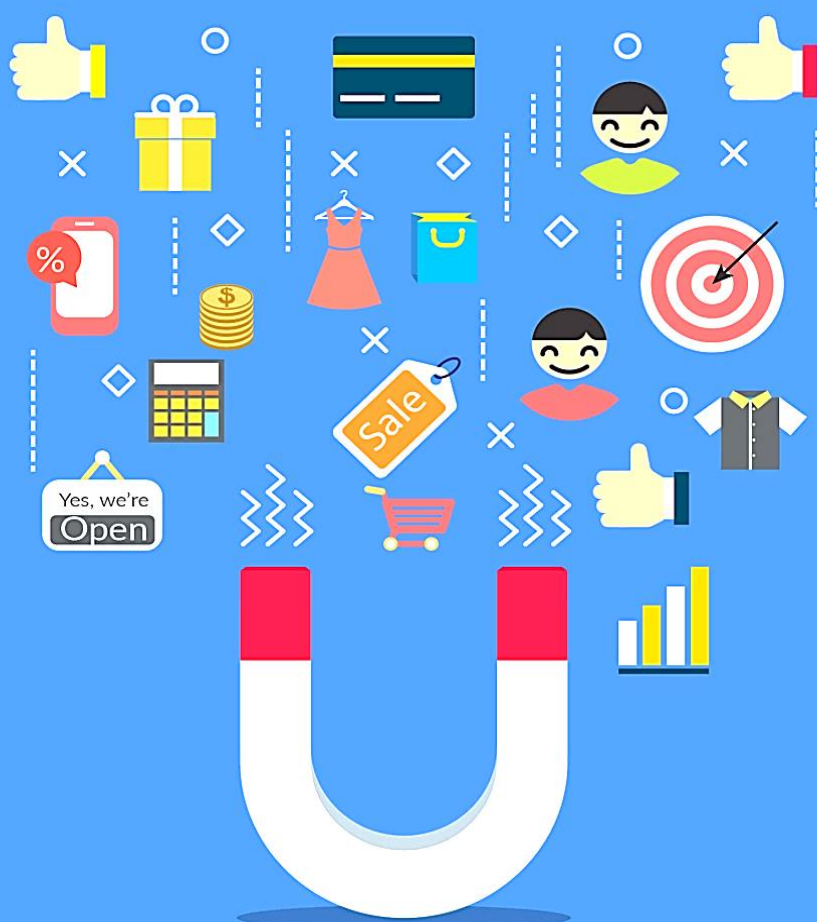


2018

OfferMagnet

A context-aware platform for products and offers



"Offers and promotion locator"

Dr. Alaa Alghamery
Computer Science Dept. –
Zagazig University



OfferMagnet System

Faculty of Computing and Information - Zagazig University

Department of Computer Science

2017-2018

Under the supervision of

Dr.Alaa Alghamery

Eng.Ahmed Alrefaey

Students

- Esraa Abdel-razik Mahmoud
- Nesma Tharwat Mansour
- Alaa fathi Awadallah
- Alshimaa Moner Atia Elnady
- Doaa Gamal Abo-younis
- Esraa Mohamed Ibrahim
- Ali Omar Elsayed

Acknowledgment

First and foremost, we would like to express our highest appreciation to supportive to Dr. Alaa El-Ghamry. His supervision and support that gave us truly help during the period of conducting our project. we also happy to present our gratefully acknowledge to Dr. Ahmed Salah who has been so warmth and kind to provide sincere assistance and good cooperation during this period.

Table of Contents

Chapter One: Introduction

1.1.	Problem definition.....	8
1.2.	Importance of Problem	8
1.3.	Current solution.....	8
1.4.	Our solution	9
1.5.	Impact of our solution.....	9

Chapter Two: Analysis

2.1.	Sequence diagram.....	10
2.2.	System requirement.....	14
2.2.1.	Requirement analysis	15
2.3.	List of requirement	15
2.3.1.	Functional requirements.....	15
2.3.2.	Non-Functional requirements.....	15
2.4.	System analysis.....	19
2.4.1.	Use Case	19
2.4.2.	Context Flow Diagram.....	20
2.4.3.	DFD Level o	20
2.5.	Policy Requirements.....	21
2.6.	Training Requirements.....	22
2.7.	General System Requirements:	22
2.7.1.	Major System Capabilities.....	22
2.7.2.	Major System Conditions.....	22
2.7.3.	System Interfaces	22
2.7.4.	System User Characteristics.....	22
2.8.	General System Requirements	23

Chapter Three: External Design Documentation

3.1.	Introduction	24
3.2.	User Interface Design	24
3.2.1.	OfferMagnet Website	26
3.2.2.	Android UI	28

Chapter Four: Implementation

4.1.	Project plane	41
4.1.1.	Schedule	41
4.2.	Hardware, software, tools and languages.....	42
4.3.	Android App Code	42
4.4.	Machine learning code	120
4.4.1.	Introduction RecommenderSystem	120
4.4.2.	Implementation ML codeing	121

List of Figures

Figure 1: SignUp	10
Figure 2: Sign In.....	11
Figure 3: GPS Sequence Diagram	11
Figure 4: Offer Sequence Diagram	12
Figure 5: Request Sequence Diagram	13
Figure 6: Requirement Analysis Factor	14
Figure 7: Use Case Diagram	19
Figure 8: Context Flow Diagram.....	20
Figure 9: DFD Level 0.....	20
Figure 10: Website Interface 1.....	26
Figure 11: Website Interface 2.....	27
Figure 12: website Interface 3	27
Figure 13 : Splash 1	28
Figure 14 : Splash 2	28
Figure 15 : Splash 3	29
Figure 16 : Splash 4	29
Figure 17: Log In.....	30
Figure 18: Sign Up	30
Figure 19: upload image	31
Figure 20: Selecting a Category.....	32
Figure 21: Request Tab	33
Figure 22: Offer Tab	33
Figure 23 : Alert Dialog For Request	34
Figure 24 : Alert Dialog For Request	34
Figure 25: Request Creation 1.....	35
Figure 26: Request Creation 2.....	35
Figure 27 : Offer Details 1	36
Figure 28 : Offer Details 2	36
Figure 29: Offer Creation Step 2	37
Figure 30: Offer Creation Step 1	37
Figure 31: Offer Creation Step 3	38
Figure 32 : Navigation Bar for menu.....	39
Figure 33: Location Feature	40
Figure 34 : Schedule.....	41
Figure 35 : ML Coding	121

1. Introduction

1.1. Problem definition

Different prices and/or level of qualities for the same product or service within the same region is an example of these problems. Here, we reverse the definition of "offer" term. We help users of the same geographic area with a common interest in certain products or services manage to resist unfair prices or unexpected quality of the provided services. Thus, instead of starting an offer from the producers, we initiating it from consumers. The offer is defined as follows. A user or a group of users initiate a request for certain product or services, based on their locations, new users join them, while producers try to attract this group of users through offers. Finally, the best offer is picked-up by the consumer.

1.2. Importance of Problem

The addressed problem has a direct effect on user incomes and their confidence. If we managed to provide the users such a service, then the spent money can be reduced, and most importantly society confidence increase.

1.3. Current solution

No current similar solutions. Applications that provide services close to the proposed application at hand offers lack the integrity among different products and services and are categorized in one product or service, for instance, "Gas Buddy" covers only gas prices and automotive-related services. Other examples of apps that provide a similar service are "YP App" and "Safeway App" do not allow users to report product prices and it depends only on data gathered from the organization itself and

in “shoutit” users can request or make offers, but they do not get them notifications about these offers nor do they appear on their feeds.

1.4. Our solution

The core idea that users of the same geographic area with a common interest in certain Product or services manage to resist unfair prices or unexpected quality of the provide services. To the date, the proposed application has two novel features:

- There is no application connects users interested in a certain product or service in order to attract producers for giving a special price and quality.
- No application provides alert, based on users’ location, for offers and lowest price of products and services of interest.

1.5. Impact of our solution

The proposed application will help societies to fight the unfair price of products or unexpected level of qualities through a touch on the screen. This is reflected in higher values like social collaboration and confidence. In addition, it helps individuals to reduce the spend money. On the other hand, the application is commercially useful, as it connects groups of consumers to producers, and as a result, increase the trading volumes.

2. Analysis

2.1. Sequence diagram

Sequence diagram are interaction diagrams that detail how operations are carried out

- Interaction diagrams model important run time interaction between the parts that makeup the system.
- Show object instances that play the roles defined in the collaboration.
- Show the order of the interaction visually by using the vertical access of the diagram to represent time what the messages are sent and when.
- Show elements as they interact over time.
- Showing interactions or interaction instances.
- Don't show the structural relationships between objects.

