@+id/placeLinearLayout"
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:background="#fff"
            android:layout_height="wrap_content">

            <ImageView
                android:id="@+id/listPlaceImage"
                android:layout_width="0dp"
                android:layout_weight="1"
                android:layout_height="70dp"
                android:layout_margin="8dp"
                android:scaleType="fitCenter"
                android:src="@mipmap/ic_launcher"
                android:contentDescription="Image" />

            <TextView
                android:id="@+id/listPlaceName"
                android:layout_width="0dp"
                android:layout_weight="3"
                android:layout_height="wrap_content"
                android:layout_below="@id/placeImage"
                android:layout_margin="9dp"
                android:textSize="20sp"
                android:textColor="#000"
                android:textAlignment="center"/>

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

login_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView android:layout_height="match_parent"
    android:layout_width="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <!--Login Text-->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:textColor="@string/mainAppThemeColor"
        android:textSize="24sp"
        android:textStyle="bold"
        android:layout_marginTop="35dp"
        android:layout_marginStart="40dp"/>

    <!--Username-->
    <android.support.design.widget.TextInputLayout
        android:id="@+id/username_til"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:textColorHint="@color/grey"
        android:textColor="@color/black"
        android:layout_marginTop="25dp"
        android:paddingStart="40dp"
        android:paddingEnd="40dp">
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Username"
            android:textSize="15sp"
            android:textColor="@color/black"
            android:paddingTop="5dp"
            android:inputType="textPersonName"
            android:paddingBottom="4dp"
            android:background="@drawable/selector_bg_edit_2"/>
    </android.support.design.widget.TextInputLayout>

    <!--Password-->
    <android.support.design.widget.TextInputLayout
        android:id="@+id/password_til"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:layout_marginTop="25dp"
        android:paddingEnd="40dp"
        android:paddingStart="40dp"
        android:textColorHint="@color/grey">
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@drawable/selector_bg_edit_2"
            android:hint="Password"
            android:inputType="textPassword"
            android:paddingBottom="4dp"
            android:paddingTop="5dp"
            android:textColor="@color/black"
            android:textSize="15sp"

android:textCursorDrawable="@drawable/yellow_gradient_background"/>
    </android.support.design.widget.TextInputLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="45dp"
    android:gravity="center">

    <!--Login Button-->
    <Button
        android:id="@+id/login_bt"
        android:layout_width="130dp"
        android:layout_height="50dp"
        android:text="Login"
        android:textColor="@color/white"
        android:backgroundTint="@string/mainAppThemeColor"
        android:background="@drawable/button_shape_white_layout"
        android:stateListAnimator="@drawable/state_list_animator_z">
    </Button>

    <!-- Google Sign In Button-->
    <com.google.android.gms.common.SignInButton
        android:id="@+id/googleButton"
        android:elevation="10dp"
        android:layout_marginLeft="20dp"
        android:layout_width="130dp"
        android:layout_height="55dp" />

       </LinearLayout>
    </LinearLayout>
</ScrollView>
```

navigation_drawer.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
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/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        layout="@layout/app_bar"/>

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        android:background="@string/mainAppThemeColor"
        app:itemTextColor="@color/white"
        app:headerLayout="@layout/navigation_drawer_header"
        app:menu="@menu/activity_navigation_drawer" />

</android.support.v4.widget.DrawerLayout>
```

navigation_drawer_header.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="@dimen/nav_header_height"
    android:background="@string/mainAppThemeColor"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/mNavHeaderImage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        app:srcCompat="@mipmap/user_image_asset_round" />

    <TextView
        android:id="@+id/mNavHeaderEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="support@wakenbake.com"
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

</LinearLayout>
```

place_card.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:layout_margin="4dp">

    <android.support.v7.widget.CardView
        android:id="@+id/card_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:elevation="5dp"
        android:backgroundTint="@color/colorAccent">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <ImageView
                android:id="@+id/placeImage"
                android:layout_width="match_parent"
                android:layout_height="180dp"
                android:scaleType="centerCrop"
                android:src="@mipmap/ic_launcher"
                android:contentDescription="Image" />

            <TextView
                android:id="@+id/placeName"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_below="@id/placeImage"
                android:layout_margin="5dp"
                android:textSize="20sp"
                android:textColor="@color/purple"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/time_stamp"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="2 hours ago"
                android:textColor="@color/black"
                android:textSize="14sp"
                android:gravity="right"
                android:layout_alignParentRight="true"
                android:layout_marginTop="220dp"
                android:layout_marginBottom="12dp"
                android:layout_marginRight="4dp"/>
        </RelativeLayout>
    </android.support.v7.widget.CardView>
</LinearLayout>
```

review_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:id="@+id/mRecyclerViewEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="abc@gmail.com"
        android:layout_margin="5dp"
        android:textSize="20sp"
        android:textColor="@color/black"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/mRecyclerViewReview"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nice!"
        android:textSize="18sp"
        android:layout_margin="3dp"
        android:textColor="@color/black"/>

</LinearLayout>
```

signup_fragment.xml

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

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TextView
            android:layout_marginTop="35dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="40dp"
            android:text="@string/register"
            android:textColor="@string/mainAppThemeColor"
            android:textSize="24sp"
            android:textStyle="bold" />

         <!--Username Edit Text-->
        <android.support.design.widget.TextInputLayout
            android:id="@+id/signupUsername_til"
            android:textColor="@color/black"
            android:layout_marginTop="25dp"
            android:paddingStart="40dp"
            android:paddingEnd="40dp"
            android:layout_width="match_parent"
            android:layout_height="45dp" >
            <EditText
                android:textSize="15sp"
                android:hint="@string/Username"
                android:textColor="#000000"

                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:inputType="textPersonName"
                android:background="@drawable/selector_bg_edit_2"

android:textCursorDrawable="@drawable/yellow_gradient_background"
                android:paddingBottom="4dp" />
        </android.support.design.widget.TextInputLayout>

        <!-- Email Edit Text-->
        <android.support.design.widget.TextInputLayout
            android:id="@+id/signupEmail_til"
            android:textColorHint="@color/grey"
            android:layout_marginTop="25dp"
            android:paddingStart="40dp"
            android:paddingEnd="40dp"
            android:layout_width="match_parent"
            android:layout_height="45dp">
            <EditText
                android:textSize="15sp"
                android:hint="Email"
                android:textColorHint="@color/grey"
                android:textColor="@color/black"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:inputType="textEmailAddress"
                android:background="@drawable/selector_bg_edit_2"

android:textCursorDrawable="@drawable/yellow_gradient_background"
                android:paddingBottom="4dp"  />
        </android.support.design.widget.TextInputLayout>

        <!--Password Edit Text-->
        <android.support.design.widget.TextInputLayout
            android:id="@+id/signupPassword_til"
            android:textColorHint="@color/grey"
            android:layout_marginTop="25dp"
            android:paddingStart="40dp"
            android:paddingEnd="40dp"
            android:orientation="horizontal"
            android:layout_width="match_parent"
            android:layout_height="45dp">
            <EditText
                android:textSize="15sp"
                android:hint="Password"
                android:textColor="@color/black"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:paddingTop="5dp"
                android:inputType="textPassword"
                android:background="@drawable/selector_bg_edit_2"

android:textCursorDrawable="@drawable/yellow_gradient_background"
                android:paddingBottom="4dp" />
        </android.support.design.widget.TextInputLayout>

        <!--Sign Up Button -->
        <Button
            android:layout_marginTop="45dp"
            android:layout_gravity="center_horizontal"
            android:id="@+id/signupButton"
            android:backgroundTint="@string/mainAppThemeColor"
            android:background="@drawable/button_shape_white_layout"
            android:stateListAnimator="@drawable/state_list_animator_z"
            android:layout_width="150dp"
            android:layout_height="50dp"
            android:text="Signup"
            android:elevation="8dp"
            android:layout_marginBottom="3dp"
            android:textColor="@color/white">
        </Button>
    </LinearLayout>
</ScrollView>
```

tiffin_places_fragment.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/tiffinplacesRecyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

10.3 Java Code

AppDatabase.java

```
package nshmadhani.com.wakenbake.Abstracts;

import android.arch.persistence.room.Database;
import android.arch.persistence.room.RoomDatabase;

import nshmadhani.com.wakenbake.Interfaces.IDoa;
import nshmadhani.com.wakenbake.Models.PlaceBookmark;


@Database(entities = {PlaceBookmark.class}, version = 1, exportSchema =
false)
public abstract class AppDatabase extends RoomDatabase {

    public abstract IDoa iDoa();
}
```

BookmarkActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;

import java.util.List;

import nshmadhani.com.wakenbake.Adapters.PlaceBookmarkAdapter;
import nshmadhani.com.wakenbake.Models.PlaceBookmark;
import nshmadhani.com.wakenbake.Models.WakeNBake;
import nshmadhani.com.wakenbake.R;

public class BookmarkActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private PlaceBookmarkAdapter adapter;
    private List<PlaceBookmark> list;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bookmark);

        recyclerView  = findViewById(R.id.bookmarkRecyclerView);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));

        list = WakeNBake.database.iDoa().fetchAllPlaces();

        for (PlaceBookmark p : list) {

            p.setPlaceNAME(p.getPlaceNAME());
            p.setPlaceURL(p.getPlaceURL());

        }

        adapter = new PlaceBookmarkAdapter(list, this);
        recyclerView.setAdapter(adapter);
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }
}
```

FirebasePlaceActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.net.Uri;
import android.net.wifi.p2p.WifiP2pManager;
import android.os.Build;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.firebase.auth.FirebaseAuth;
import com.squareup.picasso.Picasso;
import com.willy.ratingbar.BaseRatingBar;
import com.willy.ratingbar.ScaleRatingBar;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.util.Locale;

import nshmadhani.com.wakenbake.Adapters.ReviewFragmentListAdapter;
import nshmadhani.com.wakenbake.Holders.ReviewResponse;
import nshmadhani.com.wakenbake.Models.PlaceBookmark;
import nshmadhani.com.wakenbake.Models.Review;
import nshmadhani.com.wakenbake.Models.WakeNBake;
import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Interfaces.IRetrofitDataApi;
import nshmadhani.com.wakenbake.Models.APIClient;
import nshmadhani.com.wakenbake.Holders.RatingsResponse;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class FirebasePlaceActivity extends AppCompatActivity implements
OnMapReadyCallback {

    private static final String TAG =
FirebasePlaceActivity.class.getSimpleName();
    public ScaleRatingBar mVendorRatingBar;
    public String mVendorPhoneNumber = "";
    public FloatingActionButton mVendorCallButton;
    public FloatingActionButton mVendorBookmarkButton;
    public FloatingActionButton location;
    public ImageView mVendorImageView;
    public TextView name;
    public TextView mVendorAddress;
    public TextView mVendorFoodItems;
    public String number;
    public TextView mVendorTime;
    public IRetrofitDataApi apiInterface;
    public double newRatings;
    public float oldRatings;
    private List<Review> mReviewList;
    public RecyclerView mReviewsRecyclerView;
    public ReviewFragmentListAdapter mReviewsListAdapter;
    private TextView mReviewHeading;
    private Button mSubmitButton;
    private EditText mReviewBody;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_firebase_place);
        mVendorImageView = findViewById(R.id.firebasePlaceImageView);
        mVendorRatingBar = findViewById(R.id.night_simpleRatingBar);
        mVendorTime = findViewById(R.id.night_time);

        Picasso.with(this)
                .load(getIntent().getStringExtra("vendor_url"))
                .into(mVendorImageView); //Setting the vendor's image

        name = findViewById(R.id.night_placeNameTextView);
        mVendorAddress = findViewById(R.id.night_address);
        mVendorCallButton = findViewById(R.id.night_callButton);
        mVendorBookmarkButton = findViewById(R.id.night_bookmarkButton);
        location = findViewById(R.id.night_mapsButton);

        mVendorFoodItems = findViewById(R.id.night_foodType);
        mVendorFoodItems.setText("Food Items: " +
getIntent().getStringExtra("vendor_food"));
        //Setting the food items available at the vendor


        mVendorTime.setText(String.format("Time: %dAM TO %dAM",
                getIntent().getIntExtra("vendor_open", 0),
                getIntent().getIntExtra("vendor_close", 0)));
        //Setting the vendor's time

        apiInterface =
APIClient.getClient().create(IRetrofitDataApi.class);

        name.setText(getIntent().getStringExtra("vendor_name")); //Setting
vendor's name
        mVendorRatingBar.setRating(
getIntent().getFloatExtra("vendor_ratings", 0f));
        //Setting the initial ratings of the vendor

        double latitude = getIntent().getDoubleExtra("vendor_lat", 0);
        double longitude = getIntent().getDoubleExtra("vendor_lng", 0);

        getAddress(latitude, longitude); //Getting the address of the
vendor using its latitude and longitude

        mVendorPhoneNumber = getIntent().getStringExtra("vendor_phone");

        apiInterface =
APIClient.getClient().create(IRetrofitDataApi.class);

        // Get the SupportMapFragment and request notification
        // when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.night_map);
        mapFragment.getMapAsync(this);

        //While clicking on the ratings
        mVendorRatingBar.setOnRatingChangeListener(new
BaseRatingBar.OnRatingChangeListener() {
            @Override
            public void onRatingChange(BaseRatingBar baseRatingBar, float
v) {
                mVendorRatingBar.setClickable(false); //First set the
ratings to non-clickable
                oldRatings = mVendorRatingBar.getRating(); //Get the old
ratings
                String vendorName = name.getText().toString(); // Get the
vendors name
                double mNewRatings = sendRatings(vendorName,v +
oldRatings); // Getting the new ratings + old ratings
                mVendorRatingBar.setRating((float)(mNewRatings +
oldRatings)); //Setting the new ratings
            }
        });

        mVendorCallButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                callPhone(mVendorPhoneNumber); //Call the vendor
            }
        });

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

                String id = getIntent().getStringExtra("vendor_id");
                String name = getIntent().getStringExtra("vendor_name");
                String url = getIntent().getStringExtra("vendor_url");
                PlaceBookmark placeBookmark = new PlaceBookmark(id, name,
url);

                String foundPlace =
WakeNBake.database.iDoa().checkInDatabase(id);

                if (name.equalsIgnoreCase(foundPlace)) {
                    Toast.makeText(FirebasePlaceActivity.this, "Already
added to bookmarks!", Toast.LENGTH_SHORT).show();
                }
                else {
                    WakeNBake.database.iDoa().addPlace(placeBookmark);
                    Toast.makeText(FirebasePlaceActivity.this, "Bookmark
added!", Toast.LENGTH_SHORT).show();
                }
            }
        });

        mReviewHeading = findViewById(R.id.reviewHeading);
        mReviewBody = findViewById(R.id.reviewEditText);
        mSubmitButton = findViewById(R.id.submitButton);

        mReviewList = new ArrayList<>();
        mReviewsRecyclerView  = findViewById(R.id.mReviewRecyclerView);
        mReviewsRecyclerView.setLayoutManager(new
LinearLayoutManager(this));
        mReviewsListAdapter = new
ReviewFragmentListAdapter(mReviewList,this);
        mReviewsRecyclerView.setAdapter(mReviewsListAdapter);

        final String mVendorName =
getIntent().getStringExtra("vendor_name");

        getReviews(mVendorName);

        final String mUsername =
FirebaseAuth.getInstance().getCurrentUser().getEmail();

        mSubmitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                FirebasePlaceActivity.this.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        saveReview(mUsername,
mReviewBody.getText().toString(), mVendorName);
                    }
                });
            }
        });

    }

    private void getReviews(String mVendorName) {

        Call<ReviewResponse> call = apiInterface.getReviews(mVendorName);
        call.enqueue(new Callback<ReviewResponse>() {
            @Override
            public void onResponse(@NonNull Call<ReviewResponse> call,
@NonNull Response<ReviewResponse> response) {
                List<Review> reviewResponseList =
response.body().getmReviewData();
                for (Review r : reviewResponseList) {
                    Review review = new Review();
                    review.setmUsernameReview(r.getmUsernameReview());
                    review.setmReview(r.getmReview());

                    mReviewList.add(r);
                }

                Log.d(TAG, "onResponse: list size: " + mReviewList.size());

                FirebasePlaceActivity.this.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        mReviewsListAdapter.notifyDataSetChanged();
                    }
                });
            }

            @Override
            public void onFailure(Call<ReviewResponse> call, Throwable t) {
                Log.d(TAG, "onFailure: " + t);
            }
        });

    }

    private void saveReview(String mUsername, String s, String mVendorName)
{
        Call<ResponseBody> saveCall = apiInterface.saveReviews(s,
mUsername, mVendorName);
        saveCall.enqueue(new Callback<ResponseBody>() {
            @Override
            public void onResponse(@NonNull Call<ResponseBody> call,
@NonNull Response<ResponseBody> response) {
                Log.d(TAG, "onResponse: " + response.body());
                if (response.isSuccessful()) {
                    Toast.makeText(FirebasePlaceActivity.this, "Review
added!", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(FirebasePlaceActivity.this, "Server
Error!", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(@NonNull Call<ResponseBody> call,
@NonNull Throwable t) {
                Log.d(TAG, "onFailure: " + t);
            }
        });
    }

    private double sendRatings(String name, float v) {
        //Sending the ratings to the firebase real-time database
        Call<RatingsResponse> call = apiInterface.saveRatings(name, v);
//Creating a request call to the firebase
        call.enqueue(new Callback<RatingsResponse>() {
            @Override
            public void onResponse(@NonNull Call<RatingsResponse> call,
@NonNull Response<RatingsResponse> response) {
             List<Double> r = response.body().getRatings(); //Getting
response in form of JSON
                newRatings = r.get(0); //Getting the first element
            }

            @Override
            public void onFailure(@NonNull Call<RatingsResponse> call,
@NonNull Throwable t) {
                Log.d(TAG, "onFailure: server: " +
t.getLocalizedMessage());
            }
        });

        return newRatings; //Returning the new ratings
    }

    public String getVendorName () {
        return name.getText().toString();
    }

    private void getAddress(double latitude, double longitude) {
        //Getting address of the location
        Geocoder geocoder = new Geocoder(this, Locale.ENGLISH); //Creating
a geocoder object
        try {
        List<Address> addresses = geocoder.getFromLocation(latitude,
longitude, 1);
            String address = addresses.get(0).getAddressLine(0);
            // If any additional address line present than only,
            // check with max available address lines by
getMaxAddressLineIndex()

            mVendorAddress.setText("Address: " + address);

        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    private void callPhone(String phoneNumber) {
        Intent intent = new Intent(Intent.ACTION_DIAL); //Redirecting to
the Dialer Application
        intent.setData(Uri.parse("tel: " + phoneNumber)); //Getting the
phone number

        if (Build.VERSION.SDK_INT >= 23 ) { //If higher than Android M
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                return;
                //Checking for the permissions
            }
            startActivity(intent); //Start Dialer application
        }
        else {  //If lower than Android M
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                return;
                //Check for permissions
            }
            startActivity(intent); //Start Dialer application
        }
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            return;
            //Checking for permissions
        }
        googleMap.setMyLocationEnabled(true); //Setting user's current
location in blue dot

        LatLng latLng = new
LatLng(getIntent().getExtras().getDouble("vendor_lat"),
                getIntent().getExtras().getDouble("vendor_lng"));
        //Creating a location using its latitude and longitude

        //Adding place name to the place marker
        googleMap.addMarker(new MarkerOptions().position(latLng)

.title(getIntent().getExtras().getString("vendor_name"))).showInfoWindow();

        //Adding zoom to the maps
        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
14.0f));
    }
}
```

GooglePlacesActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ViewFlipper;

import com.google.android.gms.location.places.GeoDataClient;
import com.google.android.gms.location.places.Place;
import com.google.android.gms.location.places.PlaceBufferResponse;
import com.google.android.gms.location.places.PlacePhotoMetadata;
import com.google.android.gms.location.places.PlacePhotoMetadataBuffer;
import com.google.android.gms.location.places.PlacePhotoMetadataResponse;
import com.google.android.gms.location.places.PlacePhotoResponse;
import com.google.android.gms.location.places.Places;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.willy.ratingbar.ScaleRatingBar;

import nshmadhani.com.wakenbake.Models.PlaceBookmark;
import nshmadhani.com.wakenbake.Models.WakeNBake;
import nshmadhani.com.wakenbake.R;

public class GooglePlacesActivity extends AppCompatActivity implements
OnMapReadyCallback {

    private static final String TAG =
GooglePlacesActivity.class.getSimpleName();
    private GeoDataClient mGeoDataClient;
    public ViewFlipper imageFlipper;
    public TextView placeName;
    public ScaleRatingBar ratingBar;
    public String phoneNumber = "";
    public TextView address;
    public TextView website;
    public FloatingActionButton call;
    public FloatingActionButton bookmark;
    public FloatingActionButton location;