The following sequences diagrams will show the sequences parts of our system:

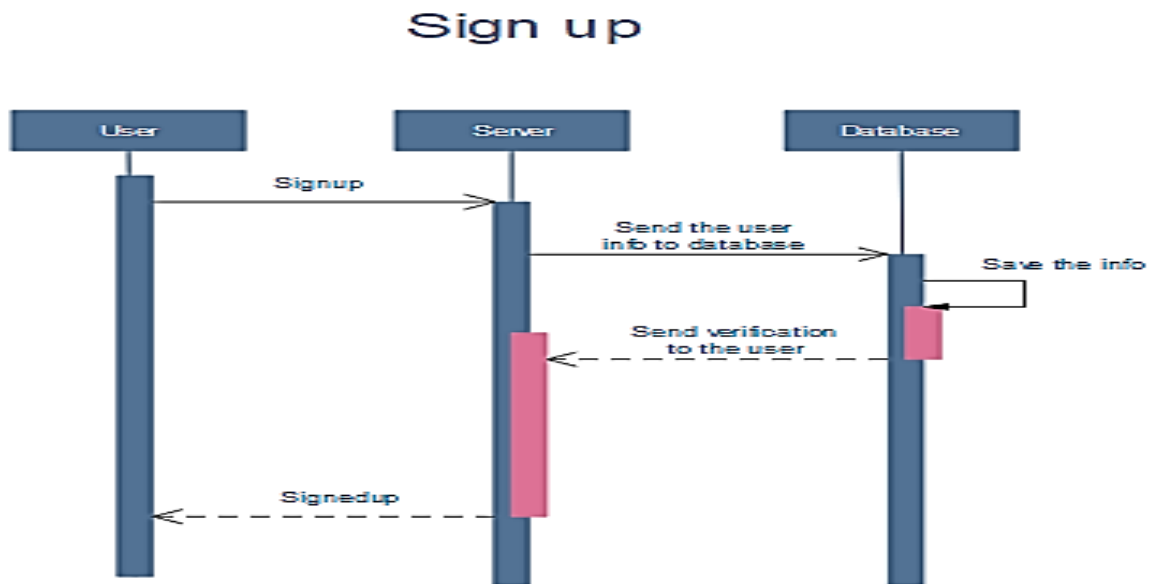


Figure 1: SignUp

Sign in

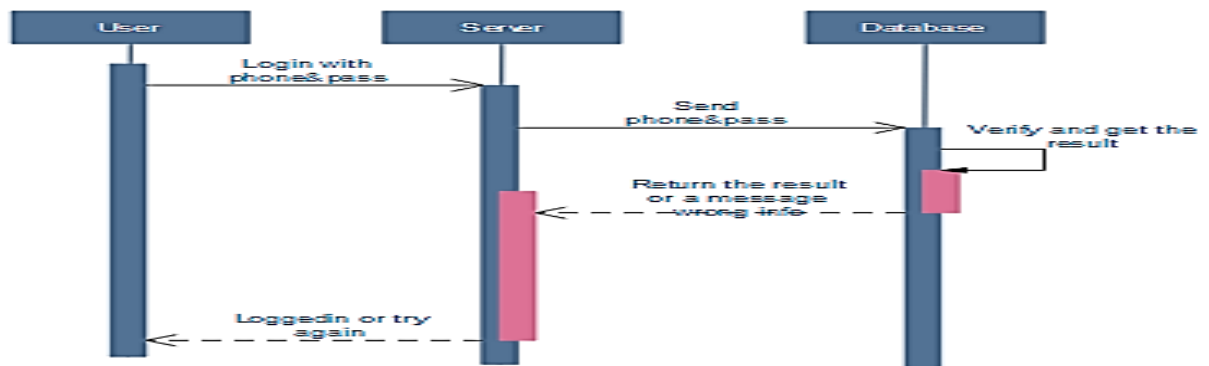


Figure 2: Sign In

GPS

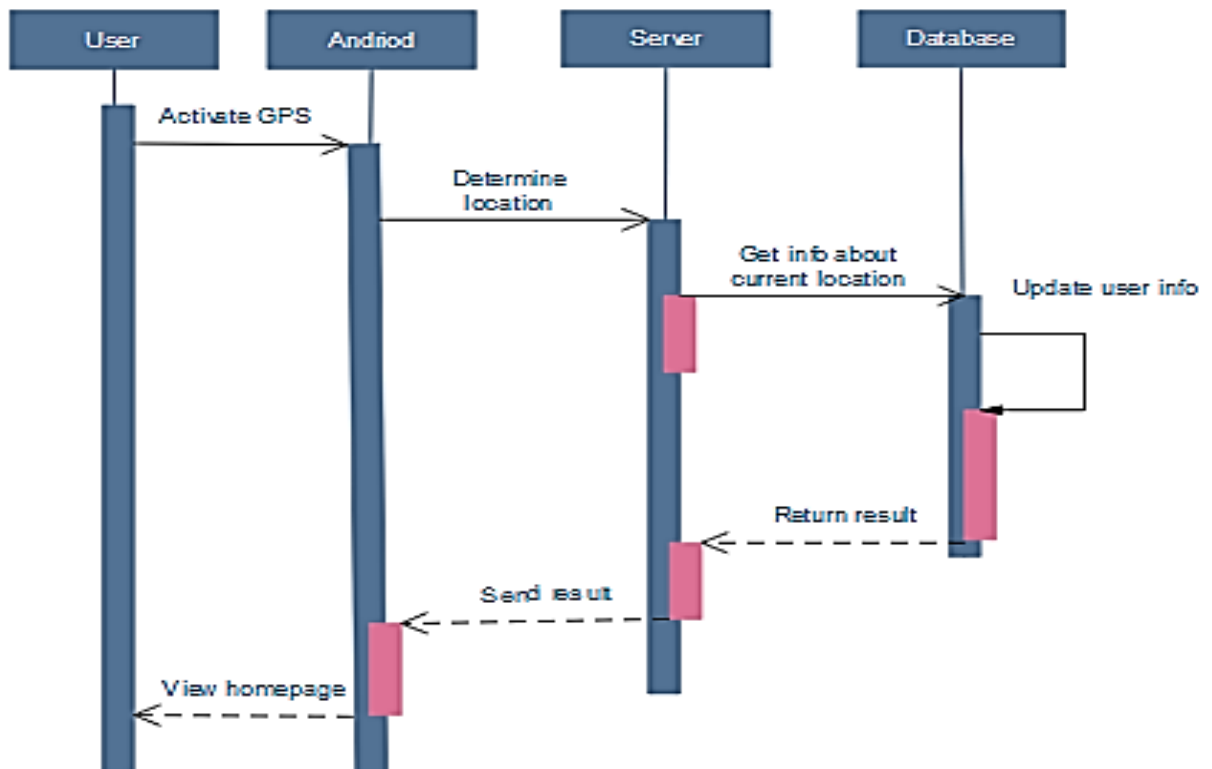


Figure 3: GPS Sequence Diagram

Offer

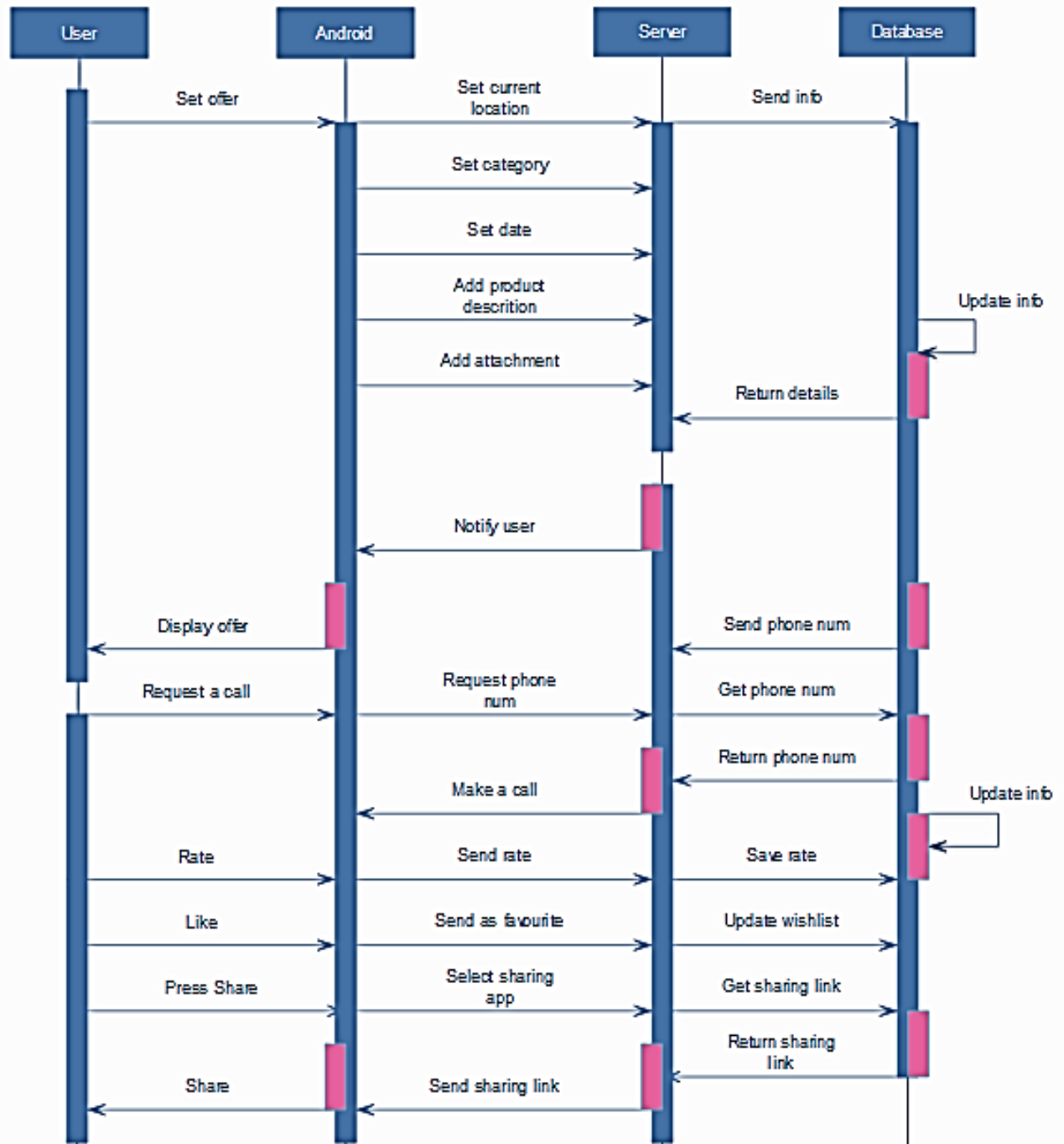


Figure 4: Offer Sequence Diagram

Request

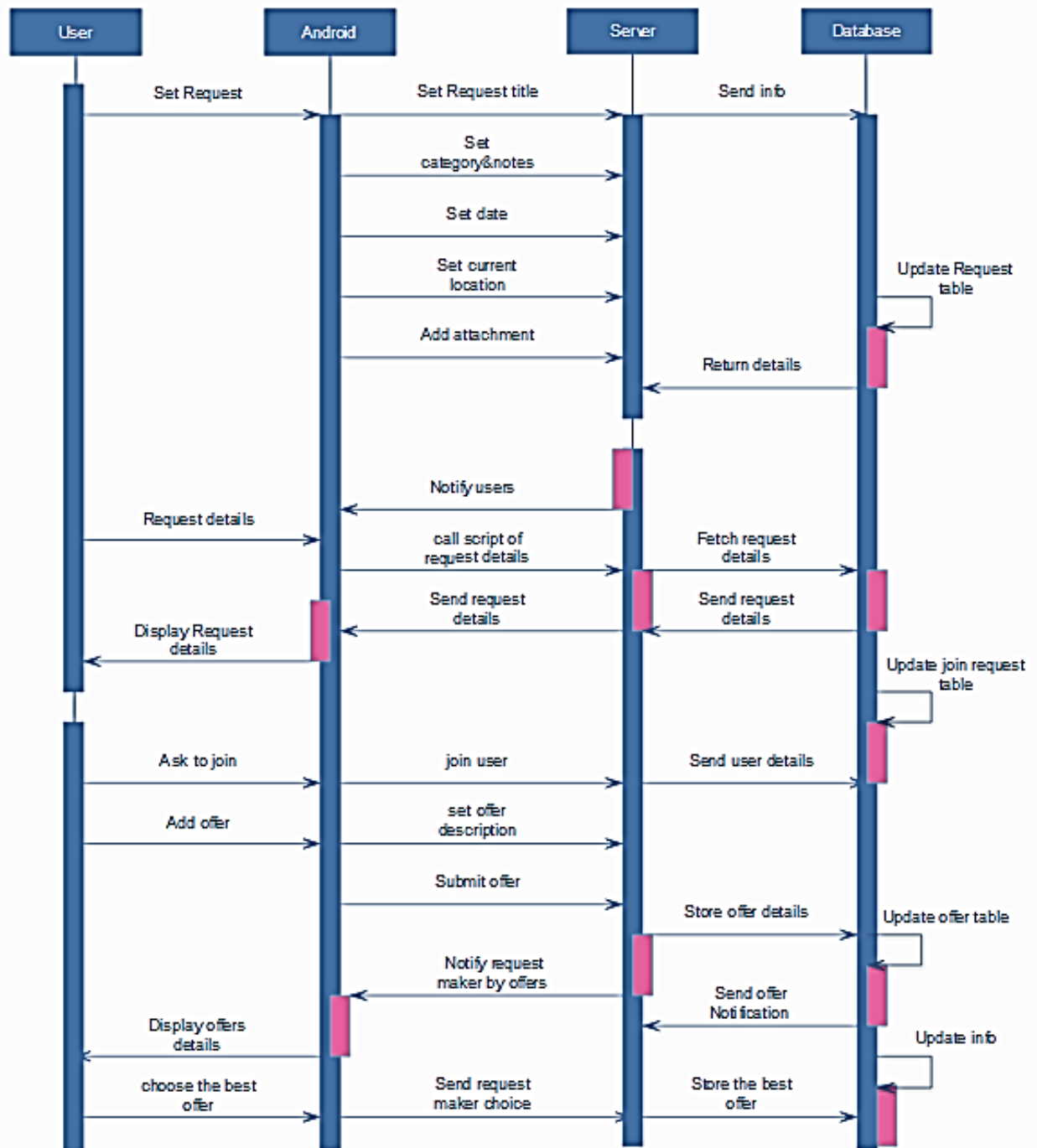


Figure 5: Request Sequence Diagram

2.2. System requirement

We analysis the requirement of the proposed system show in the following figure:

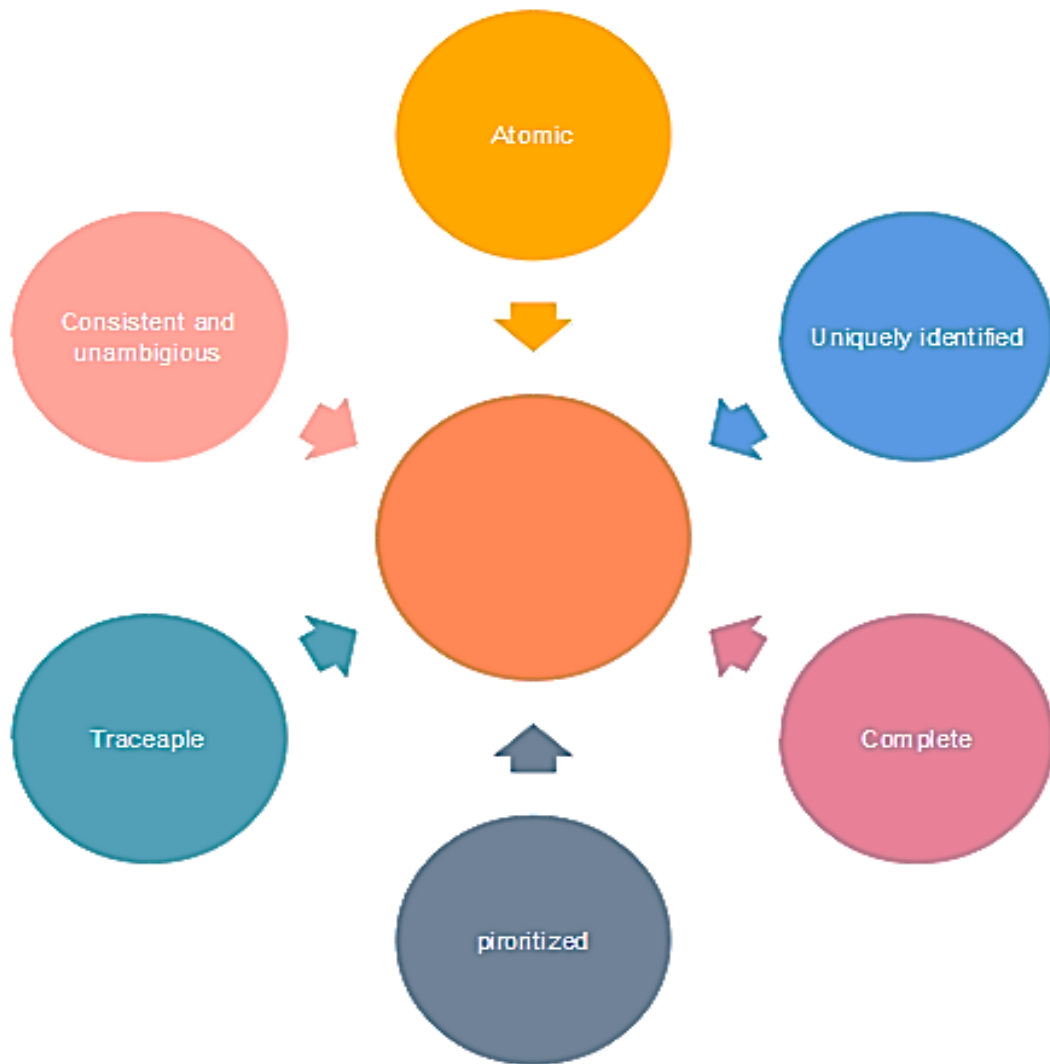


Figure 6: Requirement Analysis Factor

2.2.1. Requirement analysis

Following the system analysis approach , the system functional , non-functional , data flow diagram , architecture design are provided in the section 2 .the diagrams depicting the requirements connected in the early stages of the system. See Table 1. Page :17 and 18

2.3. List of requirement

2.3.1. Functional requirements

If a consumer initiates a request

- A consumer logs in to the system
- A user or group of users initiates a request in certain product or service.
- Based on the location, new users of the same interest join the request.
- Again based on the location, producer of the product or service of interest are notified of the number of the group and then start to send their offers.
- User accepts the best offer.

If a producer initiates an offer

- A producer logs in to the system.
- A producer makes an offer.
- A user or group of users in the same region will notify by the system with offer.
- Consumer rates how much the offer is true.

2.3.2. Non-Functional requirements

2.3.2.1. Operational

- The system is designed to have a web-based and mobile-based so the system will run on any Web browser and also downloaded from play store in android mobile

- In the event of a failure during update state the user will be able to retry connection.

2.3.2.2. Performance

A high speed for displaying the available offers. The system should be available for use 24 hours per day, 365 days per year.

Table 1: list the analysis factors for a single requirement

Atomic	1- A user logs in a system	1.1-Users enter phone number and password.
	2- User initiates request	2.1- if a user is consumer his request will send users to all nearby (consumers and producers) consumer can join to his request and producer can Send his offer.
		2.2-if a user is producer his offer will notify all nearby users and just they can do join or forward to other users

Complete	<ol style="list-style-type: none"> 1. User will log into the system by providing his phone, password. 2. System display message to the User “Turn on GPS”.
Consistent and unambiguous	<ol style="list-style-type: none"> 1. Offers & consumers’ asks in the request in different taps
Traceable	<ol style="list-style-type: none"> 1. The data of the user’s request for summary will be mapped to the corresponding architectural and design requirement
Prioritized	<ol style="list-style-type: none"> 1. User will register and select his interest 2. User will be able to initiate request if he is a consumer or making offer if he is a producer 3. Consumer will join to request or offer if he interests this offer 4. Consumer who initiates the request can select the best offer based on selected criteria (quality or price or type). 5. Consumer rates how much the offer is true