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

        mGeoDataClient = Places.getGeoDataClient(this, null);
        String placeId = getIntent().getStringExtra("placeId");
        imageFlipper = findViewById(R.id.image_flipper);
        ratingBar = findViewById(R.id.simpleRatingBar);
        ratingBar.setRating((float)
getIntent().getDoubleExtra("placeRatings", 0));
        placeName = findViewById(R.id.placeNameTextView);
        address = findViewById(R.id.address);
        website = findViewById(R.id.website);

        //SugarContext.init(this);

        ratingBar.setClickable(false);
        placeName.setText(getIntent().getStringExtra("placeName"));

        getDetailsOfPlaces(placeId);
        getPhotosOfPlaces(placeId);

        // Get the SupportMapFragment and request notification
        // when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

        call = findViewById(R.id.callButton);
        call.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                callPlace(phoneNumber); //Call to the place using its phone
number
                Log.d(TAG, "onClick: call: ");

Toast.makeText(GooglePlacesActivity.this,"Clicked",Toast.LENGTH_SHORT).show
();
            }
        });

        location = findViewById(R.id.mapsButton);
        location.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(GooglePlacesActivity.this, "Slide to Bottom
for location", Toast.LENGTH_SHORT).show();
            }
        });

        bookmark = findViewById(R.id.bookmarkButton);
        bookmark.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Adding to the database
                String id = getIntent().getStringExtra("placeId");
                String name = getIntent().getStringExtra("placeName");
                String url = getIntent().getStringExtra("placeUrl");
                PlaceBookmark placeBookmark = new PlaceBookmark(id, name,
url);
```

```
        String foundPlace =
WakeNBake.database.iDoa().checkInDatabase(id);

        if (name.equalsIgnoreCase(foundPlace)) {
          Toast.makeText(GooglePlacesActivity.this, "Already
added to bookmarks!", Toast.LENGTH_SHORT).show();
        }
        else {
          WakeNBake.database.iDoa().addPlace(placeBookmark);
          Toast.makeText(GooglePlacesActivity.this, "Bookmark
added!", Toast.LENGTH_SHORT).show();
        }

        Toast.makeText(GooglePlacesActivity.this, "Bookmark
added!", Toast.LENGTH_SHORT).show();
      }
    });

  }

  private void callPlace(String phoneNumber) {
    Intent intent = new Intent(Intent.ACTION_DIAL); //Redirecting to
the Dialer Application
    intent.setData(Uri.parse("tel: " + phoneNumber)); //Getting the
phone number
    if (Build.VERSION.SDK_INT >= 23 ) { //If higher than Android M
      if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
        return;
        //Checking for the permissions
      }
      startActivity(intent); //Start Dialer application
    }
    else { //If lower than Android M
      if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
        return;
        //Check for permissions
      }
      startActivity(intent); //Start Dialer application
    }
  }

  private void getDetailsOfPlaces(String placeId) {
    //Getting place details using its ID from Google Places API
    mGeoDataClient.getPlaceById(placeId).addOnCompleteListener(new
OnCompleteListener<PlaceBufferResponse>() {
      @Override
      public void onComplete(@NonNull Task<PlaceBufferResponse> task)
{
        if (task.isSuccessful()) {
          PlaceBufferResponse places = task.getResult();
          Place myPlace = places.get(0); //Got the place
          address.setText("Address: " + myPlace.getAddress());
//Setting address
          if (myPlace.getWebsiteUri() == null)
            website.setText("Website: No Website Yet");
          else
            website.setText("Website: " +
myPlace.getWebsiteUri()); //Setting website
```

```
                    phoneNumber +=
String.valueOf(myPlace.getPhoneNumber());
                    places.release();
                } else {
                    Log.e(TAG, "Place not found.");
                }
            }
        });

    }

    private void getPhotosOfPlaces(String placeId) {

        //Getting more than one photo of a place using its ID

        final Task<PlacePhotoMetadataResponse> photoMetadataResponse =
mGeoDataClient.getPlacePhotos(placeId);
            photoMetadataResponse.addOnCompleteListener(new
OnCompleteListener<PlacePhotoMetadataResponse>() {
                @Override
                public void onComplete(@NonNull
Task<PlacePhotoMetadataResponse> task) {
                    // Get the list of photos.
                    PlacePhotoMetadataResponse photos = task.getResult();
                    // Get the PlacePhotoMetadataBuffer (metadata for all
of the photos).
                    PlacePhotoMetadataBuffer photoMetadataBuffer =
photos.getPhotoMetadata();
                    for (int i = 0; i < photoMetadataBuffer.getCount();
i++) {
                    // Get the first photo in the list.
                    PlacePhotoMetadata photoMetadata =
photoMetadataBuffer.get(i);
                    // Get a full-size bitmap for the photo.
                    Task<PlacePhotoResponse> photoResponse =
mGeoDataClient.getPhoto(photoMetadata);
                    photoResponse.addOnCompleteListener(new
OnCompleteListener<PlacePhotoResponse>() {
                        @Override
                        public void onComplete(@NonNull
Task<PlacePhotoResponse> task) {
                            PlacePhotoResponse photo = task.getResult();
                            Bitmap bitmap = photo.getBitmap();
                            ImageView image = new
ImageView(getApplicationContext());
                            image.setImageBitmap(bitmap);
                            imageFlipper.addView(image); //Adding image to
the ViewFlipper
                            imageFlipper.setFlipInterval(3000); //3s
intervals
                            imageFlipper.startFlipping();
                        }
                    });
                }
            }
        });
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        //Called when map fragment is ready

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {
            return;
            //Checking for permissions
        }
        googleMap.setMyLocationEnabled(true); //Setting user's current
location in blue dot

        LatLng latLng = new
LatLng(getIntent().getExtras().getDouble("placeLatitude"),
                getIntent().getExtras().getDouble("placeLongitude"));
        //Creating a location using its latitude and longitude

        //Adding place name to the place marker
        googleMap.addMarker(new MarkerOptions().position(latLng)

.title(getIntent().getExtras().getString("placeName"))).showInfoWindow();

        //Adding zoom to the maps
        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng,
14.0f));
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }
}
```

LocationActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import com.skyfishjy.library.RippleBackground;

import nshmadhani.com.wakenbake.Interfaces.IConnectivityReceiver;
import nshmadhani.com.wakenbake.R;

public class LocationActivity extends AppCompatActivity implements
IConnectivityReceiver {

    public RippleBackground rippleBackground;
    public ImageView locationImageView;
    public TextView gettingLocationTextView;

    public static final String TAG =
LocationActivity.class.getSimpleName();
    private FusedLocationProviderClient mFusedLocationProviderClient;

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

        Log.d(TAG, "onCreate: ");
        rippleBackground = findViewById(R.id.content);
        locationImageView = findViewById(R.id.locationIcon);
        gettingLocationTextView = findViewById(R.id.gettingLocation);

        mFusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
        if (isNetworkAvailable()) { //Start Ripple Effect
            rippleBackground.startRippleAnimation();
            makeUseOfNewLocation();
            Log.d(TAG, "onCreate:  in if ");
        } else {
            new AlertDialog.Builder(this)
                    .setTitle("No Internet Connection")
                    .setMessage("Please check your internet connection")
                    .setPositiveButton("Retry",
                            new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                    isNetworkAvailable();
                                }
                            })

                    .setNegativeButton("Cancel",
                            new
android.content.DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                    finish();
                                }
                            }).create();
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull
String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions,
grantResults);

        if (grantResults.length > 0 && grantResults[0]
                == PackageManager.PERMISSION_GRANTED) { //If permission is
granted
            makeUseOfNewLocation();
        } else {
            Log.d(TAG, "onRequestPermissionsResult: ");
        }
    }

    private void makeUseOfNewLocation() {

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION)
                    != PackageManager.PERMISSION_GRANTED) {// If
ACCESS_FINE_LOCATION is granted
                //makeUseOfNewLocation();
                ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.ACCESS_FINE_LOCATION}, 1);
                Log.d(TAG, "onRequestPermissionsResult: in inner if ");
            } else {
                Log.d(TAG, "onRequestPermissionsResult: in inner else");
            }
            return;
        }

        //Getting the user's current location
        mFusedLocationProviderClient.getLastLocation()
                .addOnSuccessListener(this, new
OnSuccessListener<Location>() {
                    @Override
                    public void onSuccess(Location location) {
                        //Got last known location. in Some rare situations
this can be null.
                        if (location != null) {
                            Log.d(TAG, "onSuccess: " +
location.toString());
                            Intent intent = new
Intent(LocationActivity.this, NavigationActivity.class);
                            intent.putExtra("latitude",
location.getLatitude());
                            intent.putExtra("longitude",
location.getLongitude());
                            //intent.putExtra("email",
getIntent().getStringExtra("email"));
                            startActivity(intent);
                            finish();
                        } else {
                            Log.d(TAG, "onSuccess: " + "in else");
                        }
                    }
                });

    }

    @Override
    public boolean isNetworkAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = null;
        if (connectivityManager != null) {
            networkInfo = connectivityManager.getActiveNetworkInfo();
        }
        boolean networkAvailable = false;

        if (networkInfo != null && networkInfo.isConnected())
            networkAvailable = true;

        return networkAvailable;
    }

    @Override
    public void onStart() {
        super.onStart();
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
        makeUseOfNewLocation();
    }
}
```

LoginActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.ogaclejapan.smarttablayout.SmartTabLayout;
import com.ogaclejapan.smarttablayout.utils.v4.FragmentPagerItemAdapter;
import com.ogaclejapan.smarttablayout.utils.v4.FragmentPagerItems;

import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Interfaces.IConnectivityReceiver;
import nshmadhani.com.wakenbake.Fragments.LoginFragment;
import nshmadhani.com.wakenbake.Fragments.SignUpFragment;

public class LoginActivity extends AppCompatActivity implements
IConnectivityReceiver {

    public static final String TAG = LoginActivity.class.getSimpleName();
    public FirebaseAuth mAuth;

    private ViewPager mViewPager;

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

        mAuth = FirebaseAuth.getInstance(); // Creating a instance of
Firebase object

        if (isNetworkAvailable()) { // Checking Internet Connectivity
            Toast.makeText(this, "Internet Available",
Toast.LENGTH_SHORT).show();

            mViewPager = findViewById(R.id.view_pager);
            FragmentPagerItemAdapter adapter = new
FragmentPagerItemAdapter(
                    getSupportFragmentManager(),
FragmentPagerItems.with(this)
                    .add("Login", LoginFragment.class)
                    .add("Signup", SignUpFragment.class)
                    .create()); // Adding the fragments to the viewpager
object
            mViewPager.setAdapter(adapter);
            SmartTabLayout viewPagerTab = findViewById(R.id.viewpagertab);
            viewPagerTab.setViewPager(mViewPager); //setting the viewpager
to the SmartTabLayout

        } else {
            new AlertDialog.Builder(this)
                    .setTitle("No Internet Connection")
                    .setMessage("Please check your internet connection")
                    .setPositiveButton("Retry",
                            new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                   isNetworkAvailable();
                                }
                            })

                    .setNegativeButton("Cancel",
                            new
android.content.DialogInterface.OnClickListener() {
```

```
                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                    finish();
                                }
                            }).create();
        }
    }

    public FirebaseAuth getmAuth() {
        return mAuth;
    }

    public void setmAuth(FirebaseAuth mAuth) {
        this.mAuth = mAuth;
    }


    @Override
    public void onStart() {
        super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
        FirebaseUser currentUser = mAuth.getCurrentUser();

        if (currentUser != null) {
            Intent intent = new Intent(LoginActivity.this,
LocationActivity.class);
            intent.putExtra("email", currentUser.getEmail());
            startActivity(intent);
            finish();
        }
    }


    @Override
    public boolean isNetworkAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = null;
        if (connectivityManager != null) {
            networkInfo = connectivityManager.getActiveNetworkInfo();
        }
        boolean networkAvailable = false;

        if (networkInfo != null && networkInfo.isConnected())
            networkAvailable = true;

        return networkAvailable;
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }
}
```

NavigationActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.content.Intent;
import android.content.pm.ApplicationInfo;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.view.ViewPager;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBar;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.PopupMenu;
import android.support.v7.widget.Toolbar;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.Gravity;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.places.GeoDataClient;
import com.google.firebase.auth.FirebaseAuth;
import com.mancj.materialsearchbar.MaterialSearchBar;
import com.ogaclejapan.smarttablayout.SmartTabLayout;
import com.ogaclejapan.smarttablayout.utils.v4.FragmentPagerItemAdapter;
import com.ogaclejapan.smarttablayout.utils.v4.FragmentPagerItems;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Fragments.FirebasePlacesFragment;
import nshmadhani.com.wakenbake.Fragments.GooglePlacesFragment;
import nshmadhani.com.wakenbake.Fragments.TiffinPlaceFragment;
import nshmadhani.com.wakenbake.Models.FirebasePlaces;
import nshmadhani.com.wakenbake.Models.GooglePlaces;
import nshmadhani.com.wakenbake.Models.MasterData;
import nshmadhani.com.wakenbake.Models.TiffinPlaces;

public class NavigationActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener,
GoogleApiClient.OnConnectionFailedListener ,
        MaterialSearchBar.OnSearchActionListener,
PopupMenu.OnMenuItemClickListener {
    public static GooglePlacesFragment fragG;
    public static FirebasePlacesFragment fragF;
    public static TiffinPlaceFragment fragT;
    public static final String TAG =
NavigationActivity.class.getSimpleName();
    protected GeoDataClient mGeoDataClient;
    public FirebaseAuth mAuth;
    private MaterialSearchBar searchBar;
    public DrawerLayout drawer;
    public ViewPager mNavigationViewPager;
    public TextView mNavHeaderEmail;
    public ImageView mNavHeaderImage;
    public static MasterData mMaster;
    public static boolean isSearched=false;
    SmartTabLayout viewPagerTab;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.navigation_drawer);
        mAuth = FirebaseAuth.getInstance();

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        drawer = findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        ActionBar actionBar = getSupportActionBar(); // support.v7
        if (actionBar != null) {
            actionBar.setTitle("");
        }
        drawer.addDrawerListener(toggle);

        toggle.syncState();
        NavigationView navigationView = findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        View header = navigationView.getHeaderView(0);
        mNavHeaderEmail = header.findViewById(R.id.mNavHeaderEmail);
        mNavHeaderEmail.setText(mAuth.getCurrentUser().getEmail());

        mNavHeaderImage = header.findViewById(R.id.mNavHeaderImage);

        relatedToSearch();

        // Construct a GeoDataClient.
        mGeoDataClient =
com.google.android.gms.location.places.Places.getGeoDataClient(this, null);

        mNavigationViewPager = findViewById(R.id.nav_view_pager);
        FragmentPagerItemAdapter adapter = new FragmentPagerItemAdapter(
                getSupportFragmentManager(), FragmentPagerItems.with(this)
                .add("Night", FirebasePlacesFragment.class)
                .add("Day", GooglePlacesFragment.class)
                .add("Tiffin", TiffinPlaceFragment.class)
                .create()); //Adding the fragments to the SmartTabLayout

        mNavigationViewPager.setAdapter(adapter);

        viewPagerTab = findViewById(R.id.navviewpagertab);
        viewPagerTab.setViewPager(mNavigationViewPager);
    }

    private void relatedToSearch() {
        //Initializing the search bar
        searchBar = findViewById(R.id.searchBar);
        searchBar.setHint("Search...");
        searchBar.setRoundedSearchBarEnabled(false);
        searchBar.setPlaceHolder("Search");
        searchBar.setOnSearchActionListener(this);
    }

    @Override
    protected void onStart() {
        super.onStart();
    }

    public FirebaseAuth getmAuth() {
        return mAuth;
    }

    public void setmAuth(FirebaseAuth mAuth) {
        this.mAuth = mAuth;
    }

    @Override
    public void onBackPressed() {
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();
        Intent intent;

        if (id == R.id.nav_home) {

            item.setChecked(false);
            intent = new Intent(getApplicationContext(),
NavigationActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);

        } else if (id == R.id.nav_bookmarks) {

            intent = new Intent(getApplicationContext(),
BookmarkActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            startActivity(intent);
            item.setChecked(false);

        } else if (id == R.id.nav_logout) {

            item.setChecked(false);
            if (mAuth != null) {
                mAuth.signOut();
                intent = new Intent(NavigationActivity.this,
LoginActivity.class);
                startActivity(intent);
                finish();

            } else {
                Toast.makeText(this, "Please Log In!",
Toast.LENGTH_SHORT).show();
            }
        } else if (id == R.id.nav_share) {
            item.setChecked(false);

            shareApplication();
        }

        drawer = findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }

    private void shareApplication() {
        ApplicationInfo app = getApplicationContext().getApplicationInfo();
        String filePath = app.sourceDir;
        Intent intent = new Intent(Intent.ACTION_SEND);

        // MIME of .apk is "application/vnd.android.package-archive".
        // but Bluetooth does not accept this. Let's use "*/*" instead.
        intent.setType("*/*");

        // Append file and send Intent
        intent.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(new
File(filePath)));
        startActivity(Intent.createChooser(intent, "Share app via"));
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
    }

    @Override
    public void onConnectionFailed(@NonNull ConnectionResult
connectionResult) {}

    @Override
    public void onSearchStateChanged(boolean enabled) {
        //Checking the state of the search bar
        String s = enabled ? "enabled" : "disabled";
        isSearched = false;
    }


    @Override
    public void onSearchConfirmed(CharSequence text) {
        searchBar.addTextChangeListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i,
int i1, int i2) {
```

```
                isSearched = true;
                filter(charSequence.toString(),
mNavigationViewPager.getCurrentItem());
                if (charSequence.toString().length() > 0) {

                    //Searching is done as per tabs
                    switch (mNavigationViewPager.getCurrentItem()) {
                        case 0 : fragF.setAdapter();
                            break;
                        case 1 : fragG.setAdapter();
                            break;
                        case 2 : fragT.setAdapter();
                            break;
                    }
                }
            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int
i1, int i2) {
                isSearched = true;
                filter(charSequence.toString(),
mNavigationViewPager.getCurrentItem());

                if (charSequence.toString().length() > 0) {
                    //Searching is done as per tabs
                    switch (mNavigationViewPager.getCurrentItem()) {
                        case 0 : fragF.setAdapter();
                            break;
                        case 1 : fragG.setAdapter();
                            break;
                        case 2 : fragT.setAdapter();
                            break;
                    }
                }
            }

            @Override
            public void afterTextChanged(Editable editable) {
                isSearched=true;


filter(editable.toString(),mNavigationViewPager.getCurrentItem());
                if (editable.length() > 0) {
                    //Searching is done as per tabs
                    switch (mNavigationViewPager.getCurrentItem()) {
                        case 0 : fragF.setAdapter();
                            break;
                        case 1 : fragG.setAdapter();
                            break;
                        case 2 : fragT.setAdapter();
                            break;
                    }
                }
            }
        });
    }

    @Override
    public void onButtonClicked(int buttonCode) {
        switch (buttonCode){
            case MaterialSearchBar.BUTTON_NAVIGATION:
                drawer.openDrawer(Gravity.LEFT);
                break;
        }
    }

    @Override
    public boolean onMenuItemClick(MenuItem item) {
        return false;
    }

    void filter(String text,int TYPE) {

        //Creating temporary ArrayLists
        List<FirebasePlaces> tempF=new ArrayList<>();
        List<GooglePlaces> tempG=new ArrayList<>();
        List<TiffinPlaces> tempT=new ArrayList<>();
        mMaster=new MasterData();

        if (text != null) {

            switch(TYPE) {
                case 0:
                    for (FirebasePlaces f :
FirebasePlacesFragment.mFirebasePlacesList) {
                        if
(f.getmVendorName().toLowerCase().contains(text.toLowerCase())) {
                            //If the name is found, then it will be added
to the temporary list
                            tempF.add(f);
                        }
                    }
                    mMaster.setNight(tempF);
                    break;
                case 1:
                    for (GooglePlaces g :
GooglePlacesFragment.mGooglePlacesList) {
                        if
(g.getName().toLowerCase().contains(text.toLowerCase())) {
                            //If the name is found, then it will be added
to the temporary list
                            tempG.add(g);
                        }
                    }
                    mMaster.setDay(tempG);
                    break;
                case 2:
                    for (TiffinPlaces t :
TiffinPlaceFragment.mTiffinPlacesList) {
                        if
(t.getmTiffinName().toLowerCase().contains(text.toLowerCase())) {
                            //If the name is found, then it will be added
to the temporary list
                            tempT.add(t);
                        }
                    }
                    mMaster.setTiffin(tempT);
                    break;
            }

        } else {
            Toast.makeText(this, "Please enter query!",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

OtpActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.goodiebag.pinview.Pinview;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseException;
import com.google.firebase.FirebaseTooManyRequestsException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthInvalidCredentialsException;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;

import nshmadhani.com.wakenbake.Interfaces.IConnectivityReceiver;
import nshmadhani.com.wakenbake.R;

import static java.util.concurrent.TimeUnit.SECONDS;

public class OtpActivity extends AppCompatActivity implements
IConnectivityReceiver {

    private static final String TAG = OtpActivity.class.getSimpleName();
    private Pinview pinView;
    private FirebaseAuth mAuth;
    private EditText phoneNumberEditText;
    private TextView otpTextView;
    private Button confirmButton;
    private PhoneAuthProvider.OnVerificationStateChangedCallbacks
mCallbacks;
    public ProgressDialog progressDialog;
    public String phoneNumber;
    public PhoneAuthProvider.ForceResendingToken mResendToken;
    public String mVerificationId;

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

       if (isNetworkAvailable()) {
           //Phone Number
           phoneNumberEditText = findViewById(R.id.otpPhoneNumber);
           progressDialog = new ProgressDialog(this);
           //OTP
           otpTextView = findViewById(R.id.otpTextView);

           mAuth = FirebaseAuth.getInstance();

           pinView = findViewById(R.id.otpPinView);
           pinView.setInputType(Pinview.InputType.NUMBER);

           //Initializing the Confirm Button
           confirmButton = findViewById(R.id.confirmButton);

           confirmButton.setOnClickListener(new View.OnClickListener() {
               @Override
               public void onClick(View v) {
                   progressDialog.dismiss();
                   progressDialog.setMessage("Sending OTP.....");
                   progressDialog.show();
                   phoneNumber = phoneNumberEditText.getText().toString();
                   PhoneAuthProvider.getInstance().verifyPhoneNumber(
                           "+91" + phoneNumber,            // Phone number
to verify
                           60,                      // Timeout duration
                           SECONDS,                 // Unit of timeout
                           OtpActivity.this, // Activity (for callback
binding)
                           mCallbacks);
               }
           });

           mCallbacks = new
PhoneAuthProvider.OnVerificationStateChangedCallbacks() {

               @Override
               public void onVerificationCompleted(PhoneAuthCredential
credential) {

                   progressDialog.dismiss();
                   Toast.makeText(OtpActivity.this, "OTP Verified.",
Toast.LENGTH_SHORT).show();
                   signInWithPhoneAuthCredential(credential);
               }

               @Override
               public void onVerificationFailed(FirebaseException e) {
                   progressDialog.dismiss();
                   Log.d(TAG, "onVerificationFailed: " +
e.getLocalizedMessage());
                   Toast.makeText(OtpActivity.this, "Error in
Verification", Toast.LENGTH_SHORT).show();

                   if (e instanceof
FirebaseAuthInvalidCredentialsException) {
                       Toast.makeText(OtpActivity.this, "Invalid Request",
Toast.LENGTH_SHORT).show();
                   } else if (e instanceof
FirebaseTooManyRequestsException) {
                       Toast.makeText(OtpActivity.this, "The SMS quota for
the project has been exceeded", Toast.LENGTH_SHORT).show();
                   }
               }

               @Override
               public void onCodeSent(String verificationId,
                                      PhoneAuthProvider.ForceResendingToken
token) {
                   progressDialog.dismiss();
                   Toast.makeText(OtpActivity.this, "OTP Sent",
Toast.LENGTH_SHORT).show();
                   phoneNumberEditText.setVisibility(View.INVISIBLE);
                   otpTextView.setVisibility(View.VISIBLE);
                   pinView.setVisibility(View.VISIBLE);
                   confirmButton.setText("Verify OTP");
                   mVerificationId = verificationId;
                   mResendToken = token;
               }
           };
       }
       else {
            new AlertDialog.Builder(this)
                   .setTitle("No Internet Connection")
                   .setMessage("Please check your internet connection")
                   .setPositiveButton("Retry",
                           new DialogInterface.OnClickListener() {
                               @Override
                               public void onClick(DialogInterface dialog,
int which) {
                                   isNetworkAvailable();
                               }
                           })

                   .setNegativeButton("Cancel",
                           new
android.content.DialogInterface.OnClickListener() {
                               @Override
                               public void onClick(DialogInterface dialog,
int which) {
                                    finish();
                               }
                           }).create();
       }
    }

    private void signInWithPhoneAuthCredential(PhoneAuthCredential
credential){
        progressDialog.dismiss();
        progressDialog.setMessage("Signing you in...");
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task)
{
                        if (task.isSuccessful()) {
                            progressDialog.dismiss();
                            // Sign in success, update UI with the signed-
in user's information
                            Toast.makeText(OtpActivity.this, "Signing you 
in!", Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(OtpActivity.this, 
LocationActivity.class);
                            startActivity(intent);
                            finish();

                        } else {
                            progressDialog.dismiss();
                            // Sign in failed, display a message and update 
the UI
                            Toast.makeText(OtpActivity.this, "Error in 
signing you!", Toast.LENGTH_SHORT).show();
                            if (task.getException() instanceof 
FirebaseAuthInvalidCredentialsException) {
                                // The verification code entered was 
invalid
                                Toast.makeText(OtpActivity.this, "Firebase 
Exception", Toast.LENGTH_SHORT).show();
                            }
                        }
                    }
                });
    }

    @Override
    public boolean isNetworkAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager) 
getSystemService(CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = null;
        if (connectivityManager != null) {
            networkInfo = connectivityManager.getActiveNetworkInfo();
        }
        boolean networkAvailable = false;

        if (networkInfo != null && networkInfo.isConnected())
            networkAvailable = true;

        return networkAvailable;
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }
}
```

SplashScreenActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.Bundle;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.view.Window;
import android.widget.ImageView;

import com.squareup.picasso.Picasso;

import java.util.Timer;
import java.util.TimerTask;

import nshmadhani.com.wakenbake.Interfaces.IConnectivityReceiver;
import nshmadhani.com.wakenbake.R;


public class SplashScreenActivity extends AppCompatActivity implements
IConnectivityReceiver{

    private ImageView imageView;
    long Delay = 4000; // duration of the Splash Screen

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

        // Remove the Title Bar
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_splash_screen);

        if (isNetworkAvailable()) { //Checking internet connectivity
            initLayout(); // initiate the layout
        }
        else { //Displays a dialog box telling no internet connectivity
            new AlertDialog.Builder(this)
                    .setTitle("No Internet Connection")
                    .setMessage("Please check your internet connection")
                    .setPositiveButton("Retry",
                            new DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                    if (isNetworkAvailable())
                                        initLayout();
                                }
                            })

                    .setNegativeButton("Cancel",
                            new
android.content.DialogInterface.OnClickListener() {
                                @Override
                                public void onClick(DialogInterface dialog,
int which) {
                                    finish();
                                }
                            }).create();
        }
    }
    private void initLayout() {
        //Initiating the layout
        imageView = findViewById(R.id.splashScreenImage);

        Picasso.with(this)
                .load(R.drawable.splash_screen)
                .into(imageView);

        displaySplash(); // Displays the Splash Screen
    }
    private void displaySplash () {
        // Create a Timer
        Timer RunSplash = new Timer();

        // Task to do when the timer ends
        TimerTask ShowSplash = new TimerTask() {
            @Override
            public void run() {

                // Start HomeScreenActivity.class
                Intent intent = new Intent(SplashScreenActivity.this,
                        LoginActivity.class);
                startActivity(intent);
                finish();
            }
        };
        // Start the timer
        RunSplash.schedule(ShowSplash, Delay);
    }

    @Override
    protected void onPause() {
        super.onPause();
    }

    @Override
    protected void onResume() {
        super.onResume();
    }

    @Override
    public boolean isNetworkAvailable() {
        ConnectivityManager connectivityManager = (ConnectivityManager)
getSystemService(CONNECTIVITY_SERVICE);
        NetworkInfo networkInfo = null;
        if (connectivityManager != null) {
            networkInfo = connectivityManager.getActiveNetworkInfo();
        }
        boolean networkAvailable = false;

        if (networkInfo != null && networkInfo.isConnected())
            networkAvailable = true;

        return networkAvailable;
    }
}
```

TiffinPlacesActivity.java

```
package nshmadhani.com.wakenbake.Activities;

import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.FloatingActionButton;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.squareup.picasso.Picasso;
import com.willy.ratingbar.BaseRatingBar;
import com.willy.ratingbar.ScaleRatingBar;

import java.util.ArrayList;
import java.util.List;

import nshmadhani.com.wakenbake.Adapters.ReviewFragmentListAdapter;
import nshmadhani.com.wakenbake.Holders.ReviewResponse;
import nshmadhani.com.wakenbake.Models.PlaceBookmark;
import nshmadhani.com.wakenbake.Models.Review;
import nshmadhani.com.wakenbake.Models.WakeNBake;
import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Interfaces.IRetrofitDataApi;
import nshmadhani.com.wakenbake.Models.APIClient;
import nshmadhani.com.wakenbake.Holders.RatingsResponse;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TiffinPlacesActivity extends AppCompatActivity {

    public static final String TAG =
TiffinPlacesActivity.class.getSimpleName();
    public ImageView tiffinImage;
    public TextView tiffinName;
    public TextView tiffinFoodItems;
    public ScaleRatingBar tiffinRatings;
    public FloatingActionButton tiffinCall;
    public FloatingActionButton tiffinBookmark;
    private float oldRatings;
    private IRetrofitDataApi apiInterface;
    private double newRatings;
    private List<Review> mReviewList;
    public RecyclerView mReviewsRecyclerView;
    public ReviewFragmentListAdapter mReviewsListAdapter;
    private TextView mReviewHeading;
    private Button mSubmitButton;
    private EditText mReviewBody;

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

        tiffinImage= findViewById(R.id.tiffinPlaceImageView);
        tiffinName = findViewById(R.id.tiffin_placeNameTextView);
        tiffinFoodItems = findViewById(R.id.tiffin_foodType);
        tiffinRatings = findViewById(R.id.tiffin_simpleRatingBar);
        tiffinBookmark = findViewById(R.id.tiffin_bookmarkButton);
        tiffinCall = findViewById(R.id.tiffin_callButton);

        Picasso.with(this)
                .load(getIntent().getStringExtra("tiffin_url"))
                .into(tiffinImage);

        tiffinName.setText(getIntent().getStringExtra("tiffin_name"));
//Setting the name
        tiffinFoodItems.setText(String.format("Food Items: %s",
getIntent().getStringExtra("tiffin_food"))); //Setting the food items
        tiffinCall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                call(getIntent().getStringExtra("tiffin_number")); //Call
the vendor
            }
        });

        apiInterface =
APIClient.getClient().create(IRetrofitDataApi.class);

        tiffinRatings.setRating(getIntent().getFloatExtra("tiffin_ratings",
0));
        //Setting the initial ratings of the vendor

        tiffinRatings.setOnRatingChangeListener(new
BaseRatingBar.OnRatingChangeListener() {
            @Override
            public void onRatingChange(BaseRatingBar baseRatingBar, float
v) {
                tiffinRatings.setClickable(false); //First set the ratings
to non-clickable
                oldRatings = tiffinRatings.getRating(); //Get the old
ratings
                String vendorName = tiffinName.getText().toString(); // Get
the vendors name
                double mNewRatings = changeRatings(vendorName,v +
oldRatings); // Getting the new ratings + old ratings
                tiffinRatings.setRating((float)(mNewRatings + oldRatings));
//Setting the new ratings
            }
        });

        tiffinBookmark.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String id = getIntent().getStringExtra("tiffin_id");
                String name = getIntent().getStringExtra("tiffin_name");
                String url = getIntent().getStringExtra("tiffin_url");
                PlaceBookmark placeBookmark = new PlaceBookmark(id, name,
url);

                String foundPlace =
WakeNBake.database.iDoa().checkInDatabase(id);

                if (name.equalsIgnoreCase(foundPlace)) {
                    Toast.makeText(TiffinPlacesActivity.this, "Already
added to bookmarks!", Toast.LENGTH_SHORT).show();
                }
                else {
                    WakeNBake.database.iDoa().addPlace(placeBookmark);
                    Toast.makeText(TiffinPlacesActivity.this, "Bookmark
added!", Toast.LENGTH_SHORT).show();
                }
            }
        });

        mReviewHeading = findViewById(R.id.reviewHeading);
        mReviewBody = findViewById(R.id.reviewEditText);
        mSubmitButton = findViewById(R.id.submitButton);

        mReviewList = new ArrayList<>();
        mReviewsRecyclerView  = findViewById(R.id.mReviewRecyclerView);
        mReviewsRecyclerView.setLayoutManager(new
LinearLayoutManager(this));
        mReviewsListAdapter = new
ReviewFragmentListAdapter(mReviewList,this);
        mReviewsRecyclerView.setAdapter(mReviewsListAdapter);

        getReviews(tiffinName.getText().toString());

        final String mUsername =
FirebaseAuth.getInstance().getCurrentUser().getEmail();

        mSubmitButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                TiffinPlacesActivity.this.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        saveReview(mUsername,
mReviewBody.getText().toString(), tiffinName.getText().toString());
                    }
                });
            }
        });
    }

    private void getReviews(String mVendorName) {

        Call<ReviewResponse> call = apiInterface.getReviews(mVendorName);
        call.enqueue(new Callback<ReviewResponse>() {
            @Override
            public void onResponse(Call<ReviewResponse> call,
Response<ReviewResponse> response) {
                List<Review> reviewResponseList =
response.body().getmReviewData();
```

```
                for (Review r : reviewResponseList) {
                    Review review = new Review();
                    review.setmUsernameReview(r.getmUsernameReview());
                    review.setmReview(r.getmReview());

                    mReviewList.add(r);
                }

                Log.d(TAG, "onResponse: list size: " + mReviewList.size());

                TiffinPlacesActivity.this.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        mReviewsListAdapter.notifyDataSetChanged();
                    }
                });
            }

            @Override
            public void onFailure(Call<ReviewResponse> call, Throwable t) {
                Log.d(TAG, "onFailure: " + t);
            }
        });
    }


    private void saveReview(String mUsername, String s, String mVendorName)
{
        Call<ResponseBody> saveCall = apiInterface.saveReviews(s,
mUsername, mVendorName);
        saveCall.enqueue(new Callback<ResponseBody>() {
            @Override
            public void onResponse(@NonNull Call<ResponseBody> call,
@NonNull Response<ResponseBody> response) {
                Log.d(TAG, "onResponse: " + response.body());
                if (response.isSuccessful()) {
                    Toast.makeText(TiffinPlacesActivity.this, "Review
added!", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(TiffinPlacesActivity.this, "Server
Error!", Toast.LENGTH_SHORT).show();
                }
            }

            @Override
            public void onFailure(@NonNull Call<ResponseBody> call,
@NonNull Throwable t) {
                Log.d(TAG, "onFailure: " + t);
            }
        });
    }

    private double changeRatings(String name, float v) {
        //Sending the ratings to the firebase real-time database
        Call<RatingsResponse> call = apiInterface.saveTiffinRatings(name,
v);  //Creating a request call to the firebase
        call.enqueue(new Callback<RatingsResponse>() {
            @Override
            public void onResponse(@NonNull Call<RatingsResponse> call,
@NonNull Response<RatingsResponse> response) {
                List<Double> r = response.body().getRatings(); //Getting
```

```
response in form of JSON
                newRatings = r.get(0); //Getting the first element
            }

            @Override
            public void onFailure(Call<RatingsResponse> call, Throwable t)
{
                Log.d(TAG, "onFailure: server: " +
t.getLocalizedMessage());
            }
        });

        return newRatings; //Returning the new ratings
    }


    private void call(String tiffin_number) {
        Intent intent = new Intent(Intent.ACTION_DIAL); //Redirecting to
the Dialer Application
        intent.setData(Uri.parse("tel: " + tiffin_number)); //Getting the
phone number

        if (Build.VERSION.SDK_INT >= 23 ) { //If higher than Android M
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                return;
                //Checking for the permissions
            }
            startActivity(intent); //Start Dialer application
        }
        else { //If lower than Android M
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                return;
                //Check for permissions
            }
            startActivity(intent); //Start Dialer application
        }
    }
}
```

FirebasePlacesListAdapter.java

```
package nshmadhani.com.wakenbake.Adapters;

import android.content.Context;
import android.content.Intent;
import android.support.v7.widget.CardView;
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.squareup.picasso.Picasso;

import java.util.List;
import java.util.Objects;
import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Activities.FirebasePlaceActivity;
import nshmadhani.com.wakenbake.Models.FirebasePlaces;


public class FirebasePlacesListAdapter extends RecyclerView.Adapter
        <FirebasePlacesListAdapter.MyViewHolder> {

    private List<FirebasePlaces> mFirebasePlacesList;
    private Context context;

    public FirebasePlacesListAdapter(List<FirebasePlaces>
mFirebasePlacesList, Context context) {
        this.mFirebasePlacesList = mFirebasePlacesList;
        this.context = context;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
{
        LayoutInflater li = LayoutInflater.from(parent.getContext());
        View view = li.inflate(R.layout.place_card,parent,false);

        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(MyViewHolder holder, int position) {

        final FirebasePlaces places = mFirebasePlacesList.get(position);
        holder.mVendorName.setText(places.getmVendorName());

        if (Objects.equals(places.getmVendorUrl(), "")) {
            Picasso.with(context)
                    .load(R.drawable.no_image)
                    .into(holder.mVendorImage);
        }
        else {
            Picasso.with(context)
                    .load(places.getmVendorUrl())
                    .into(holder.mVendorImage);
        }

        holder.mCardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context,
FirebasePlaceActivity.class);
                intent.putExtra("vendor_name", places.getmVendorName());
                intent.putExtra("vendor_ratings",
places.getmVendorRatings());
                intent.putExtra("vendor_phone",
places.getmVendorPhoneNumber());
                intent.putExtra("vendor_id", places.getmVendorId());
                intent.putExtra("vendor_lat", places.getmVendorLatitude());
                intent.putExtra("vendor_lng",
places.getmVendorLongitude());
                intent.putExtra("vendor_food",
places.getmVendorFoodItems());
                intent.putExtra("vendor_url", places.getmVendorUrl());
                intent.putExtra("vendor_open",
places.getmVendorOpenTime());
                intent.putExtra("vendor_close",
places.getmVendorCloseTime());
                context.startActivity(intent);
            }
        });
    }

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

    public class MyViewHolder extends RecyclerView.ViewHolder{

        private ImageView mVendorImage;
        private TextView mVendorName;
        private CardView mCardView;


        public MyViewHolder(View itemView) {
            super(itemView);

            mVendorImage = itemView.findViewById(R.id.placeImage);
            mVendorName = itemView.findViewById(R.id.placeName);
            mCardView = itemView.findViewById(R.id.card_view);
        }
    }
}
```

GooglePlacesListAdapter.java

```
package nshmadhani.com.wakenbake.Adapters;

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

import com.squareup.picasso.Picasso;

import java.util.List;
import java.util.Objects;

import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Activities.GooglePlacesActivity;
import nshmadhani.com.wakenbake.Models.GooglePlaces;

import static nshmadhani.com.wakenbake.Activities.LocationActivity.TAG;