2.4. System analysis

2.4.1. Use Case

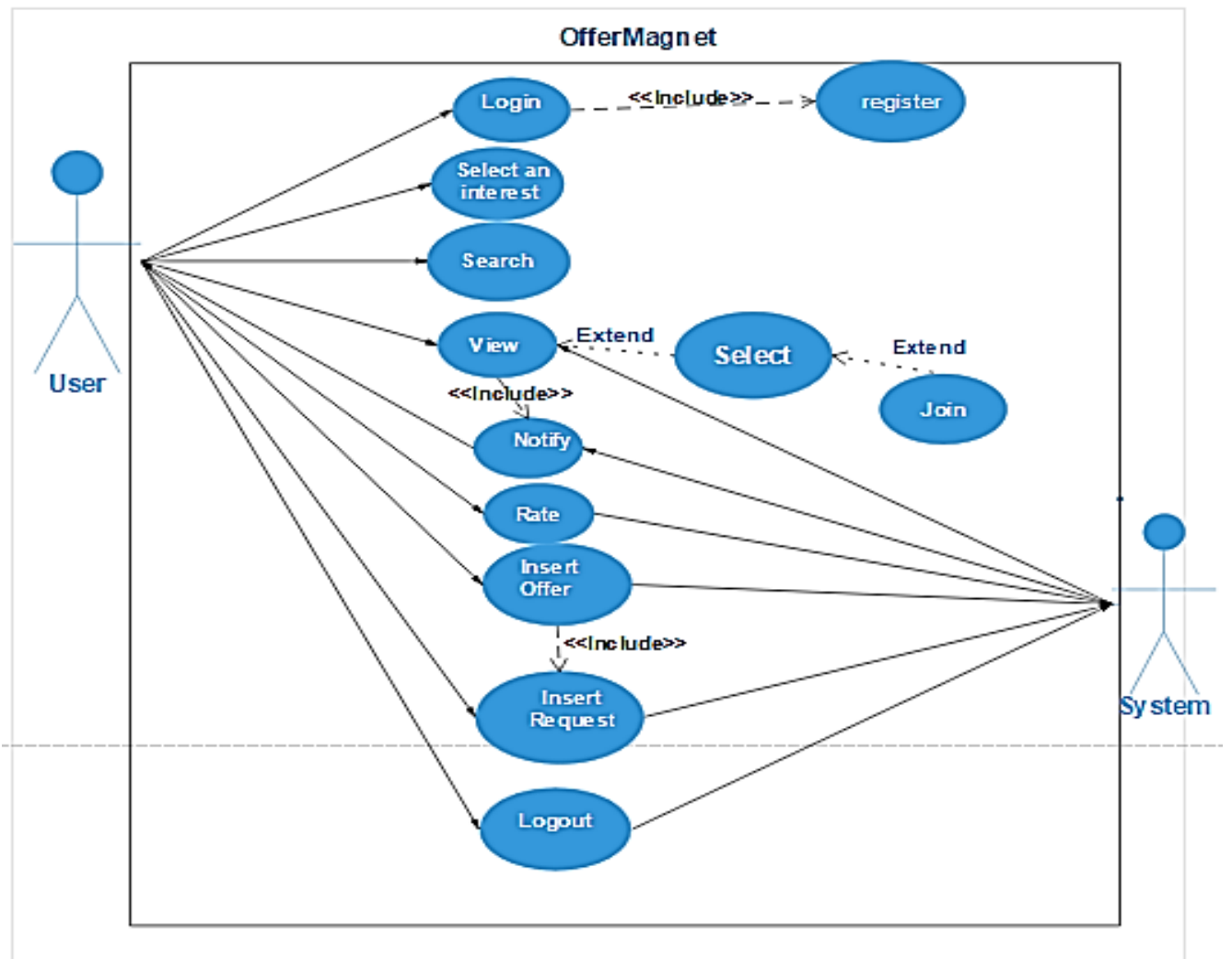


Figure 7: Use Case Diagram

2.4.2. Context Flow Diagram

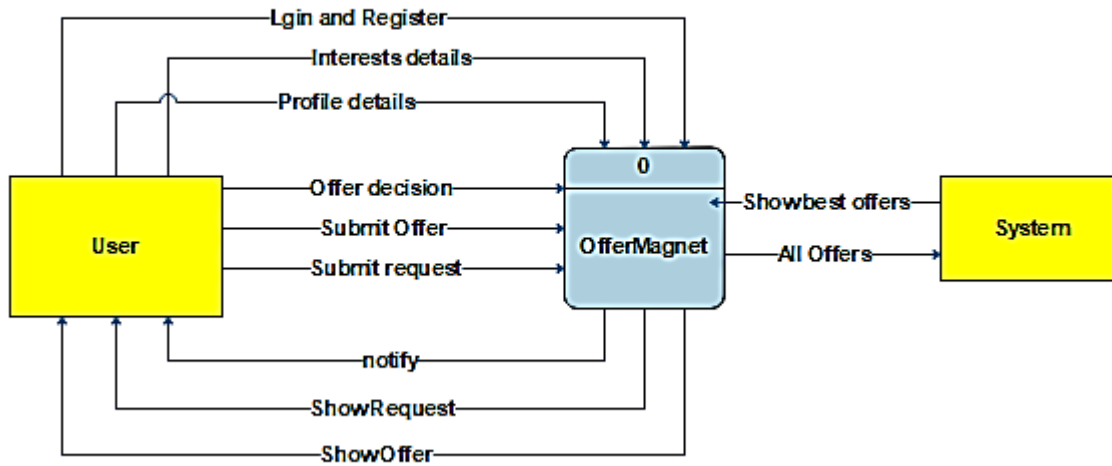


Figure 8: Context Flow Diagram

2.4.3. DFD Level 0

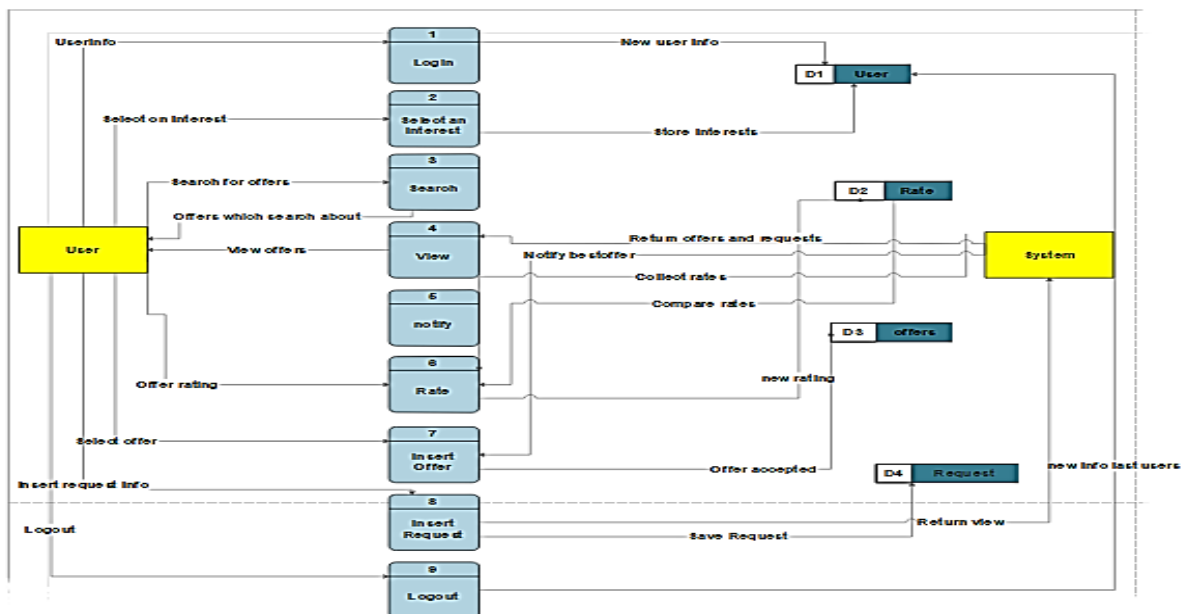


Figure 9: DFD Level 0

2.5. Policy Requirements

Users must accept these policies:

- OfferMagnet is not available to persons whose membership has been suspended or blocked by the system according to their behavior.
- The membership of offerMagnet is not available to persons under the age of 18 years.
- No person may register as a member of the system more than once.
- The system has the right to block or withdraw a person who provide any information that is untrue, inaccurate, not current or incomplete for customers.
- As far as the law allows, the user is solely responsible for his own request or offer posted on, he can update or delete its own requests.

On becoming a member of the system, a member must agree:

- He is responsible for maintaining the confidentiality of, and restricting access to and use of, his account and password, and accept responsibility for all activities that occur under his account and password. He must agree to immediately notify offerMagnet of any unauthorized use of his password or account or any other breach of security.
- He can contact with the providers by joining to the request and comment on it.
- Providers must specify the price of the items that offer for.

2.6. Training Requirements

- Integrate github in the development for source versioning and other issues.
- Applied machine learning with python.
- Mobile application
- Web application

2.7. General System Requirements:

2.7.1. Major System Capabilities

- System responds to user request.
- System can be considered as an intermediary matcher just connect producer to consumer.
- System can notify users in the same location in both case (offer or request).
- System must be available on the Internet.
- System must be available 24 hours per day.
- System must be accessible by mobile devices.

2.7.2. Major System Conditions

- AI & machine learning to be able to display new Offers.
- System must use GIS Framework for location.
- Server for storing data & making system available for 24 hours per day.

2.7.3. System Interfaces

- GIS framework for location.

2.7.4. System User Characteristics

- Consumer is a person who initiate a request & browses the service.
- Provider is a person who make an offer or provide a service.

2.8. General System Requirements

Specify the general system acceptance criteria specified and agreed upon by the project sponsor and key stakeholders that will be used to accept the final end product. For example:

- Historical requests for a user must be existed in his profile.
- Offers must be checked by system to make sure that they are true.
- New system must run in parallel with current production system for 2 months if any migration occurs.
- System will detect any fake account or user and block him.
- Fast response to the user.

3. External Design Documentation

3.1. Introduction

To the best of our knowledge, our core idea is context aware on user location. Display for them offers depending on their interests and the similarity among users (this part is achieved machine learning). We help users of the same geographic area with a common interest in certain product or services manage to resist unfair prices or unexpected quality of the provided services instead of starting an offer from the producers, we initiating it form consumers. The offer is defined as follows. A user of a group of users initiate a request for certain product or services, based on their locations, new users join them, while producers try to attract this group of users through offers. Finally, the best offer is picked-up by the consumer.

3.2. User Interface Design

In our user interface design

1. Splash screens for identifying the idea
2. Login page (log in in the system by phone and password) and sign up page (entering full info about user)
3. Navigation menu
 - profile
 - Home
 - Wishlist
 - MyOrder
 - About us
 - Privacy
 - Log out
4. Home page for discovering and search for offers and requests
5. Our Main processes :
 - Submit Offer.
 - Submit Request.

Usage Scenario

- In this part we will describe how the app will be work, for example the **request** scenario will be as following these steps:
 1. The user opens the app, Splash screen will appear after that he registers to the system and must full this information or if he already has an account, just log in to the system with this information (Phone, Password)
 2. User can search for offers for a specific item or can initiate a request, once he initiates a request, a request will send to all users in the same region. Those users can join to a request, Producers can view his offer by viewing the notification and can initiate offers for this request.
 3. Then the Initiator consumer accepts the best offer for the group, other joined users will be notified by the best offer.

- the **offer** scenario will be as following these steps:
 1. The user opens the app, Splash screen will appear after that he registers to the system and must full this information or if he already has an account, just log in to the system with this information (Phone, Password)
 2. User can search for offers for a specific item or can initiate an offer, once he initiates offer, a notification will send to all users in the same region. Those users can like and rate for this offer.

3.2.1. OfferMagnet Website

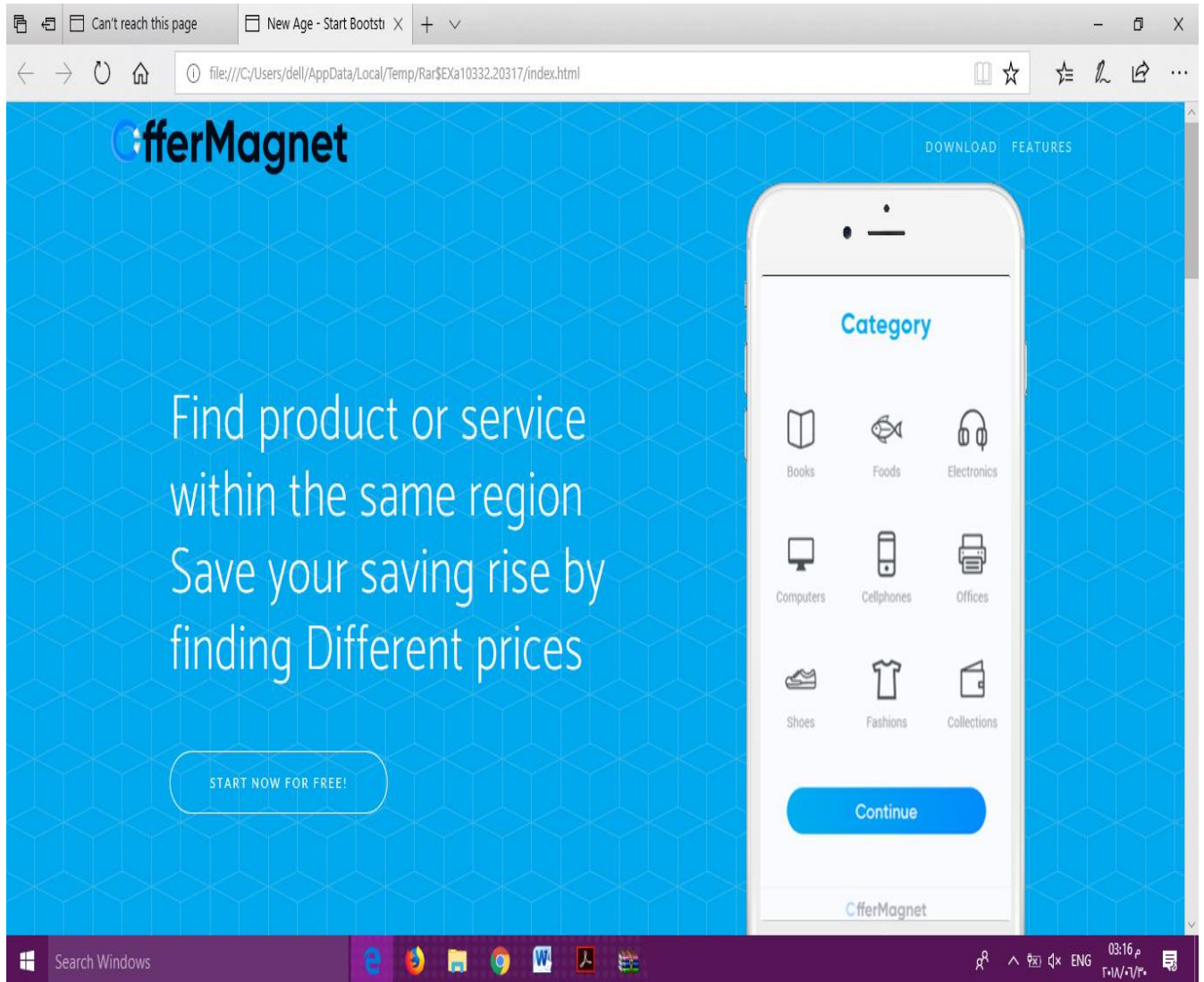


Figure 10: Website Interface 1

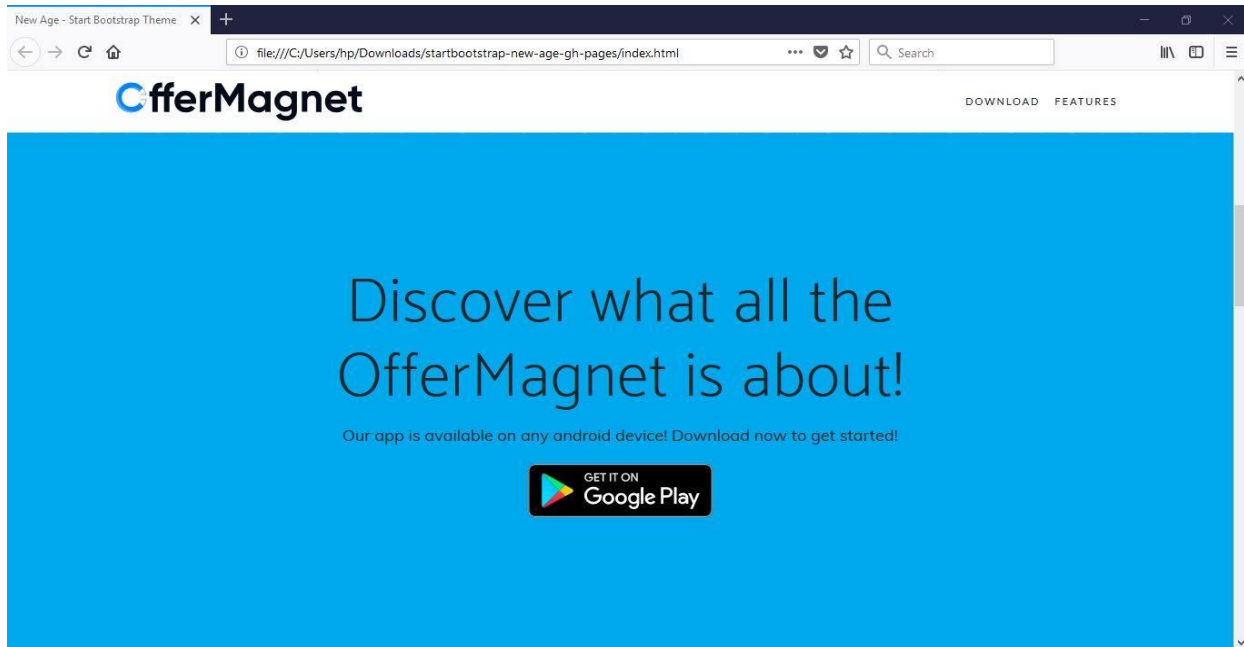


Figure 11: Website Interface 2

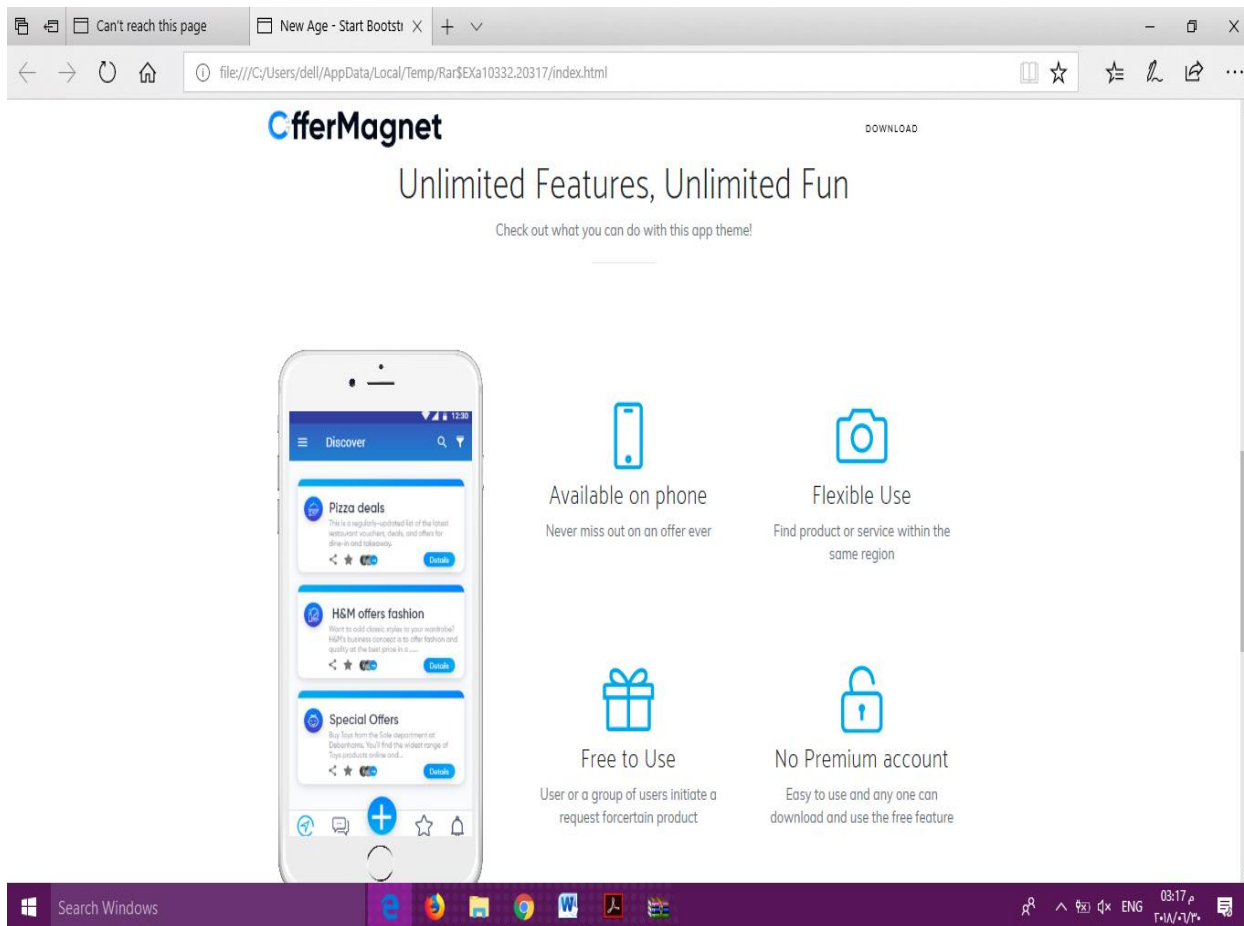


Figure 12: website Interface 3

3.2.2. Android UI

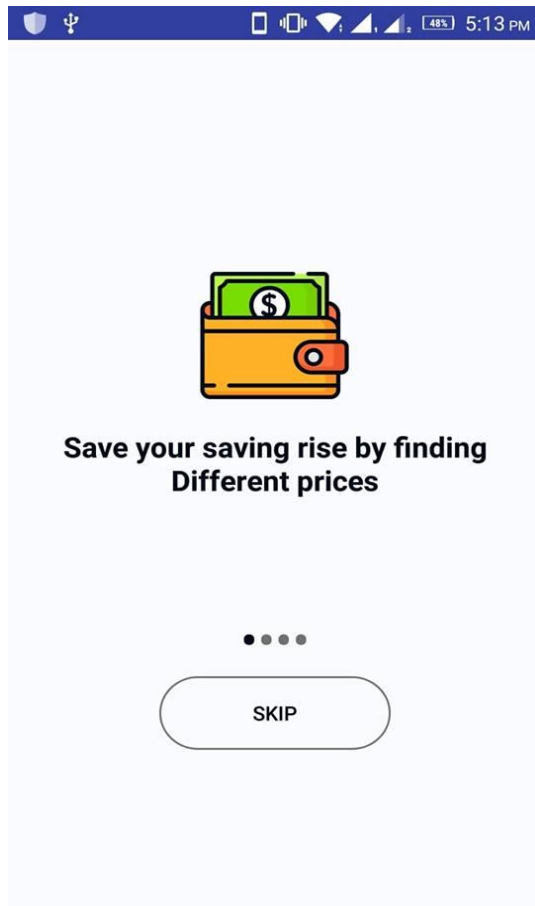


Figure 13 : Splash 1