/**
```

```
 * Created by Nachiket on 03-Feb-18.
 */

public class GooglePlacesListAdapter extends
RecyclerView.Adapter<GooglePlacesListAdapter.ViewHolder> {

    private List<GooglePlaces> googlePlacesList;
    private Context context;

    public GooglePlacesListAdapter(List<GooglePlaces> googlePlacesList,
Context context) {
        this.googlePlacesList = googlePlacesList;
        this.context = context;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater li = LayoutInflater.from(parent.getContext());
        View view = li.inflate(R.layout.place_card,parent,false);
        return new ViewHolder(view);
    }


    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position)
{

        final GooglePlaces googlePlaces = googlePlacesList.get(position);

        holder.mName.setText(googlePlaces.getName());

        if (Objects.equals(googlePlaces.getImageUrl(), "")) {
            Picasso.with(context)
                    .load(R.drawable.no_image)
                    .into(holder.mImage);
        }
        else {
            Picasso.with(context)
                    .load(googlePlaces.getImageUrl())
                    .into(holder.mImage);
        }

        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(context,
GooglePlacesActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                intent.putExtra("placeLatitude",
googlePlaces.getLatitude());
                intent.putExtra("placeLongitude",
googlePlaces.getLongitude());
                intent.putExtra("placeName", googlePlaces.getName());
                intent.putExtra("placeId", googlePlaces.getPlaceId());
                intent.putExtra("placeRatings", googlePlaces.getRatings());
                intent.putExtra("placeUrl", googlePlaces.getImageUrl());
                context.startActivity(intent);
            }
        });
    }
    @Override
    public int getItemCount() {
        return googlePlacesList.size();
    }

    public void updateList(List<GooglePlaces> list){
        googlePlacesList = list;
        notifyDataSetChanged();
    }

    public class ViewHolder extends RecyclerView.ViewHolder {

        private TextView mName;
        private ImageView mImage;
        private CardView cardView;

        private ViewHolder (View itemView) {
            super(itemView);
            Log.d(TAG, "ListViewHolder: "+itemView);
            mName = itemView.findViewById(R.id.placeName);
            mImage = itemView.findViewById(R.id.placeImage);
            cardView = itemView.findViewById(R.id.card_view);
        }
    }
}
```

PlaceBookmarkAdapter.java

```
package nshmadhani.com.wakenbake.Adapters;

import android.content.Context;
import android.support.v7.widget.CardView;
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 android.widget.Toast;

import com.squareup.picasso.Picasso;

import java.util.List;

import nshmadhani.com.wakenbake.Models.PlaceBookmark;
import nshmadhani.com.wakenbake.R;

/**
 * Created by Nachiket on 17-Mar-18.
 */

public class PlaceBookmarkAdapter extends
RecyclerView.Adapter<PlaceBookmarkAdapter.MyViewHolder> {

    private List<PlaceBookmark> list;
    private Context context;

    public PlaceBookmarkAdapter(List<PlaceBookmark> list, Context context)
{
        this.list = list;
        this.context = context;
    }

    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
{
        LayoutInflater li = LayoutInflater.from(parent.getContext());
        View view = li.inflate(R.layout.place_card,parent,false);
        return new MyViewHolder(view);
    }

    @Override
    public void onBindViewHolder(MyViewHolder holder, int position) {
        PlaceBookmark placeBookmark = list.get(position);

        holder.mName.setText(placeBookmark.getPlaceNAME());

        Picasso.with(context)
                .load(placeBookmark.getPlaceURL())
                .into(holder.mImage);

        //holder.context.getApplicationContext();

        holder.mCardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast.makeText(context, "Item clicked",
Toast.LENGTH_SHORT).show();
            }
        });
    }

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

    public class MyViewHolder extends RecyclerView.ViewHolder {

        private TextView mName;
        private ImageView mImage;
        private CardView mCardView;
        private Context context;

        public MyViewHolder(View itemView) {
            super(itemView);

            mName = itemView.findViewById(R.id.placeName);
            mImage = itemView.findViewById(R.id.placeImage);
            mCardView = itemView.findViewById(R.id.card_view);
        }
    }

}
```

ReviewFragmentListAdapter.java

```
package nshmadhani.com.wakenbake.Adapters;

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

import java.util.List;

import nshmadhani.com.wakenbake.Models.Review;
import nshmadhani.com.wakenbake.R;

/**
 * Created by Nachiket on 27-Mar-18.
 */

public class ReviewFragmentListAdapter extends
RecyclerView.Adapter<ReviewFragmentListAdapter.ListReviewHolder> {

    private List<Review> mReviewList;
    private Context context;

    public ReviewFragmentListAdapter(List<Review> mReviewList, Context
context) {
        this.mReviewList = mReviewList;
        this.context = context;
    }

    @Override
    public ListReviewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {

        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.review_layout,parent, false);
        return new ListReviewHolder(view);
    }

    @Override
    public void onBindViewHolder(ListReviewHolder holder, int position) {

        Review review = mReviewList.get(position);

        holder.mUsername.setText("Username: " +
review.getmUsernameReview());

        holder.mReview.setText("Review: " + review.getmReview());

    }

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

    public class ListReviewHolder extends RecyclerView.ViewHolder {

        private TextView mUsername;
        private TextView mReview;

        public ListReviewHolder(View itemView) {
            super(itemView);

            mUsername = itemView.findViewById(R.id.mRecyclerViewEmail);
            mReview = itemView.findViewById(R.id.mRecyclerViewReview);

        }
    }
}
```

TiffinPlacesListAdapter.java

```
package nshmadhani.com.wakenbake.Adapters;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.widget.CardView;
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.squareup.picasso.Picasso;

import java.util.List;

import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Activities.TiffinPlacesActivity;
import nshmadhani.com.wakenbake.Models.TiffinPlaces;

public class TiffinPlacesListAdapter extends RecyclerView.Adapter
        <TiffinPlacesListAdapter.ListViewHolder>{

    private List<TiffinPlaces> mTiffinPlacesList;
    private Context context;

    public TiffinPlacesListAdapter(List<TiffinPlaces> tiffinPlacesList,
Context context) {
        this.mTiffinPlacesList = tiffinPlacesList;
        this.context = context;
    }

    @Override
    public ListViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View view = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.list_places, parent, false);
        return new ListViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ListViewHolder holder, int position) {

        final TiffinPlaces mTiffinPlace = mTiffinPlacesList.get(position);
        holder.mTiffinName.setText(mTiffinPlace.getmTiffinName());
        Picasso.with(context)
                .load(R.drawable.no_image)
                .into(holder.mTiffinImage);

        final Uri imageURL =
Uri.parse(String.valueOf(R.drawable.user_image));

        holder.mCardView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context,
TiffinPlacesActivity.class);
                intent.putExtra("tiffin_name",
mTiffinPlace.getmTiffinName());
                intent.putExtra("tiffin_food",
mTiffinPlace.getmTiffinFoodItems());
                intent.putExtra("tiffin_ratings",
mTiffinPlace.getmTiffinRatings());
                intent.putExtra("tiffin_number",
mTiffinPlace.getmTiffinNumber());
                intent.putExtra("tiffin_id", mTiffinPlace.getmTiffinId());
                intent.putExtra("tiffin_url", imageURL);
                context.startActivity(intent);
            }
        });
    }

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

    public class ListViewHolder extends RecyclerView.ViewHolder {

        private ImageView mTiffinImage;
        private TextView mTiffinName;
        //private TextView mTimeStamp;
        private CardView mCardView;

        public ListViewHolder(View itemView) {
            super(itemView);

            mTiffinImage = itemView.findViewById(R.id.listPlaceImage);
            mTiffinName = itemView.findViewById(R.id.listPlaceName);
            //mTimeStamp = itemView.findViewById(R.id.time_stamp);
            mCardView = itemView.findViewById(R.id.list_card_view);
        }
    }
}
```

FirebasePlacesFragment.java

```
package nshmadhani.com.wakenbake.Fragments;
import android.app.ProgressDialog;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;

import java.util.ArrayList;
import java.util.List;

import nshmadhani.com.wakenbake.Holders.FirebasePlacesHolder;
import nshmadhani.com.wakenbake.Adapters.FirebasePlacesListAdapter;
import nshmadhani.com.wakenbake.Activities.NavigationActivity;
import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Interfaces.IRetrofitDataApi;
import nshmadhani.com.wakenbake.Models.APIClient;
import nshmadhani.com.wakenbake.Models.FirebasePlaces;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class FirebasePlacesFragment extends android.support.v4.app.Fragment
{

    public static List<FirebasePlaces> mFirebasePlacesList;
    private NavigationActivity mActivity;
    private RecyclerView mFirebasePlacesRecyclerView;
    private FirebaseAuth mAuth;
    private FirebasePlacesListAdapter mFirebasePlacesListAdapter;
    public static final String TAG =
FirebasePlacesFragment.class.getSimpleName();
    public IRetrofitDataApi iRetrofitDataApi;
    public ProgressDialog progressDialog;

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater
                .inflate(R.layout.firebase_places_fragment, container,
false);

        mFirebasePlacesList = new ArrayList<>();
        mFirebasePlacesRecyclerView  =
rootView.findViewById(R.id.firebasePlacesRecyclerView);
        mFirebasePlacesRecyclerView.setLayoutManager(new
LinearLayoutManager(mActivity));
        mFirebasePlacesListAdapter = new
FirebasePlacesListAdapter(mFirebasePlacesList, mActivity);
        mFirebasePlacesRecyclerView.setAdapter(mFirebasePlacesListAdapter);
        progressDialog = new ProgressDialog(mActivity);

        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                progressDialog.setMessage("Loading...");
                progressDialog.show();
                getPlacesFromFirebaseDatabase();
            }
        });

        return rootView;
    }

    @Override
    public void onViewStateRestored(@Nullable Bundle savedInstanceState) {
        super.onViewStateRestored(savedInstanceState);
        String firebase = (savedInstanceState != null) ?
savedInstanceState.getString("firebase") : "null";
        Log.i(TAG, " onViewStateRestored: " + firebase);
    }

    @Override
    public void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);

        Log.i(TAG, " onSaveInstanceState.");
        outState.putString("firebase", "FirebasePlaces");
    }

    @Override
    public void onPause() {
        super.onPause();
    }

    @Override
    public void onResume() {
        super.onResume();
    }

    public void setAdapter(){
        if(NavigationActivity.isSearched){
            List<FirebasePlaces> temp=
NavigationActivity.mMaster.getNight();
            mFirebasePlacesListAdapter = new
FirebasePlacesListAdapter(temp, mActivity);

mFirebasePlacesRecyclerView.setAdapter(mFirebasePlacesListAdapter);
        }
    }


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

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        NavigationActivity.fragF=this;
        mActivity = (NavigationActivity) getActivity();
        mAuth = mActivity.getmAuth();

        iRetrofitDataApi =
APIClient.getClient().create(IRetrofitDataApi.class);
    }

    private void getPlacesFromFirebaseDatabase() {

        Call<FirebasePlacesHolder> call =
iRetrofitDataApi.getPlacesFromFirebase("");
        Log.d(TAG, "onCreate: Connection successful");
        call.enqueue(new Callback<FirebasePlacesHolder>() {
            @Override
            public void onResponse(@NonNull Call<FirebasePlacesHolder>
call, @NonNull Response<FirebasePlacesHolder> response) {
                progressDialog.dismiss();
                List<FirebasePlaces> firebasePlacesList =
response.body().getmPlaces();
                if (firebasePlacesList != null) {
                    for(FirebasePlaces f : firebasePlacesList) {
                        final FirebasePlaces mFirebasePlaces = new
FirebasePlaces();
                        mFirebasePlaces.setmVendorName(f.mVendorName);

mFirebasePlaces.setmVendorPhoneNumber(f.mVendorPhoneNumber);

mFirebasePlaces.setmVendorLatitude(f.mVendorLatitude);

mFirebasePlaces.setmVendorLongitude(f.mVendorLongitude);

mFirebasePlaces.setmVendorRatings(f.mVendorRatings);
                        mFirebasePlaces.setmVendorId(f.mVendorId);

mFirebasePlaces.setmVendorOpenTime(f.mVendorOpenTime);

mFirebasePlaces.setmVendorCloseTime(f.mVendorCloseTime);

mFirebasePlaces.setmVendorFoodItems(f.mVendorFoodItems);

                        if (f.mVendorUrl == null) {
                            mFirebasePlaces.setmVendorUrl("");
                        } else {
                            mFirebasePlaces.setmVendorUrl(f.mVendorUrl);
                        }
                        Log.d(TAG, "onResponse: " + f.getmVendorId());
                        Log.d(TAG, "onResult: "+f.getmVendorName());
                        Log.d(TAG, "onResponse: url : " + f.mVendorUrl);
                        mFirebasePlacesList.add(f);
                    }
                }
                Log.d(TAG, "onResult: "+ mFirebasePlacesList.size());
                Log.d(TAG, "onResponse: "+response.body());
                mActivity.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        mFirebasePlacesListAdapter.notifyDataSetChanged();
                    }
                });
            }

            @Override
            public void onFailure(@NonNull Call<FirebasePlacesHolder> call,
@NonNull Throwable t) {
                progressDialog.dismiss();
                Log.e(TAG, "onFailure: ",t);
                Toast.makeText(mActivity, "Server Error!",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

GooglePlacesFragment.java

```
package nshmadhani.com.wakenbake.Fragments;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.maps.GeoApiContext;
import com.google.maps.NearbySearchRequest;
import com.google.maps.PendingResult;
import com.google.maps.model.LatLng;
import com.google.maps.model.PlaceType;
import com.google.maps.model.PlacesSearchResponse;
import com.google.maps.model.PlacesSearchResult;
import com.google.maps.model.RankBy;

import java.util.ArrayList;
import java.util.List;

import nshmadhani.com.wakenbake.Adapters.GooglePlacesListAdapter;
import nshmadhani.com.wakenbake.Activities.NavigationActivity;
import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Models.GooglePlaces;

public class GooglePlacesFragment extends android.support.v4.app.Fragment {

    public static  List<GooglePlaces> mGooglePlacesList;
    private RecyclerView mGooglePlacesRecyclerView;
    private GooglePlacesListAdapter mGooglePlacesListAdapter;
    public static String apiKey =
"AIzaSyBiREqfd8QWqyeTii3djqE0IhVmKCfoHjs";
    private NavigationActivity mActivity;
    private FirebaseAuth mAuth;
    private List<GooglePlaces> mMasterGooglePlaces;
    public static final String TAG =
GooglePlacesFragment.class.getSimpleName();
    public ProgressDialog progressDialog;

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(
                R.layout.google_places_fragment, container, false);

        mGooglePlacesList = new ArrayList<>();
        mGooglePlacesRecyclerView  =
rootView.findViewById(R.id.googlePlacesFragment);
        mGooglePlacesRecyclerView.setLayoutManager(new
LinearLayoutManager(mActivity));
        mGooglePlacesListAdapter = new
GooglePlacesListAdapter(mGooglePlacesList, mActivity);
        mGooglePlacesRecyclerView.setAdapter(mGooglePlacesListAdapter);

        progressDialog = new ProgressDialog(mActivity);

        final Intent intent = mActivity.getIntent();

        mMasterGooglePlaces = new ArrayList<>();

        final double latitude = intent.getDoubleExtra("latitude", 0.0);
        final double longitude = intent.getDoubleExtra("longitude", 0.0);

        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                getPlacesFromGoogle(latitude, longitude);
                progressDialog.setMessage("Loading...");
                progressDialog.show();
            }
        });
        return rootView;
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        NavigationActivity.fragG=this;
        mActivity = (NavigationActivity) getActivity();
        mAuth = mActivity.getmAuth();

    }


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

    @Override
    public void onActivityCreated(@Nullable Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
    }
    @Override
    public void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);

        Log.i(TAG, " onSaveInstanceState.");
        outState.putString("google", "GooglePlaces");
    }

    public void setAdapter(){
        if(NavigationActivity.isSearched){
            List<GooglePlaces> temp= NavigationActivity.mMaster.getDay();
            mGooglePlacesListAdapter = new GooglePlacesListAdapter(temp,
mActivity);
            mGooglePlacesRecyclerView.setAdapter(mGooglePlacesListAdapter);
        }
    }

    @Override
    public void onPause() {
        super.onPause();
    }

    @Override
    public void onResume() {
        super.onResume();
    }

    @Override
    public void onViewStateRestored(@Nullable Bundle savedInstanceState) {
        super.onViewStateRestored(savedInstanceState);

        String google = (savedInstanceState != null) ?
savedInstanceState.getString("google") : "null";
        Log.i(TAG, " onViewStateRestored: " + google);
    }

    private void getPlacesFromGoogle(final double latitude, final double
longitude) {
        GeoApiContext geoApiContext = new
GeoApiContext.Builder().apiKey(getString(R.string.google_api_key)).build();
        NearbySearchRequest nearbySearchRequest = new
NearbySearchRequest(geoApiContext);

        nearbySearchRequest
                .location(new LatLng(latitude, longitude))
                .type(PlaceType.RESTAURANT , PlaceType.CAFE ,PlaceType.FOOD
,PlaceType.BAKERY)
                .openNow(true)
                .rankby(RankBy.DISTANCE)
                .setCallback(new
PendingResult.Callback<PlacesSearchResponse>() {
                @Override
                public void onResult(PlacesSearchResponse result) {
                    progressDialog.dismiss();
                    for (PlacesSearchResult place : result.results) {
                        GooglePlaces googlePlaces = new GooglePlaces();
                        googlePlaces.setName(place.name);

googlePlaces.setLatitude(place.geometry.location.lat);

googlePlaces.setLongitude(place.geometry.location.lng);
```

```
                        googlePlaces.setPlaceId(place.placeId);
                        googlePlaces.setRatings(place.rating);
                        Log.d(TAG, "onResult: " + place.placeId);
                        String URL = "";

                        try {
                            URL =
"https://maps.googleapis.com/maps/api/place/photo?maxwidth=400&photoreferen
ce="
                                    + place.photos[0].photoReference +
"&key=" + apiKey;
                        } catch (Exception e) {
                            URL = "";
                        }
                        googlePlaces.setImageUrl(URL);
                        Log.d(TAG, "onResult: "+ googlePlaces.getName() +
googlePlaces.getRatings());
mGooglePlacesList.add(googlePlaces);
                                            }
                    Log.d(TAG, "onResult: "+ mGooglePlacesList.size());
                    mActivity.runOnUiThread(new Runnable() {
                        @Override
                        public void run() {

mGooglePlacesListAdapter.notifyDataSetChanged();
                        }
                    });

                }
                @Override
                public void onFailure(Throwable e) {
                    progressDialog.dismiss();
                    Log.e(TAG , "onFailure: ", e);
                    Toast.makeText(mActivity, "Server Error!",
Toast.LENGTH_SHORT).show();
                }
            });
    }
}
```