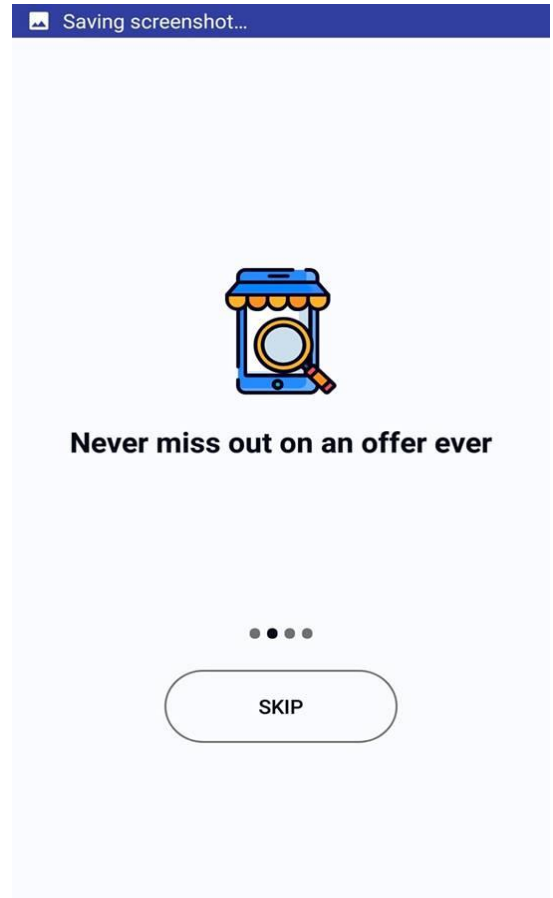


Figure 14 : Splash 2

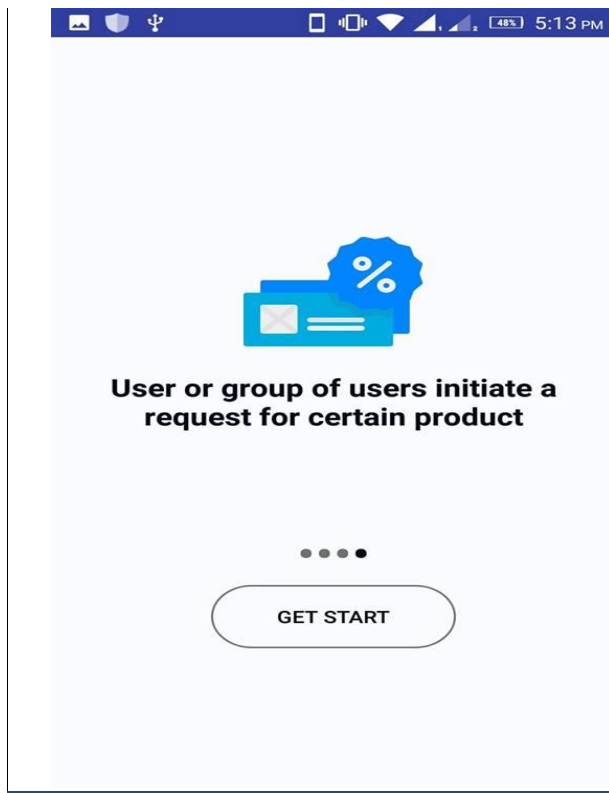


Figure 15 : Splash 3

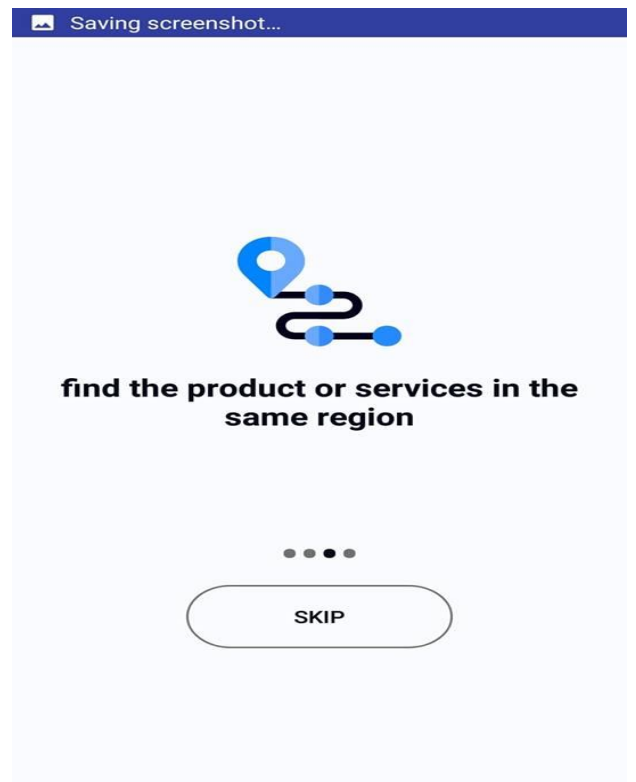
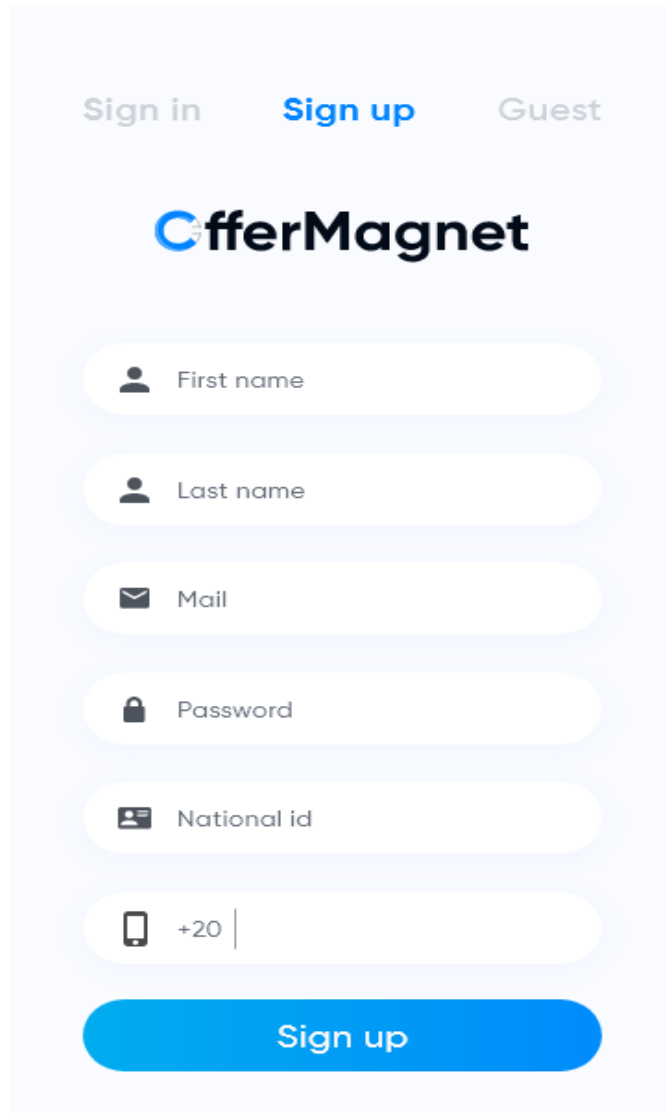


Figure 16 : Splash 4

1) Splash Screen

User can explore the story of the application and how to use it.