LoginFragment.java

```
package nshmadhani.com.wakenbake.Fragments;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.design.widget.TextInputLayout;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;
import com.google.android.gms.auth.api.Auth;
import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.SignInButton;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.GoogleAuthProvider;

import nshmadhani.com.wakenbake.Activities.LocationActivity;
import nshmadhani.com.wakenbake.Activities.LoginActivity;
import nshmadhani.com.wakenbake.R;

/**
 * Created by Nisha on 3/15/2018.
 */

public class LoginFragment extends Fragment {


    private static final int RC_SIGN_IN = 2;
    private TextInputLayout mUserNameTextInputLayout;
    private TextInputLayout mPasswordTextInputLayout;
    private Button mLoginButton;
    public GoogleSignInClient mGoogleSignInClient;
    private FirebaseAuth mAuth;
    private LoginActivity mActivity;
    public static final String TAG = LoginFragment.class.getSimpleName();
    public GoogleSignInAccount acct;
    public GoogleApiClient mGoogleApiClient;
    public SignInButton mGoogleButton;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, @Nullable Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(
                R.layout.login_fragment, container, false);
        //Initializing the layout of the fragment

        mUserNameTextInputLayout =
rootView.findViewById(R.id.username_til);
        mPasswordTextInputLayout =
rootView.findViewById(R.id.password_til);
        mLoginButton = rootView.findViewById(R.id.login_bt);

        // Configure Google Sign In
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();

        mGoogleApiClient = new GoogleApiClient.Builder(mActivity)
```

```
                .enableAutoManage(mActivity, new
GoogleApiClient.OnConnectionFailedListener() {
                    @Override
                    public void onConnectionFailed(@NonNull
ConnectionResult connectionResult) {
                        Toast.makeText(mActivity,
"OnConnectFailedListener", Toast.LENGTH_SHORT).show();
                    }
                })
                .addApi(Auth.GOOGLE_SIGN_IN_API, gso)
                .build();

        mGoogleButton = rootView.findViewById(R.id.googleButton);

        mGoogleButton.setSize(SignInButton.SIZE_WIDE);
        mLoginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Log.d(TAG, "onClick: click");
                attemptLogin();
            }
        });


        mGoogleButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Log.d(TAG, "onClick: google button" );
                final Intent signInIntent =
Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
                mActivity.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        startActivityForResult(signInIntent, RC_SIGN_IN);
                    }
                });
            }
        });
        return rootView;
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mActivity = (LoginActivity) getActivity();

        if (mActivity != null) {
            mAuth = mActivity.getmAuth();
        }

    }

    public void toggleInputs(){
        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {

mUserNameTextInputLayout.getEditText().setEnabled(!mUserNameTextInputLayout
.getEditText().isEnabled());

mPasswordTextInputLayout.getEditText().setEnabled(!mPasswordTextInputLayout
.getEditText().isEnabled());
                    mLoginButton.setEnabled(!mLoginButton.isEnabled());
            }
        });
    }

    public void attemptLogin(){
        toggleInputs();
        String username =
mUserNameTextInputLayout.getEditText().getText().toString();
        String password =
mPasswordTextInputLayout.getEditText().getText().toString();

        if (!username.equals("") && !password.equals("")) {
            try {
                Toast.makeText(mActivity, "All is Cool",
Toast.LENGTH_SHORT).show();
                signIn(username, password);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        //Checking if username is empty and password is not empty
        else if (username.equals("") && !password.equals("")) {
            Toast.makeText(getActivity().getApplicationContext(), "Enter
correct username", Toast.LENGTH_LONG).show();
        }
        //Checking if the username is not empty and password is empty
        else if (!username.equals("") && password.equals("")) {
            Toast.makeText(getActivity().getApplicationContext(), "Enter
correct password", Toast.LENGTH_LONG).show();
        }
        //Checking if the both the fields are filled
        else {
            Toast.makeText(getActivity().getApplicationContext(), "Enter
both username and password", Toast.LENGTH_LONG).show();
        }
        toggleInputs();
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent
data) {
        super.onActivityResult(requestCode, resultCode, data);

        // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
        if (requestCode == RC_SIGN_IN && resultCode == Activity.RESULT_OK)
{
            Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
            try {
                // Google Sign In was successful, authenticate with
Firebase
                GoogleSignInAccount account =
task.getResult(ApiException.class);
                firebaseAuthWithGoogle(account);
            } catch (ApiException e) {
                // Google Sign In failed, update UI appropriately
                Toast.makeText(mActivity, "Google sign in failed",
Toast.LENGTH_SHORT).show();
```

```
                // ...
            }
        }
    }

    private void firebaseAuthWithGoogle(GoogleSignInAccount account) {

        if (account != null) {

            Log.d(TAG, "firebaseAuthWithGoogle:" + account.getId());

            AuthCredential credential =
GoogleAuthProvider.getCredential(account.getIdToken(), null);
            mAuth.signInWithCredential(credential)
                    .addOnCompleteListener(mActivity, new
OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult>
task) {
                            if (task.isSuccessful()) {
                                // Sign in success, update UI with the
signed-in user's information
                                Log.d(TAG, "signInWithCredential:success");
                                Intent intent = new Intent(mActivity,
LocationActivity.class);
                                startActivity(intent);
                                mActivity.finish();
                            } else {
                                // If sign in fails, display a message to
the user.
                                Log.w(TAG, "signInWithCredential:failure",
task.getException());
                                Toast.makeText(mActivity, "Authentication
Failed.", Toast.LENGTH_SHORT).show();
                            }
                        }
                    });
        } else {
            Toast.makeText(mActivity, "No Google Sign IN",
Toast.LENGTH_SHORT).show();
        }
    }

    public void signIn (final String email, String password) throws
Exception {
        if(mActivity.isNetworkAvailable()) {
            Toast.makeText(mActivity, "Signing In",
Toast.LENGTH_SHORT).show();
            final ProgressDialog progressDialog = new
ProgressDialog(mActivity);
            progressDialog.setMessage("Verifying your credentials..");
            progressDialog.show();

            mAuth.signInWithEmailAndPassword(email, password)
                    .addOnCompleteListener(mActivity, new
OnCompleteListener<AuthResult>() {
                        @Override
                        public void onComplete(@NonNull Task<AuthResult>
task) {
                            if (task.isSuccessful()) {
                                progressDialog.dismiss();
                                // Sign in success, update UI with the
signed-in user's information
                                Log.i(TAG, "Successfully logged in!");
                                Intent intent = new Intent(mActivity,
LocationActivity.class);
                                intent.putExtra("email", email);
                                startActivity(intent);
                                mActivity.finish();

                            } else {
                                progressDialog.dismiss();
                                // If sign in fails, display a message to
the user.
                                Log.i(TAG, "signInWithEmail:failure",
task.getException());
                                Toast.makeText(mActivity, "Please enter
correct username and password",
                                        Toast.LENGTH_LONG).show();
                            }
                        }
                    });
        }
        else {
            Toast.makeText(mActivity, "NO Internet",
Toast.LENGTH_SHORT).show();
        }

    }
}
```

SignUpFragment.java

```
package nshmadhani.com.wakenbake.Fragments;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.design.widget.TextInputLayout;
import android.support.v4.app.Fragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;

import nshmadhani.com.wakenbake.Activities.LoginActivity;
import nshmadhani.com.wakenbake.Activities.OtpActivity;
import nshmadhani.com.wakenbake.R;


public class SignUpFragment extends Fragment {
    private TextInputLayout mUserNameTextInputLayout;
    private TextInputLayout mPasswordTextInputLayout;
    private TextInputLayout mEmailTextInputLayout;
    private Button mSignUpButton;
    private FirebaseAuth mAuth;
    private LoginActivity mActivity;
    public static final String TAG = SignUpFragment.class.getSimpleName();

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup 
container, @Nullable Bundle savedInstanceState) {
        ViewGroup rootView = (ViewGroup) inflater.inflate(
                R.layout.signup_fragment, container, false);

        mPasswordTextInputLayout = 
rootView.findViewById(R.id.signupPassword_til);
        mEmailTextInputLayout = 
rootView.findViewById(R.id.signupEmail_til);
        mSignUpButton = rootView.findViewById(R.id.signupButton);
        mSignUpButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                attemptSignup();
            }
        });
        return rootView;
    }

    public void toggleInputs(){
        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {

mPasswordTextInputLayout.getEditText().setEnabled(!mPasswordTextInputLayout
.getEditText().isEnabled());

mEmailTextInputLayout.getEditText().setEnabled(!mEmailTextInputLayout.getEd
itText().isEnabled());
                mSignUpButton.setEnabled(!mSignUpButton.isEnabled());
            }
        });
    }
    private void attemptSignup() {
        toggleInputs();

        String password = 
mPasswordTextInputLayout.getEditText().getText().toString();
        String email = 
mEmailTextInputLayout.getEditText().getText().toString();
        if ( !password.equals("")) {
            try {
                Toast.makeText(mActivity, "All is Cool", 
Toast.LENGTH_SHORT).show();
                signUp(email, password);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        //Checking if username is empty and password is not empty
        else if (!password.equals("")) {
            Toast.makeText(getActivity().getApplicationContext(), "Enter
correct username", Toast.LENGTH_LONG).show();
        }
        //Checking if the username is not empty and password is empty
        else if ( password.equals("")) {
            Toast.makeText(getActivity().getApplicationContext(), "Enter
correct password", Toast.LENGTH_LONG).show();
        }
        //Checking if the both the fields are filled
        else {
            Toast.makeText(getActivity().getApplicationContext(), "Enter
both username and password", Toast.LENGTH_LONG).show();
        }
        toggleInputs();
    }
    public void signUp(String email,String password) {
        final ProgressDialog progressDialog = new
ProgressDialog(mActivity);
        progressDialog.setMessage("Creating Account..\nThis might take a
while...");
        progressDialog.show();

        mAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(mActivity, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task)
{

                        if (task.isSuccessful()) {
                            progressDialog.dismiss();
                            // Sign in success, update UI with the signed-
in user's information
                            Intent intent = new Intent(mActivity,
OtpActivity.class);
                            startActivity(intent);
                            mActivity.finish();

                        } else {
                            progressDialog.dismiss();
                            // If sign in fails, display a message to the
user.
                            Log.i(TAG, "createUserWithEmail:failure",
task.getException());
                            Toast.makeText(mActivity, "Authentication
failed.",
                                    Toast.LENGTH_SHORT).show();
                        }
                    }
                });
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mActivity = (LoginActivity) getActivity();
        mAuth = mActivity.getmAuth();
    }
}
```

TiffinPlaceFragment.java

```java
package nshmadhani.com.wakenbake.Fragments;

import android.app.ProgressDialog;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;

import java.util.ArrayList;
import java.util.List;

import nshmadhani.com.wakenbake.Activities.NavigationActivity;
import nshmadhani.com.wakenbake.R;
import nshmadhani.com.wakenbake.Holders.TiffinPlacesHolder;
import nshmadhani.com.wakenbake.Adapters.TiffinPlacesListAdapter;
import nshmadhani.com.wakenbake.Interfaces.IRetrofitDataApi;
import nshmadhani.com.wakenbake.Models.APIClient;
import nshmadhani.com.wakenbake.Models.TiffinPlaces;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class TiffinPlaceFragment extends android.support.v4.app.Fragment{

    public static final String TAG =
TiffinPlaceFragment.class.getSimpleName();
    private FirebaseAuth mAuth;
    private NavigationActivity mActivity;
    public IRetrofitDataApi apiInterface;
    public static List<TiffinPlaces> mTiffinPlacesList;
    private TiffinPlacesListAdapter mTiffinPlacesListAdapter;
    private RecyclerView mTiffinPlacesRecyclerView;
    public ProgressDialog progressDialog;

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable
ViewGroup container, Bundle savedInstanceState) {

        ViewGroup rootView = (ViewGroup) inflater
                .inflate(R.layout.tiffin_places_fragment, container,
false);

        mTiffinPlacesList = new ArrayList<>();
        mTiffinPlacesRecyclerView  =
rootView.findViewById(R.id.tiffinplacesRecyclerView);
        mTiffinPlacesRecyclerView.setLayoutManager(new
LinearLayoutManager(mActivity));
        mTiffinPlacesListAdapter = new
TiffinPlacesListAdapter(mTiffinPlacesList, mActivity);
        mTiffinPlacesRecyclerView.setAdapter(mTiffinPlacesListAdapter);
        progressDialog = new ProgressDialog(mActivity);

        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                progressDialog.setMessage("Loading...");
                progressDialog.show();
                getTiffinPlacesFromFirebase();
            }
        });
        return rootView;
    }



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

    public void setAdapter() {
        if(NavigationActivity.isSearched){
            List<TiffinPlaces> temp=
NavigationActivity.mMaster.getTiffin();
            mTiffinPlacesListAdapter = new TiffinPlacesListAdapter(temp,
mActivity);
            mTiffinPlacesRecyclerView.setAdapter(mTiffinPlacesListAdapter);
        }
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        NavigationActivity.fragT = this;
        mActivity = (NavigationActivity) getActivity();
        mAuth = mActivity.getmAuth();

        apiInterface =
APIClient.getClient().create(IRetrofitDataApi.class);
    }

    @Override
    public void onViewStateRestored(@Nullable Bundle savedInstanceState) {
        super.onViewStateRestored(savedInstanceState);

        String tiffin = (savedInstanceState != null) ?
savedInstanceState.getString("tiffin") : "null";
        Log.i(TAG, " onViewStateRestored: " + tiffin);
    }

    @Override
    public void onResume() {
        super.onResume();
    }

    @Override
    public void onPause() {
        super.onPause();
    }

    @Override
    public void onSaveInstanceState(@NonNull Bundle outState) {
        super.onSaveInstanceState(outState);

        Log.i(TAG, " onSaveInstanceState.");
        outState.putString("tiffin", "TiffinPlaces");
    }

    private void getTiffinPlacesFromFirebase() {
        Call<TiffinPlacesHolder> call = apiInterface.getmTiffinPlaces("");
        Log.d(TAG, "onCreate: Connection successful");
        call.enqueue(new Callback<TiffinPlacesHolder>() {
            @Override
            public void onResponse(@NonNull Call<TiffinPlacesHolder> call,
@NonNull Response<TiffinPlacesHolder> response) {
                progressDialog.dismiss();
                List<TiffinPlaces> mTiffinPlaces =
response.body().getmTiffinPlaces();
                if (mTiffinPlaces != null) {
                    for (TiffinPlaces t : mTiffinPlaces) {
                        TiffinPlaces tiffinPlaces = new TiffinPlaces();
                        tiffinPlaces.setmTiffinName(t.mTiffinName);
                        tiffinPlaces.setmTiffinNumber(t.mTiffinNumber);

tiffinPlaces.setmTiffinFoodItems(t.mTiffinFoodItems);
                        tiffinPlaces.setmTiffinRatings(t.mTiffinRatings);
                        tiffinPlaces.setmTiffinId(t.mTiffinId);
                        Log.d(TAG, "onResponse: " +
tiffinPlaces.getmTiffinName());
                        mTiffinPlacesList.add(t);
                    }
                }
                Log.d(TAG, "onResult: tiffin"+ mTiffinPlacesList.size());
                Log.d(TAG, "onResponse: "+response.body());
                mActivity.runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        mTiffinPlacesListAdapter.notifyDataSetChanged();
                    }
                });
            }

            @Override
            public void onFailure(Call<TiffinPlacesHolder> call, Throwable
t) {
                progressDialog.dismiss();
                Log.e(TAG, "onFailure: ", t );
                Toast.makeText(mActivity, "Server Error!",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

FirebasePlacesHolder.java

```
package nshmadhani.com.wakenbake.Holders;

import com.google.gson.annotations.SerializedName;

import java.util.List;

import nshmadhani.com.wakenbake.Models.FirebasePlaces;


public class FirebasePlacesHolder {

    @SerializedName("data")
    private List<FirebasePlaces> mPlaces;

    public List<FirebasePlaces> getmPlaces() {
        return mPlaces;
    }

    public void setmPlaces(List<FirebasePlaces> mPlaces) {
        this.mPlaces = mPlaces;
    }
}
```

RatingsResponse.java

```
package nshmadhani.com.wakenbake.Holders;

import java.util.List;
import com.google.gson.annotations.Expose;
import com.google.gson.annotations.SerializedName;

public class RatingsResponse {

    @SerializedName("ratings")
    @Expose
    private List<Double> ratings = null;

    public List<Double> getRatings() {
        return ratings;
    }

    public void setRatings(List<Double> ratings) {
        this.ratings = ratings;
    }

}
```

ReviewResponse.java

```
package nshmadhani.com.wakenbake.Holders;

import com.google.gson.annotations.SerializedName;

import java.util.List;

import nshmadhani.com.wakenbake.Models.Review;

/**
 * Created by Nachiket on 27-Mar-18.
 */

public class ReviewResponse {

    @SerializedName("data")
    private List<Review> mReviewData;

    public List<Review> getmReviewData() {
        return mReviewData;
    }

    public void setmReviewData(List<Review> mReviewData) {
        this.mReviewData = mReviewData;
    }
}
```

TiffinPlacesHolder.java

```
package nshmadhani.com.wakenbake.Holders;

import com.google.gson.annotations.SerializedName;

import java.util.List;

import nshmadhani.com.wakenbake.Models.TiffinPlaces;

public class TiffinPlacesHolder {

    @SerializedName("data")
    private List<TiffinPlaces> mTiffinPlaces;

    public List<TiffinPlaces> getmTiffinPlaces() {
        return mTiffinPlaces;
    }

    public void setmTiffinPlaces(List<TiffinPlaces> mTiffinPlaces) {
        this.mTiffinPlaces = mTiffinPlaces;
    }
}
```

IConnectivityReceiver.java

```
package nshmadhani.com.wakenbake.Interfaces;

import static android.app.PendingIntent.getActivity;

public interface IConnectivityReceiver {
        boolean isNetworkAvailable ();
}
```

IDoa.java

```
package nshmadhani.com.wakenbake.Interfaces;

import android.arch.persistence.room.Dao;
import android.arch.persistence.room.Insert;
import android.arch.persistence.room.Query;

import java.util.List;

import nshmadhani.com.wakenbake.Models.PlaceBookmark;

@Dao
public interface IDoa {

    @Insert
    void addPlace(PlaceBookmark placeBookmark);

    @Query("SELECT * FROM place_bookmark")
    List<PlaceBookmark> fetchAllPlaces();

    @Query("SELECT name FROM place_bookmark WHERE id =:id")
    String checkInDatabase(String id);

}
```

IRetrofitDataApi.java

```
package nshmadhani.com.wakenbake.Interfaces;

import nshmadhani.com.wakenbake.Holders.FirebasePlacesHolder;
import nshmadhani.com.wakenbake.Holders.ReviewResponse;
import nshmadhani.com.wakenbake.Holders.TiffinPlacesHolder;
import nshmadhani.com.wakenbake.Holders.RatingsResponse;
import nshmadhani.com.wakenbake.Models.Review;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.http.Body;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.GET;
import retrofit2.http.POST;
import retrofit2.http.Query;

public interface IRetrofitDataApi {

    @GET("search.php")
    Call<FirebasePlacesHolder> getPlacesFromFirebase(@Query("search")
String name);

    @GET("searchtiffin.php")
    Call<TiffinPlacesHolder> getmTiffinPlaces(@Query("search") String
name);

    @POST("dynamicratings.php")
    @FormUrlEncoded
    Call<RatingsResponse> saveRatings(
            @Field("vendors") String vendorName,
            @Field("ratings") float ratings);

    @POST("tiffinratings.php")
    @FormUrlEncoded
    Call<RatingsResponse> saveTiffinRatings(
            @Field("vendor") String tiffinName,
            @Field("ratings") float ratings);

    @POST("reviewpost.php")
    @FormUrlEncoded
    Call<ResponseBody> saveReviews (
            @Field("review") String review,
            @Field("usrname") String username,
            @Field("vendorname") String vendorName);


    @POST("reviewretrieve.php")
    @FormUrlEncoded
    Call<ReviewResponse> getReviews(@Field("vendorname") String
mVendorName);

}
```

APIClient.java

```
package nshmadhani.com.wakenbake.Models;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

/**
 * Created by Nachiket on 22-03-2018.
 */

public class APIClient {

    public static final String BASE_URL =
"http://192.168.0.102:8085/Project/";
    private static Retrofit retrofit = null;

    public static Retrofit getClient() {

        Gson gson = new GsonBuilder()
                .setLenient()
                .create();
        if (retrofit == null) {
            retrofit = new Retrofit.Builder()
                    .baseUrl(BASE_URL)
                    .addConverterFactory(GsonConverterFactory.create(gson))
                    .build();
        }
        return retrofit;
    }
}
```

FirebasePlaces.java

```
package nshmadhani.com.wakenbake.Models;

import com.google.gson.annotations.SerializedName;

public class FirebasePlaces{

    @SerializedName("name")
    public String mVendorName;

    @SerializedName("id")
    public String mVendorId;

    @SerializedName("open")
    public int mVendorOpenTime;

    @SerializedName("close")
    public int mVendorCloseTime;

    @SerializedName("fooditems")
    public String mVendorFoodItems;

    @SerializedName("ratings")
    public float mVendorRatings;

    @SerializedName("number")
    public String mVendorPhoneNumber;

    @SerializedName("latitude")
    public double mVendorLatitude;

    @SerializedName("longitude")
    public double mVendorLongitude;

    @SerializedName("base_url")
    public String mVendorUrl;

    public String getmVendorId() {
        return mVendorId;
    }

    public void setmVendorId(String mVendorId) {
        this.mVendorId = mVendorId;
    }

    public String getmVendorUrl() {
        return mVendorUrl;
    }

    public void setmVendorUrl(String mVendorUrl) {
        this.mVendorUrl = mVendorUrl;
    }

    public String getmVendorName() {
        return mVendorName;
    }

    public void setmVendorName(String mVendorName) {
        this.mVendorName = mVendorName;
    }

    public int getmVendorOpenTime() {
        return mVendorOpenTime;
    }

    public void setmVendorOpenTime(int mVendorOpenTime) {
        this.mVendorOpenTime = mVendorOpenTime;
    }

    public int getmVendorCloseTime() {
        return mVendorCloseTime;
    }

    public void setmVendorCloseTime(int mVendorCloseTime) {
        this.mVendorCloseTime = mVendorCloseTime;
    }

    public String getmVendorFoodItems() {
        return mVendorFoodItems;
    }

    public void setmVendorFoodItems(String mVendorFoodItems) {
        this.mVendorFoodItems = mVendorFoodItems;
    }

    public float getmVendorRatings() {
        return mVendorRatings;
    }

    public void setmVendorRatings(float mVendorRatings) {
        this.mVendorRatings = mVendorRatings;
    }

    public String getmVendorPhoneNumber() {
        return mVendorPhoneNumber;
    }

    public void setmVendorPhoneNumber(String mVendorPhoneNumber) {
        this.mVendorPhoneNumber = mVendorPhoneNumber;
    }

    public double getmVendorLatitude() {
        return mVendorLatitude;
    }

    public void setmVendorLatitude(double mVendorLatitude) {
        this.mVendorLatitude = mVendorLatitude;
    }
    public double getmVendorLongitude() {
        return mVendorLongitude;
    }

    public void setmVendorLongitude(double mVendorLongitude) {
        this.mVendorLongitude = mVendorLongitude;
    }
}
```

GooglePlaces.java

```
package nshmadhani.com.wakenbake.Models;

public class GooglePlaces {

    public String name;
    public String number;
    public double ratings;
    public String photoReference;
    public String placeId;
    public String placeAddress;
    public String imageUrl;
    public double latitude;
    public double longitude;

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getNumber() {
        return number;
    }

    public void setNumber(String number) {
        this.number = number;
    }

    public double getRatings() {
        return ratings;
    }

    public void setRatings(double ratings) {
        this.ratings = ratings;
    }

    public String getPhotoReference() {
        return photoReference;
    }

    public void setPhotoReference(String photoReference) {
        this.photoReference = photoReference;
    }

    public String getPlaceId() {
        return placeId;
    }

    public void setPlaceId(String placeId) {
        this.placeId = placeId;
    }

    public String getPlaceAddress() {
        return placeAddress;
    }

    public void setPlaceAddress(String placeAddress) {
        this.placeAddress = placeAddress;
    }

    public String getImageUrl() {
        return imageUrl;
    }

    public void setImageUrl(String imageUrl) {
        this.imageUrl = imageUrl;
    }

    public double getLatitude() {
        return latitude;
    }

    public void setLatitude(double latitude) {
        this.latitude = latitude;
    }

    public double getLongitude() {
        return longitude;
    }

    public void setLongitude(double longitude) {
        this.longitude = longitude;
    }
}
```

MasterData.java

```
package nshmadhani.com.wakenbake.Models;

import java.util.List;

/**
 * Created by Nachiket on 24-Mar-18.
 */

public class MasterData {
    List<GooglePlaces> day;
    List<FirebasePlaces> night;
    List<TiffinPlaces> tiffin;

    public void setDay(List<GooglePlaces> day) {
        this.day = day;
    }
    public void setNight(List<FirebasePlaces> night) {
        this.night = night;
    }

    public void setTiffin(List<TiffinPlaces> tiffin) {
        this.tiffin = tiffin;
    }

    public List<GooglePlaces> getDay() {
        return day;
    }

    public List<FirebasePlaces> getNight() {
        return night;
    }

    public List<TiffinPlaces> getTiffin() {
        return tiffin;
    }
}
```

PlaceBookmark.java

```
package nshmadhani.com.wakenbake.Models;

import android.arch.persistence.room.ColumnInfo;
import android.arch.persistence.room.Entity;
import android.arch.persistence.room.PrimaryKey;
import android.support.annotation.NonNull;

@Entity(tableName = "place_bookmark")
public class PlaceBookmark {

    public PlaceBookmark(@NonNull String placeID, String placeNAME, String
placeURL) {
        this.placeID = placeID;
        this.placeNAME = placeNAME;
        this.placeURL = placeURL;
    }

    @PrimaryKey
    @ColumnInfo(name = "id")
    @NonNull
    private String placeID;

    @ColumnInfo(name = "name")
    private String placeNAME;

    @ColumnInfo(name = "image_url")
    private String placeURL;

    @NonNull
    public String getPlaceID() {
        return placeID;
    }

    public void setPlaceID(@NonNull String placeID) {
        this.placeID = placeID;
    }
    public String getPlaceNAME() {
        return placeNAME;
    }

    public void setPlaceNAME(String placeNAME) {
        this.placeNAME = placeNAME;
    }

    public String getPlaceURL() {
        return placeURL;
    }

    public void setPlaceURL(String placeURL) {
        this.placeURL = placeURL;
    }
}
```

Review.java

```
package nshmadhani.com.wakenbake.Models;

import com.google.gson.annotations.SerializedName;

/**
 * Created by Nachiket on 23-Mar-18.
 */

public class Review {

    @SerializedName("username")
    private String mUsernameReview;

    @SerializedName("review")
    private String mReview;


    public String getmUsernameReview() {
        return mUsernameReview;
    }

    public void setmUsernameReview(String mUsernameReview) {
        this.mUsernameReview = mUsernameReview;
    }

    public String getmReview() {
        return mReview;
    }

    public void setmReview(String mReview) {
        this.mReview = mReview;
    }
}
```

TiffinPlaces.java

```java
package nshmadhani.com.wakenbake.Models;

import com.google.gson.annotations.SerializedName;


public class TiffinPlaces  {

    @SerializedName("name")
    public String mTiffinName;

    @SerializedName("number")
    public String mTiffinNumber;

    @SerializedName("fooditems")
    public String mTiffinFoodItems;

    @SerializedName("ratings")
    public float mTiffinRatings;

    @SerializedName("id")
    public String mTiffinId;

    public String getmTiffinId() {
        return mTiffinId;
    }

    public void setmTiffinId(String mTiffinId) {
        this.mTiffinId = mTiffinId;
    }

    public String getmTiffinName() {
        return mTiffinName;
    }

    public void setmTiffinName(String mTiffinName) {
        this.mTiffinName = mTiffinName;
    }

    public String getmTiffinNumber() {
        return mTiffinNumber;
    }

    public void setmTiffinNumber(String mTiffinNumber) {
        this.mTiffinNumber = mTiffinNumber;
    }

    public String getmTiffinFoodItems() {
        return mTiffinFoodItems;
    }

    public void setmTiffinFoodItems(String mTiffinFoodItems) {
        this.mTiffinFoodItems = mTiffinFoodItems;
    }

    public float getmTiffinRatings() {
        return mTiffinRatings;
    }

    public void setmTiffinRatings(float mTiffinRatings) {
        this.mTiffinRatings = mTiffinRatings;
    }
}
```

WakeNBake.java

```
package nshmadhani.com.wakenbake.Models;

import android.app.Application;
import android.arch.persistence.room.Room;
import android.content.Context;
import android.support.multidex.MultiDex;

import nshmadhani.com.wakenbake.Abstracts.AppDatabase;

/**
 * Created by Nachiket on 17-Mar-18.
 */

public class WakeNBake extends Application {

    public static AppDatabase database;

    @Override
    protected void attachBaseContext(Context base) {
        super.attachBaseContext(base);
        MultiDex.install(this);
    }

    @Override
    public void onCreate() {
        super.onCreate();

        database = Room
                .databaseBuilder(getApplicationContext(),
AppDatabase.class, "bookmarks")
                .allowMainThreadQueries()
                .build();
    }
}
```

10.4 PHP Code

databasepost.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
//including all the preloaded functional files
$serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/serviceAccount.json');
//includes the firebase service account json file
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
$database = $firebase->getDatabase();
```

```
//gets the database of the current service account
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    //requesting input of food joints by post method
    $name = $_POST['name'];
    $number = $_POST['number'];
    $fooditem = $_POST['fooditems'];
    $rating = $_POST['ratings'];
    $latitude=$_POST['latitude'];
    $longitude=$_POST['longitude'];
    $open=$_POST['open'];
    $close=$_POST['close'];

$postData = array("");
//pushing every data of the vendors with vendors as root element and then
unique id of each vendor and inside that the data
$postRef = $database->getReference('vendors')->push([
    'name' => $name,
    'number' => $number ,
    'fooditems' => $fooditem ,
    'ratings' => $rating,
    'latitude' => $latitude,
    'longitude' => $longitude,
    'open' => $open,
    'close' => $close,
]);
header('Content-type: application/json');
//changes the header type to application/json
?>
```

dynamicratings.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
//including all the preloaded functional files
$serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/serviceAccount.json');
//includes the firebase service account json file
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    $ref = $database->getReference('/vendors');
    $vendors = $ref->getValue();
    //gets the reference of the current database at the particular child
node
    $vendorname = "";
    $usrrat = "";
    $finalrat = "";
    if($_SERVER["REQUEST_METHOD"] == "POST")
    {
        //requesting user's ratings by post method
        $vendorname = $_POST['vendors'];
        $usrrat = $_POST['ratings'];
        foreach ($vendors as $key => $value)
    {
        //loops through each and every child vendor
        if($vendorname === $value['name'])
         {
            //checks if the vendorname in the database is equal to entered
user id
            $rat = $value['ratings'];
            $finalrat = $rat + $usrrat;
            $finalrat = $finalrat/2;
            //takes the aggregate of user entered rating and pre loaded
rating in the database
            $database->getReference('/vendors/'.$key.'/ratings')-
>set($finalrat);
            //updates the aggregate rating in the database
            //echo json_encode($finalrat);
            $json_array = array("ratings" => $finalrat);
            echo json_encode($json_array);
         }


    }
    }
header('Content-type: application/json');
//changes the header type to application/json
?>
```

location.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
//including all the preloaded functional files
$serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
//includes the firebase service account json file
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    $ref = $database->getReference('/vendors');
    $vendors = $ref->getValue();
    //gets the reference of the current database at the particular child
node
    $usrlat ="";
    $usrlon ="";
    $endlat1 = "";
    $endlon1 = "";
    $endlat2 = "";
    $endlon2 = "";
    $json_array = array();
        $usrlat = $_GET["usrlat"];
        $usrlon = $_GET["usrlon"];
        //getting user location
        foreach($vendors as $key =>$value)
        {
            //loops through each and every child vendor
            $endlat1 = $usrlat+0.009009009;
            $endlon1 = $usrlon+0.009009009;
            $endlat2 = $usrlat-0.009009009;
            $endlon2 = $usrlon-0.009009009;
            //adding up the equivalent degree of latitude and longitude of 
1km radius

        if($value['latitude']<=$endlat1 && $value['longitude']<=$endlon1 && 
$value['latitude']>=$endlat2 && $value['longitude']>=$endlon2)
        {
            //checking if the vendor is in 1km raidus of the user's 
location
            array_push($json_array,$value);
            //pushing the whole content child of vendor into the array
        }
        }
        $myarr = array("data" => $json_array);
        //adding data key in associative array and passing its value of 
array containing all the vendors list
        header('Content-type: application/json');
        //changes the header type to application/json
        echo json_encode($myarr);
        //printing the output in json format
    ?>
```

retrieve.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;

$serviceAccount = 
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
$database = $firebase->getDatabase();

if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $input=$_POST['vendor'];
    //Input validation for string
    $rootSnapshot = $database->getReference("/")->getSnapshot();
    if($rootSnapshot->hasChild($input) === true)
    {
    $vendorReference = $database->getReference($input);
    echo json_encode($vendorReference->getValue());
    }
    else
    {
        $error = array("msg"=>"No Vendor Found");
        //$error->msg="No Vendors Found";
        $myJson=json_encode($error);
        echo $myJson;
    }
}
//$database->getReference('vendors')
```

```
  // ->set([
     //  'name' => 'damodar',
     //  'number' => '7738877342',
      // 'fooditems' => 'South Indian and Beverages',
      //]);

if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    //requesting input of food joints by post method
    $input1 = $_POST['name'];
    $input2 = $_POST['number'];
    $input3 = $_POST['fooditems'];
    $rating = $_POST['ratings'];

$postData = array("");
$postRef = $database->getReference('vendors')->push([
    'name' => $input1 ,
    'number' => $input2 ,
    'fooditems' => $input3 ,
    'ratings' => $rating,
]);
//$rating = database->getreference('/vendors/$postKey');
echo $rating;

$postKey = $postRef->getKey();
echo $postKey;
}
$count = 1;
if($_SERVER["REQUEST_METHOD"] == "POST")
{
    $vendor = $_POST['vendors'];
    $key = $database->getReference('/vendors/'.$postKey);
    echo $key;
}
$key1 = $database->getReference('/vendors/'.$postKey.'/name');
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    $ref = $database->getReference('/vendors');
    $ven = $ref->getValue();
    foreach ($ven as $value) {
        if($value===$vendor)
        {
            $usrrat = $_POST['userrating'];
            $count++;
            $frating = $rating+$usrrat;
            $frating = $frating/$count;
            echo $frating;
        }
    }
}
?>
```

reviewpost.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
//including all the preloaded functional files
    $serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
    //includes the firebase service account json file
    $firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    if ($_SERVER["REQUEST_METHOD"] == "POST")
    {
        //getting review of the vendor by the particular username using
post request
        $review = $_POST["review"];
        $username = $_POST["usrname"];
        //$vendorid = $_POST("vendorid");
        $vendorname = $_POST["vendorname"];
        $postdata = array("");
        $postRef = $database->getReference("reviews")->push([
            'username' => $username,
            'review' => $review,
            'jointname' => $vendorname,
        ]);
        //pushing the data into the database
    }
?>
```

reviewretrieve.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
    //including all the preloaded functional files
$serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
//includes the firebase service account json file
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
 $database = $firebase->getDatabase();
    //gets the database of the current service account
    $ref = $database->getReference('/reviews');
    $reviews = $ref->getValue();
    //gets the reference of the current database at the particular child
node
$vendorname;
$json_array= array();
$username;
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
    //taking the vendorname's name as post request
    $vendorname = $_POST["vendorname"];

}
$reviews['username']['review']=  array();
//creating a new associative array of reviews
$arrayName = array();
//storing array
foreach($reviews as $key => $value)
{
    //loops through each and every child vendor
    if(isset($value['jointname']))
    {
        //checks whether there the joint name has a value in it
    if(strcmp($vendorname,$value['jointname'])==0)
    {
        //comparing the review database vendor name and the post request 
vendor name
        $username = $value['username'];
        //storing the value of the username key of database into a variable
        $review = $value['review'];
        //storing the value of the review key of database into a variable
        $newReview = array('username' => $username,'review' => $review);
        //storing value of username and review into the array
        array_push($arrayName,$newReview);
        //pushing the contents into another array
    }
}
}
        $json_array = array("data" =>$arrayName);
        //adding data key in associative array and passing its value of 
array containing all the vendors list
        echo json_encode($json_array);
        //printing the output in json format
        header('Content-type: application/json');
        //changes the header type to application/json

?>
```

search.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
    //including all the preloaded functional files
$serviceAccount = 
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
//includes the firebase service account json file
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $i=0;
    $query="";
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    $ref = $database->getReference('/vendors');
    $vendors = $ref->getValue();
```

```
        //gets the reference of the current database at the particular 
child node
    $query = $_GET["search"];
    //getting the search query by get request
    $count = strlen($query);
    //counts the length of the query string
    $substring = "";
    $name = "";
    $item = "";
    $substring1 = "";
    $itemcount = 0;
    $namecount = 0;
    $max = 0;
    $flag = 0;
    $json_array = array();
    foreach($vendors as $key =>$value)
    {
        //loops through each and every child vendor
        $flag = 0;
        $item = $value['fooditems'];
        //stores the string value of each vendor's fooditems into a 
variable
        $itemcount = strlen($item);
        //stores the length of the string of food items of each vendor
        $name = $value['name'];
        //stores the string value of each vendor's name into a variable
        $namecount = strlen($name);
        //stores the length of the string of name of each vendor
        if($itemcount>$namecount)
        {
            $max = $itemcount;
        }
        else
        {
            $max = $namecount;
        }
        //this if-else block is used to count the maximum length of two 
string name and string fooditems to increase the optimization
        for($i=0;$i<$max;$i++)
        {
            //loops throught each and every character of the string
            $substring = substr($value['name'],$i,$count);
            $substring1 = substr($value['fooditems'],$i,$count);
            //takes the substring till ith character
            if(strcmp($substring,$query)==0 || 
strcmp($substring1,$query)==0)
            {
                //compares the substring and the query
                if($flag == 0) {
                    array_push($json_array,$value);
                //pushing the whole content child of vendor into the array
                    $flag = 1;
                    //using flag variable not to repeat the output vendors 
list
                }
            }
            else
            {
            }
        }
    }
    $myarr = array("data" => $json_array);
    //adding data key in associative array and passing its value of array
containing all the vendors list
    header('Content-type: application/json');
    //changes the header type to application/json
    echo json_encode($myarr);
    //printing the output in json format
?>
```

searchtiffin.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
    //including all the preloaded functional files
$serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
//includes the firebase service account json file
$firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $i=0;
    $query="";
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    $ref = $database->getReference('/tiffin');
    $tiffin = $ref->getValue();
    //gets the reference of the current database at the particular child
node
    $query = $_GET["search"];
    //getting the search query by get request
    $count = strlen($query);
    //counts the length of the query string
    $substring = "";
    $name = "";
    $item = "";
    $substring1 = "";
    $itemcount = 0;
    $namecount = 0;
    $max = 0;
    $flag = 0;
    $json_array = array();
    foreach($tiffin as $key =>$value)
    {
        //loops through each and every child vendor
        $flag = 0;
        $item = $value['fooditems'];
        //stores the string value of each vendor's fooditems into a
variable
        $itemcount = strlen($item);
        //stores the length of the string of food items of each vendor
        $name = $value['name'];
        //stores the string value of each vendor's name into a variable
        $namecount = strlen($name);
        //stores the length of the string of food items of each vendor
        if($itemcount>$namecount)
        {
            $max = $itemcount;
        }
        else
        {
            $max = $namecount;
        }
                //this if-else block is used to count the maximum length of
two string name and string fooditems to increase the optimization
        for($i=0;$i<$max;$i++)
        {
            $substring = substr($value['name'],$i,$count);
            $substring1 = substr($value['fooditems'],$i,$count);
            //takes the substring till ith character
            if(strcmp($substring,$query)==0 ||
strcmp($substring1,$query)==0)
            {
                //compares the substring and the query
                if($flag == 0) {
                    array_push($json_array,$value);
                    //pushing the whole content child of vendor into the
array
                    $flag = 1;
                    //using flag variable not to repeat the output vendors
list
                }
            }
            else
            {
            }
        }

    }
    $myarr = array("data" => $json_array);
    //adding data key in associative array and passing its value of array
containing all the vendors list
    header('Content-type: application/json');
    //changes the header type to application/json
    echo json_encode($myarr);
    //printing the output in json format
?>
```

tiffindatabase.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
//including all the preloaded functional files
    $serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
    //includes the firebase service account json file
    $firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    if ($_SERVER["REQUEST_METHOD"] == "POST")
    {
        //requesting input of food joints by post method
        $name = $_POST["name"];
        $number =$_POST["number"];
        $fooditem =$_POST["fooditem"];
        $rating = $_POST["rating"];
        $postdata = array("");

//pushing every data of the vendors with vendors as root element and then
unique id of each vendor and inside that the data
        $postRef = $database->getReference("tiffin")->push([
            'name' => $name,
            'number' => $number,
            'fooditems' => $fooditem,
            'ratings' => $rating,
        ]);
    }
    header('Content-type: application/json');
    //changes the header type to application/json
?>
```

tiffinratings.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
    //including all the preloaded functional files
    $serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
    //includes the firebase service account json file
    $firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    $ref = $database->getReference('/tiffin');
    $tiffin = $ref->getValue();
    //gets the reference of the current database at the particular child
node
    $vendorname = "";
    $usrrat = "";
    $finalrat = "";
    $rat = "";
    if($_SERVER["REQUEST_METHOD"] == "POST")
    {
        //requesting user's ratings by post method
        $vendorname = $_POST["vendor"];
        $usrrat = $_POST["ratings"];
        foreach ($tiffin as $key => $value)
    {
        //loops through each and every child vendor
        if($vendorname === $value["name"])
         {
            //checks if the vendorname in the database is equal to entered
user id
```

```
            $rat = $value["ratings"];
            $finalrat = $rat + $usrrat;
            $finalrat = $finalrat/2;
            //takes the aggregate of user entered rating and pre loaded
rating in the database
            $database->getReference('/tiffin/'.$key.'/ratings')-
>set($finalrat);
            //updates the aggregate rating in the database
         }
    }
}
?>
```

Tiffinreview.php

```
<?php
require __DIR__.'/vendor/autoload.php';
    use Kreait\Firebase\Factory;
    use Kreait\Firebase\ServiceAccount;
    use Kreait\Firebase;
//including all the preloaded functional files
    $serviceAccount =
ServiceAccount::fromJsonFile(__DIR__.'/ServiceAccount.json');
    //includes the firebase service account json file
    $firebase = (new Factory)
    ->withServiceAccount($serviceAccount)
    ->create();
    $database = $firebase->getDatabase();
    //gets the database of the current service account
    if ($_SERVER["REQUEST_METHOD"] == "POST")
    {
        //getting review of the vendor by the particular username using
post request
        $review = $_POST["review"];
        $username = $_POST["usrname"];
        //$vendorid = $_POST("vendorid");
        $vendorname = $_POST["vendorname"];
        $postdata = array("");
        $postRef = $database->getReference("reviews")->push([
            'username' => $username,
            'review' => $review,
            'jointname' => $vendorname,
        ]);
        //pushing the data into the database
    }
?>
```

11. RESULTS

In this chapter, the results of our project are shown with a short description of the screens. The screens are as follows: -

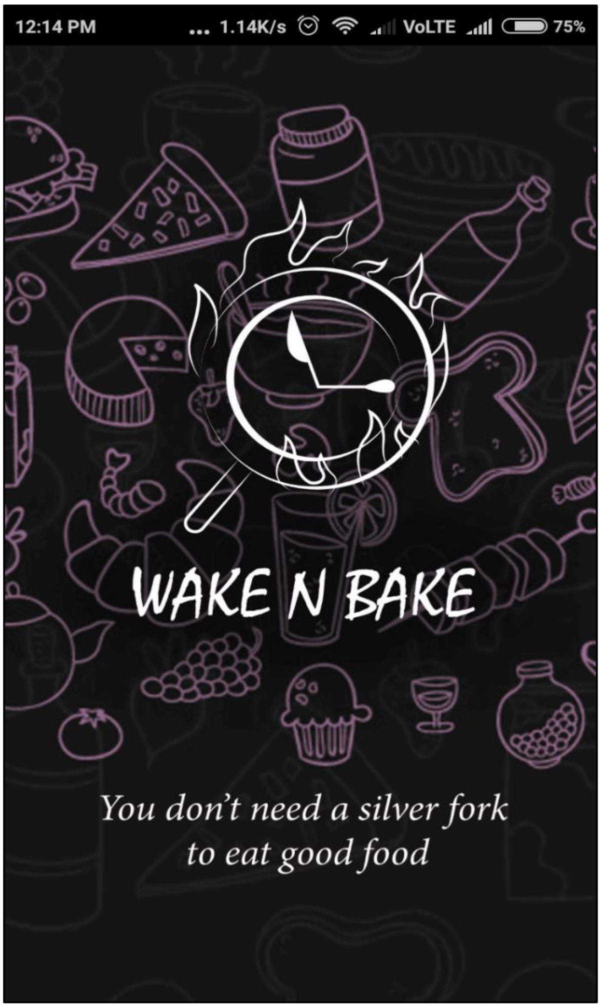


Figure 6 : Splash Screen

- This is the first screen when the user opens the application.
- This screen also checks that the Internet connection is available or not.

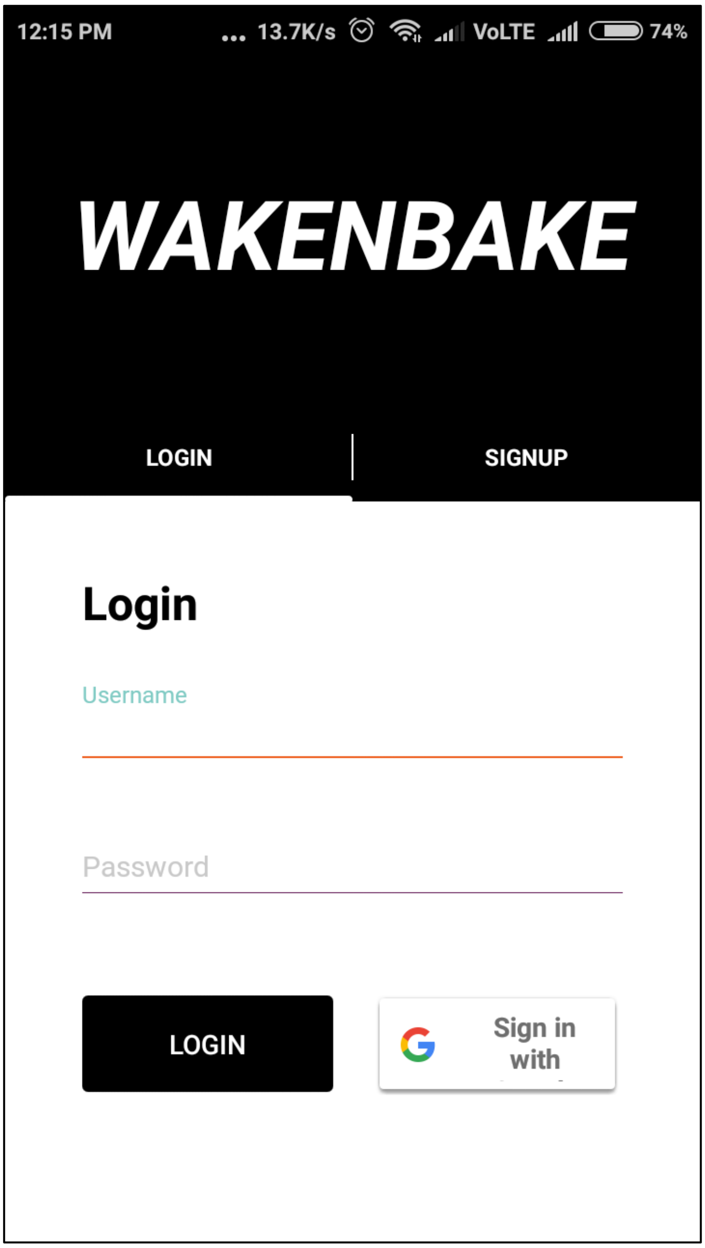


Figure 7 : Login Screen

- After Splash Screen, The login screen is visible to the user.
- The user can also choose to sign-in to the application using Google Sign In.
- The user has to enter his/her credentials to log in.

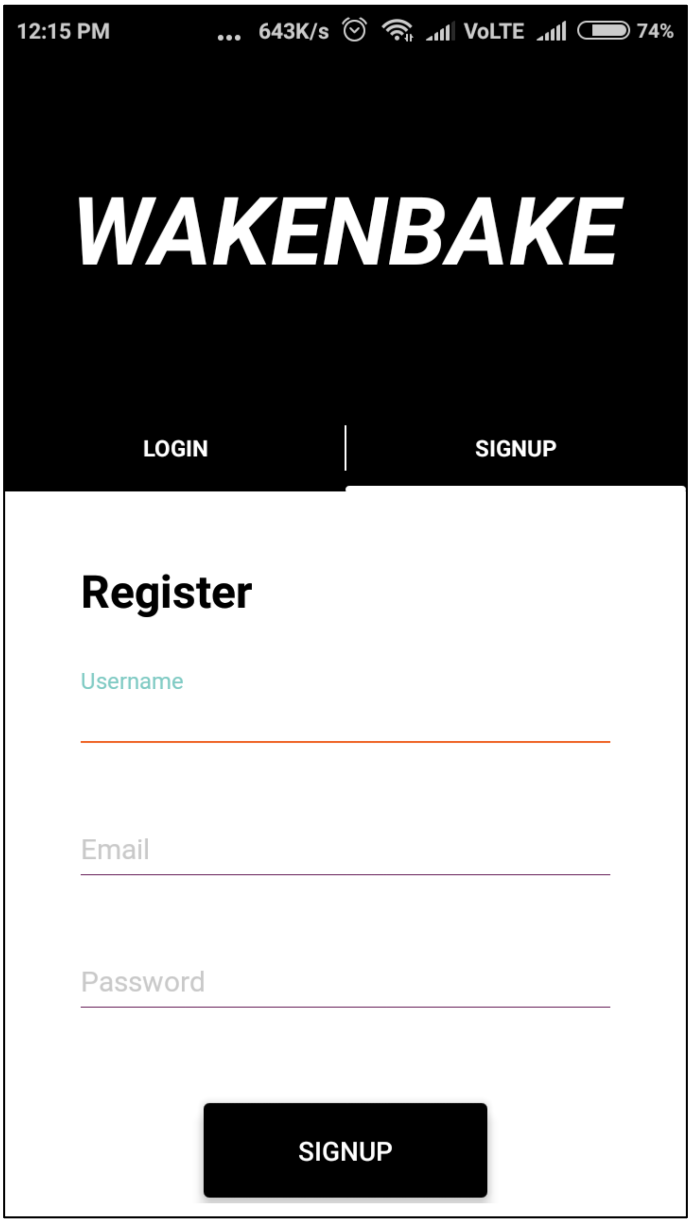


Figure 8 : Register Screen

- If the user is using the application for the first time, he/she has to register to the application.
- The user has to enter his username, email, password to register to the application.

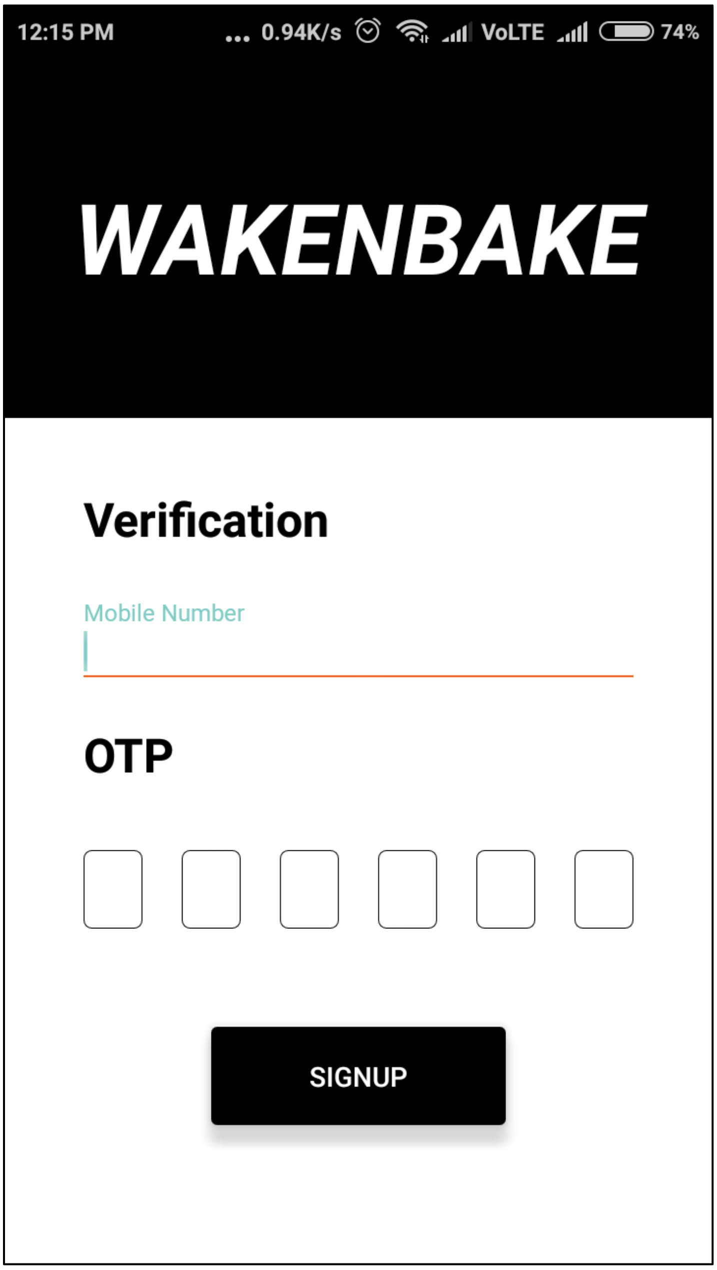
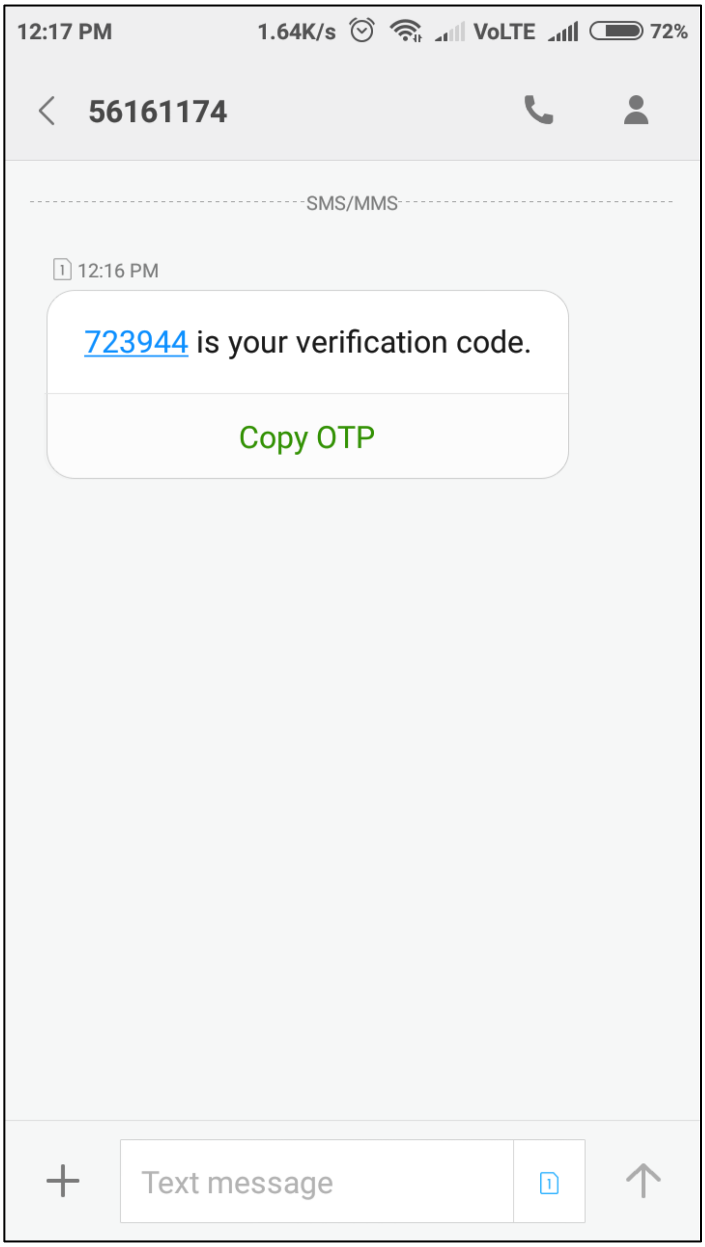


Figure 9 : OTP Screen

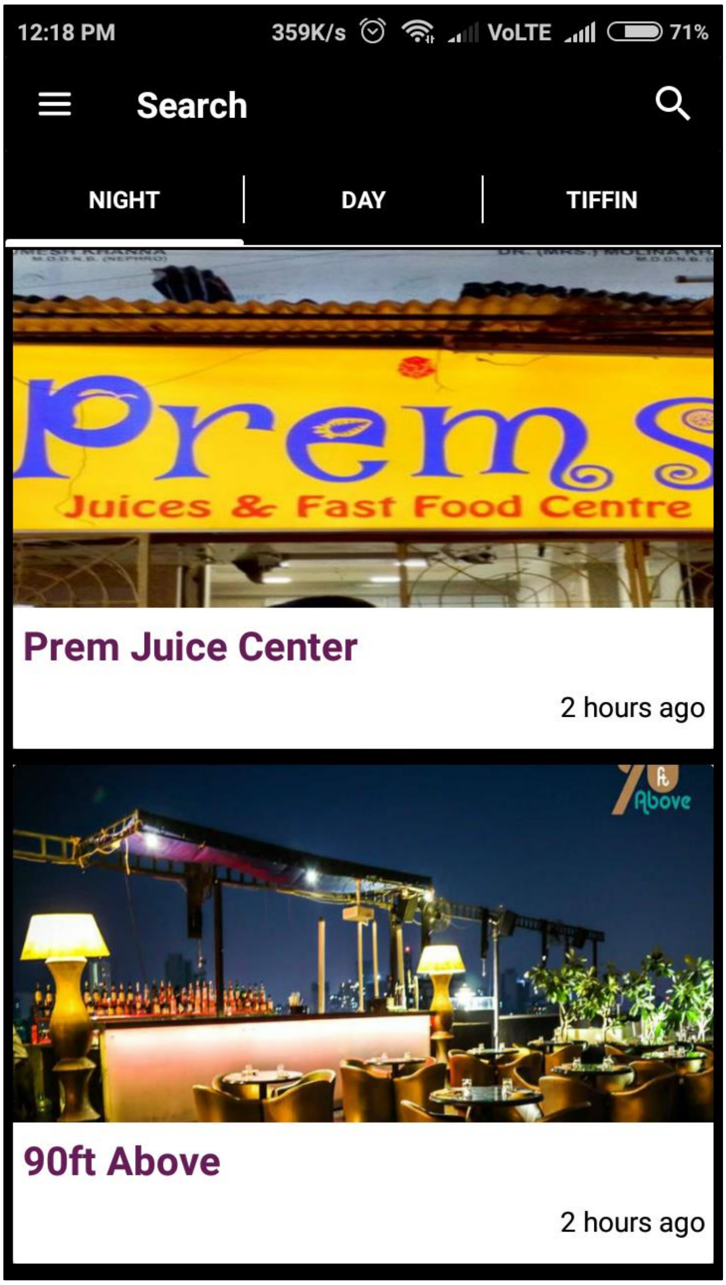
- There is a 2-factor authentication for the user who have registered in the app for the first time.
- The user has to enter the 10 digit mobile number.
- The Firebase Authenticator will send an OTP to the entered mobile number.



723944 is your verification code.

Figure 10 : Example of OTP

- The above image is an example of the OTP sent by Firebase Authentication.
- The user has to enter the OTP and press Submit.
- Sometimes the OTP is automatically detected if the mobile number entered is in the same mobile phone which is registering for the application.



12:18 PM 359K/s VoLTE 71%

Search

NIGHT | DAY | TIFFIN

Prem Juice Center

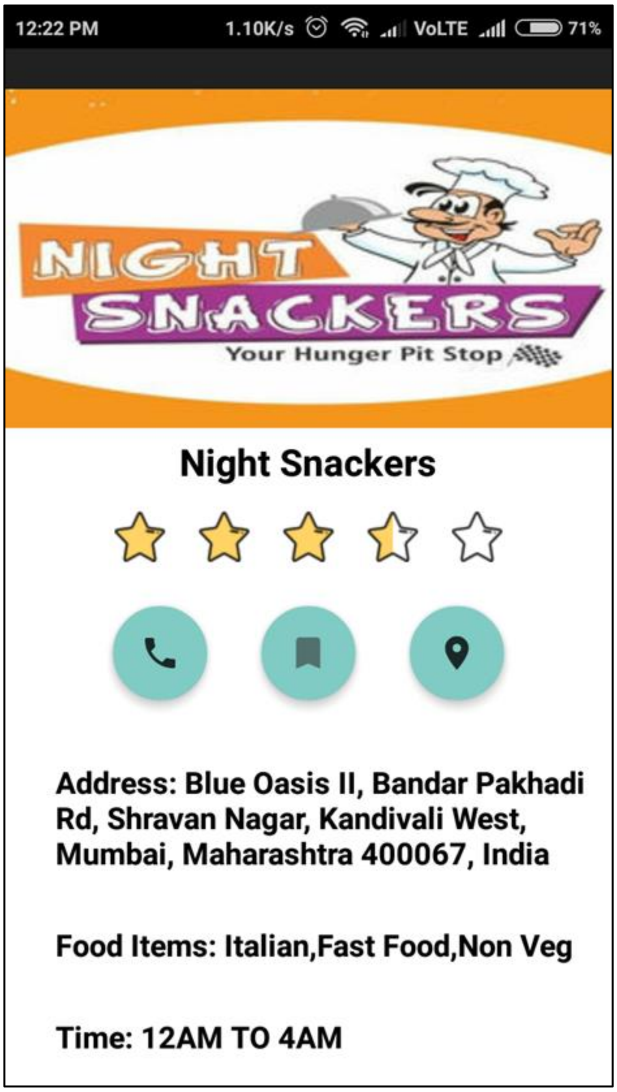
2 hours ago

90ft Above

2 hours ago

Figure 11 : List of available Food Joints during Night

- The user’s current location will be detected using Google Maps API.
- The list of available food joints will be displayed to the user which are within 1KM of the user’s current location.



Night Snackers

☆☆☆☆☆

☎

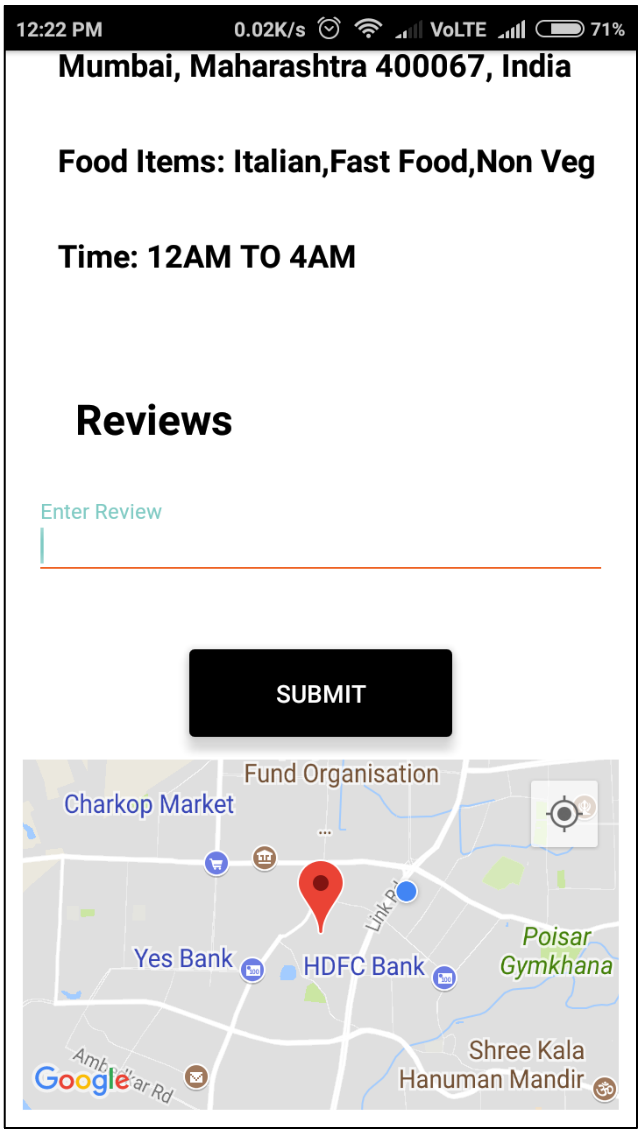
Address: Blue Oasis II, Bandar Pakhadi Rd, Shravan Nagar, Kandivali West, Mumbai, Maharashtra 400067, India

Food Items: Italian,Fast Food,Non Veg

Time: 12AM TO 4AM

Figure 12 : Details of the food joint

- The details of the food joint will be displayed to the user.
- The user can call the food joint, can save this place as a bookmark, and provides its feedback using reviews and ratings.
- The user has to slide down to see the location of the food joint.
- The address, cuisine provided by the food joint & the time will also be displayed to the user.



12:22 PM 0.02K/s VoLTE 71%

Mumbai, Maharashtra 400067, India

Food Items: Italian,Fast Food,Non Veg

Time: 12AM TO 4AM

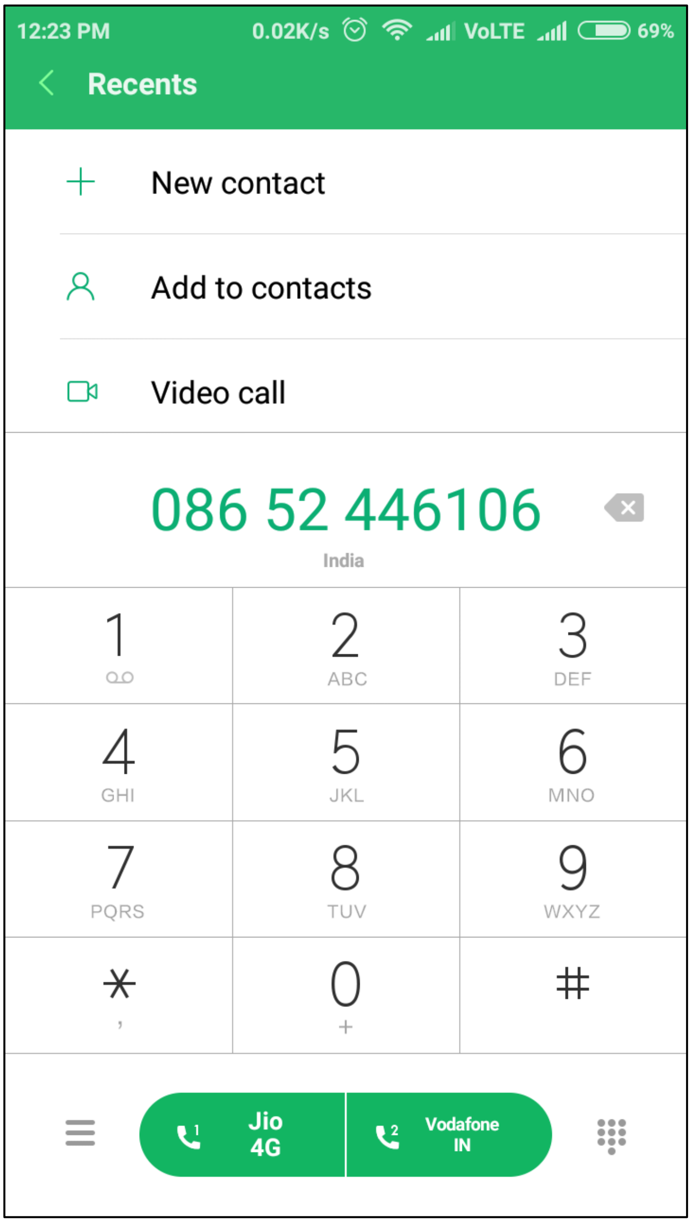
Reviews

Enter Review

SUBMIT

Figure 13 : Reviews and location of the food joint on Google Maps

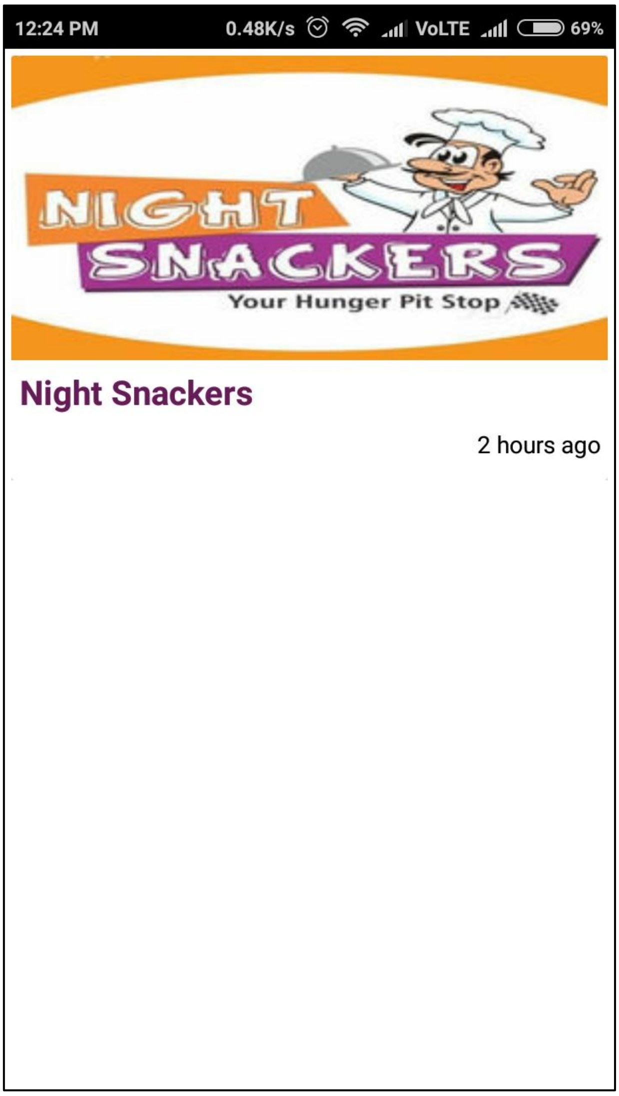
- The user also provides his/her feedback by giving reviews about the food joint.
- The user also finds its current location (blue dot) on the Google Maps and also the location of the food joint (red pin marker).



086 52 446106

Figure 14 : Calling the food joint

- Once the user clicks on the call button, the user will be redirected to the dialer of the mobile phone.
- The user can choose the service operator to call.
- The mobile number of the food joint is only visible in the dialer.



12:24 PM 0.48K/s VoLTE 69%

Night Snackers

2 hours ago

Figure 15 : Food joints which are bookmarked by the user

- When the user clicks the bookmark button, the place is added to the database.
- First, the database is checked for the existing food joint.
- If the place exists in the database, it displays a message to the user that the food joint is already added to the bookmarks, else it will be added to bookmarks.

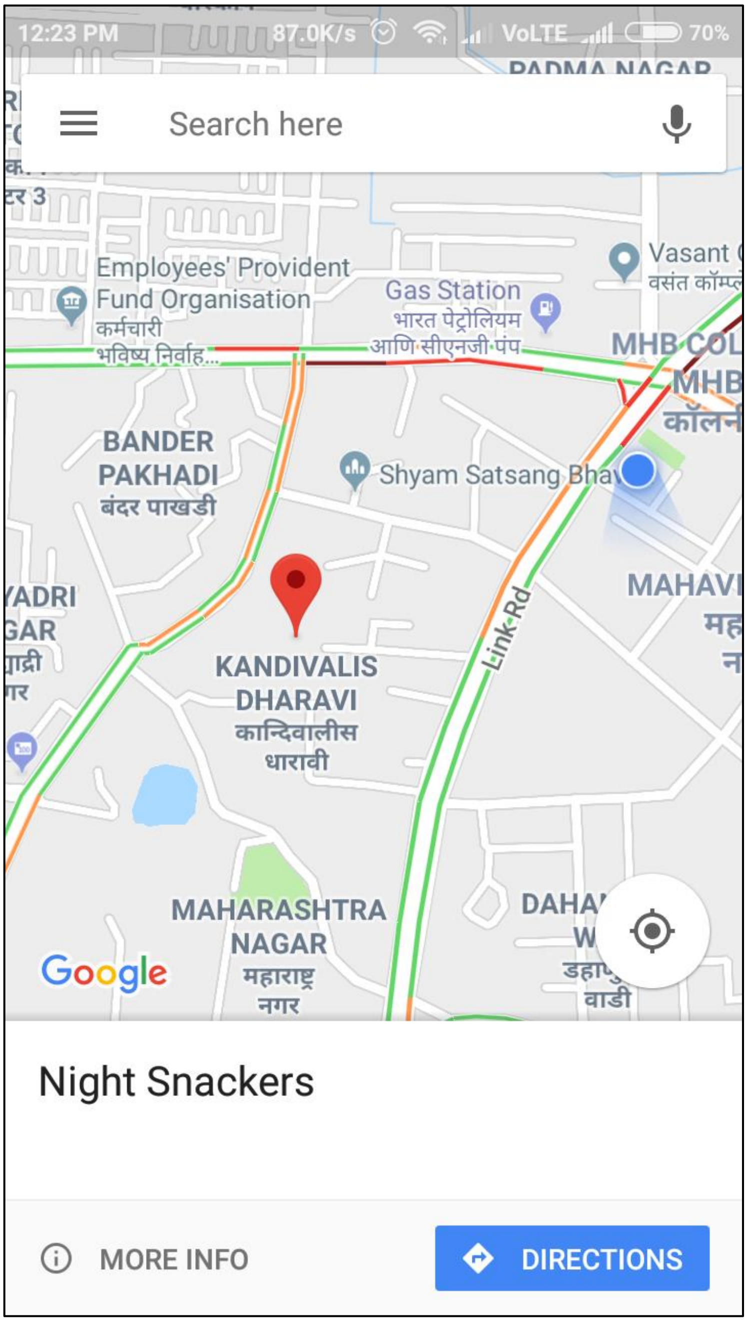


Figure 16 : Food Joint in Google Maps

- When the user clicks on the Google Maps Snippet, the user will be directed to Google Maps.
- The user can find additional details about the food joint in the Google Maps.

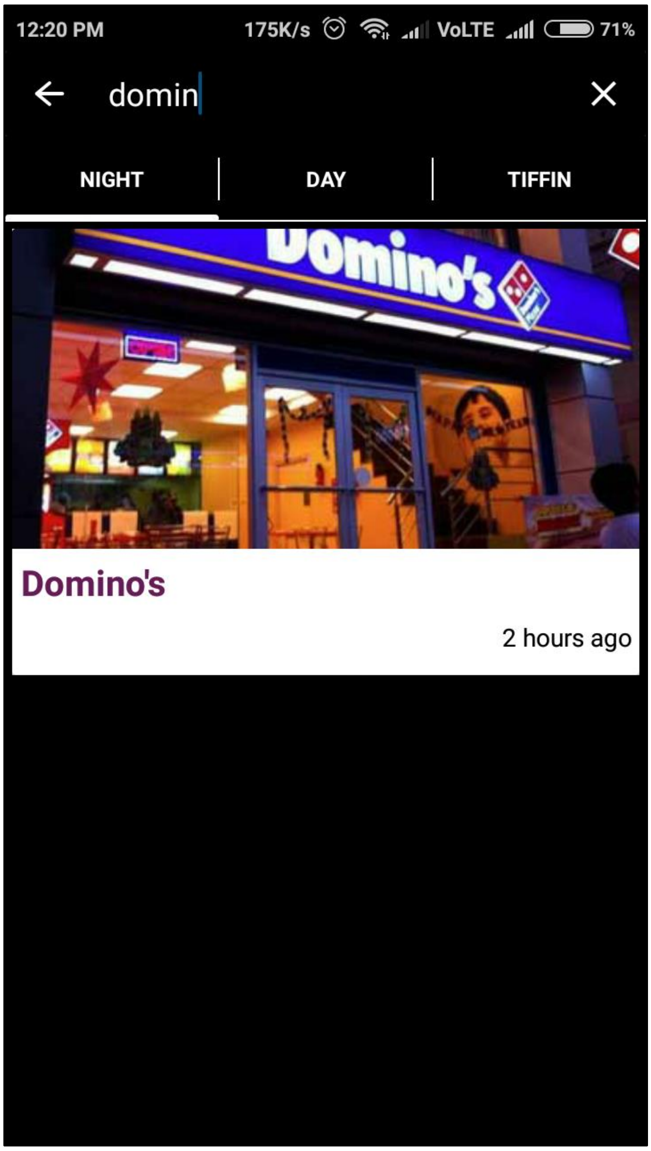


Figure 17 : Searching for a specific food joint

- The filtering of the food joints can be done by searching the specific place by its name.

12. FEATURES

- Search easily for food joints by location, cuisine and name at that time 1:00am-6:00am)
- View menus, pictures, phone numbers, directions, reviews, ratings, and other information you need to find the best food joint near you.
- Rate and review the food joints you’ve been to and share your foodie moments directly from the app.
- Maintain an easy reference list of your favourites or places you want to visit with bookmarks.
- Use map view to explore food joints around you.

13. LIMITATIONS

- This application applicable for particular locality which is between Borivali and Malad.

14. APPLICATIONS

- Find local food joints near you.
- Select food joints based on reviews, ratings and also on the type of cuisines select by user.

15. FUTURE SCOPE

- Can be further extended to the food joints available throughout the city.
- Can also be used by home-made food vendors to set up their respective businesses in various types of cuisines.
- Online ordering can be started using this application by the food joints and home-made vendors.
- Can be extended to provide a more advanced search feature by taking pin code as the input.
- Home delivery feature can also be added in the near future.
- The application can be improved in many ways and can be extended to support more devices like the tablets and iOS devices.

16. CONCLUSION

This is our first attempt in developing a mobile application which gave us a basic understanding of development and challenges of mobile application development. We also understood how to work in a team to fufil the user’s needs and how projects can be done by using VCS (Version Control System) with the most skilled developers across the world. We also understood how to use some GitHub Libraries which are developed by the developers from all over the world and to use them in our project.

The main aim of the project is to provide an easy to use application for searching the places during day and night time as well as providing the details of the home-made food vendors. The application has been implemented and tested on real devices.

The application can be further extended to iOS devices from Android by using some cross platform technologies such as React-Native, Flutter, etc. The application can also be featured using online ordering and home delivering features by collaborating with the companies such as Swiggy for providing home delivery services.

17. REFERENCES

[1]. Rajinder Singh, Department of Computer Science and Applications DCSA Panjab University SSGRC Hoshiarpur, An Overview of Android Operating System and Its Security Features, dated February 2014.

https://pdfs.semanticscholar.org/11f4/b8efd1a9af746f17ac5e8d6a789bd3c3a9b7.pdf

[2]. Bob Brown, A Gentle Introduction to PHP, School of Computing and Software Enginering, copyright in 2002.

http://ksuweb.kennesaw.edu/~rbrow211/papers/php1.html

[3]. GeekyAnts, Introduction to Firebase, blog post published on December 28, 2017.

https://hackernoon.com/introduction-to-firebase-218a23186cd7

[4]. Stefan Otte, Introduction to Version Control System, Computer Systems and Telematics Institute of Computer Science Freie Universitat Berlin, Germany.

https://pdfs.semanticscholar.org/5093/c4b55ad419379ad22e0c60008f133a2b6e3b.pdf

[5]. A book of Version Control System using Git by Jon Loeliger published in May 2009.

https://www.foo.be/cours/dess20122013/b/OReilly%20Version%20Control%20with%20GIT.pdf

[6]. Xiao Li, A Tutorial for Git and Github, Department of Informatics, University of Zurich.

https://www.ifi.uzh.ch/dam/jcr:ff780599d5e24d05b9231c333cbf2842/A%20Tutorial%20for%20GitHub.pdf

[7]. A Presentation by Pratheepan Raveendranathan on Introduction to Google API, University of Minnesota Duluth.

https://www.d.umn.edu/~rave0029/research/GoogleAPI.ppt

[8]. Pankti Doshi, Pooja Jain, Abhishek Shakwala on Location Based Services and Integration of Google Maps in Android, NMIMS University, dated March 3, 2014.

http://www.ijecs.in/issue/v3-i3/30%20ijecs.pdf

[9]. Leigh Williamson, Roland Barcia, Omkar Chandgadkar, Ashish Mathur, Soma Ray, Darrell Schrag, Roger Snook, Jianjun Zhang, ‘Mobile: The New Generation of Information Technology ‘, Jan 12, 2016.

[10]. Ian G. Clifton, ‘Using Advanced Techniques in Android User Interface Design’, Jan 12, 2016.

[11]. Carmen Delessio, Shane Conder, Lauren Darcey, Working with ImageViews and Bitmaps in Android Application Development', Jul 27, 2015.

[12]. B.M. Harwani, 'Accessing Google Maps in Android Apps', Dec 30, 2013.

[13]. Joseph Annuzzi, Shane Conder, Lauren Darcey, 'Advanced Android Application Development: Handling Advanced User Input', dated Nov 27, 2014.

[14]. Paresh Mayani, 'JSON Parsing Library', Apr 02, 2014.