Sign in **Sign up** Guest

OfferMagnet

First name

Last name

Mail

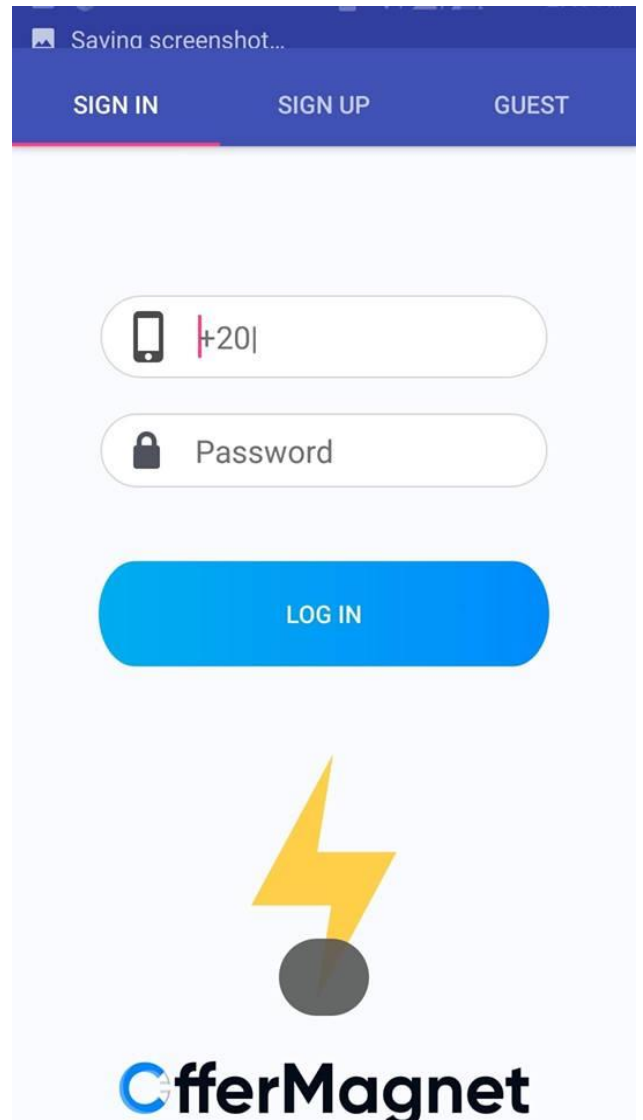
Password

National id

+20 |

Sign up

Figure 18: Sign Up




Saving screenshot...

SIGN IN SIGN UP GUEST

+20 |

Password

LOG IN



OfferMagnet

Figure 17: Log In

2) Log in & Sign Up

User can log in and sign up easel as well he could continue in guest option

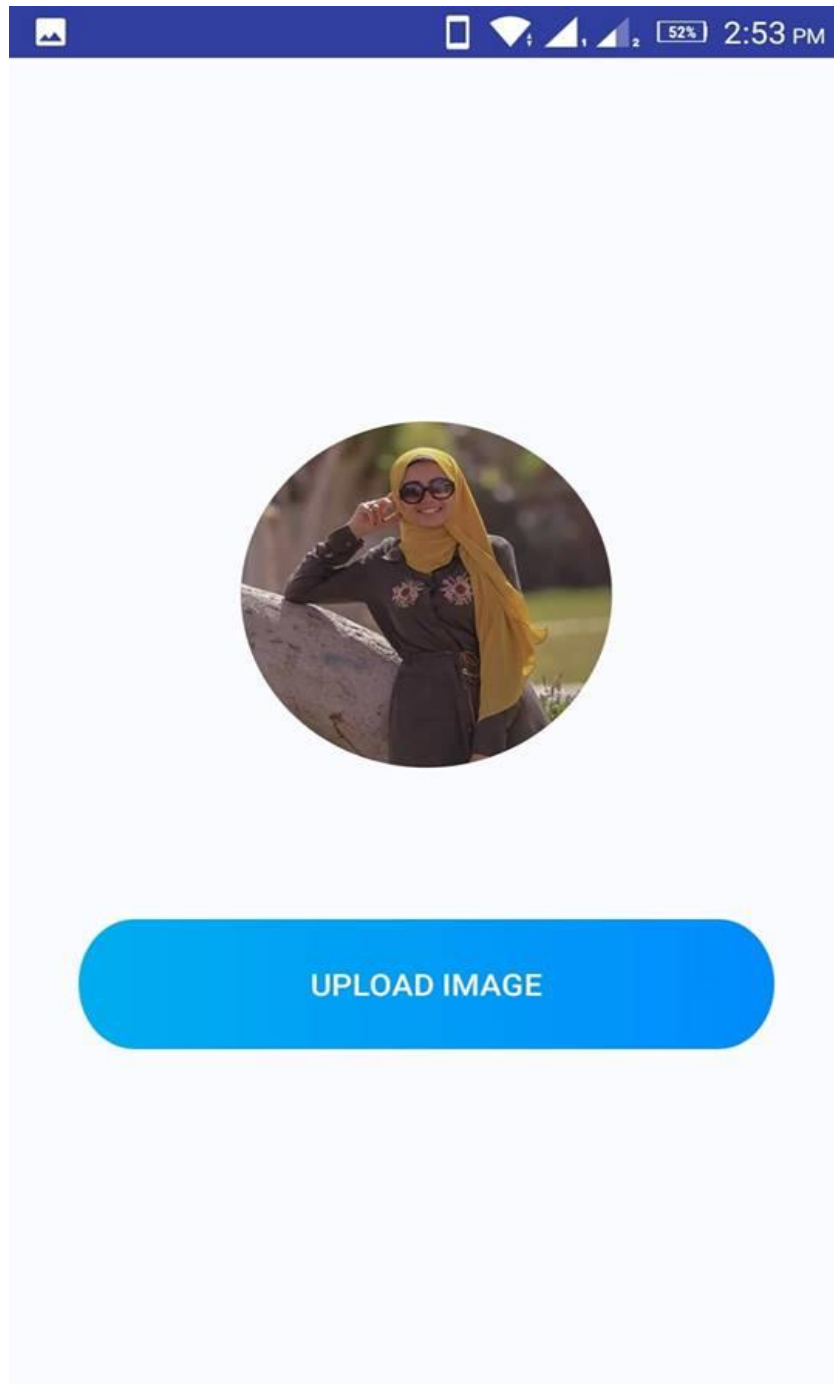


Figure 19: upload image

3) User can upload his image

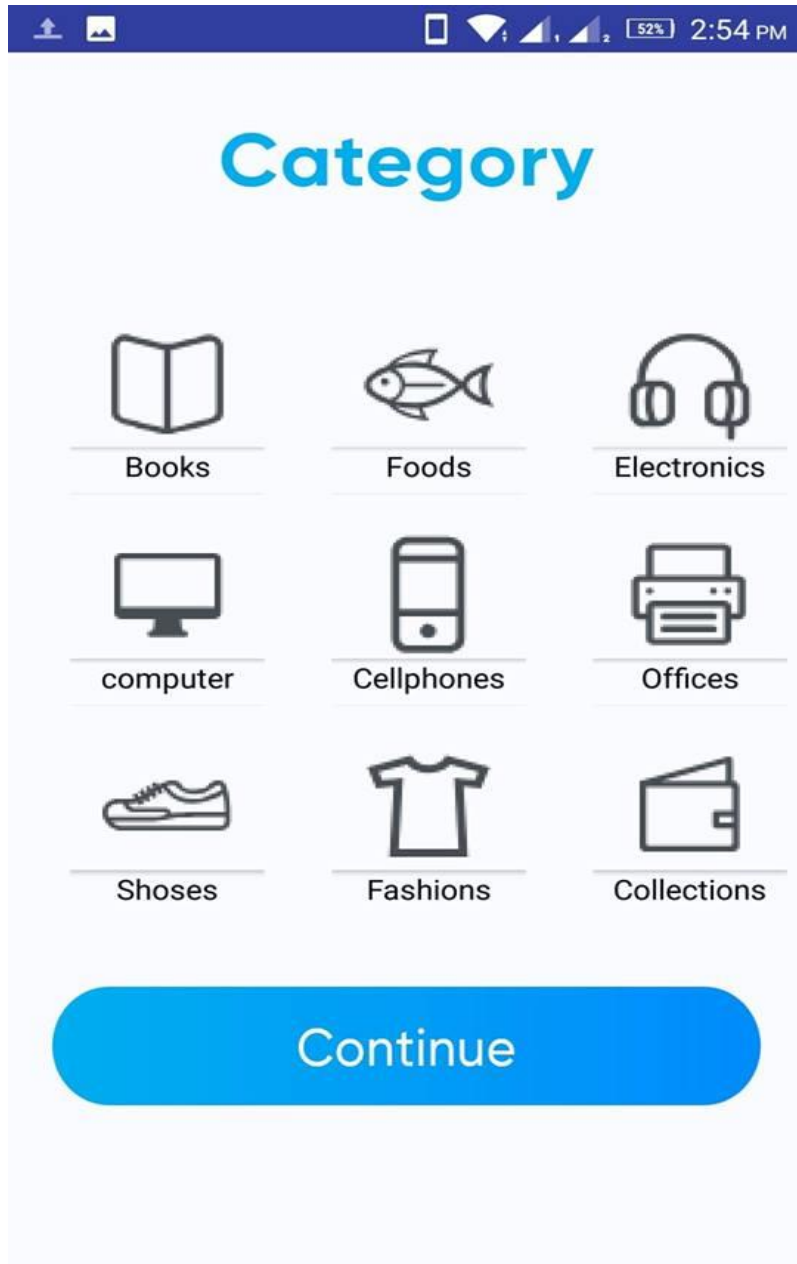


Figure 20: Selecting a Category

4) Category

The only user who logged in the application or signs up can choose between his interests to the suggestion will be form the category he chooses.

5) Offer & Request Tabs

Two Tabs used to show requests or offers.

Floating Button used to make requests and offers .

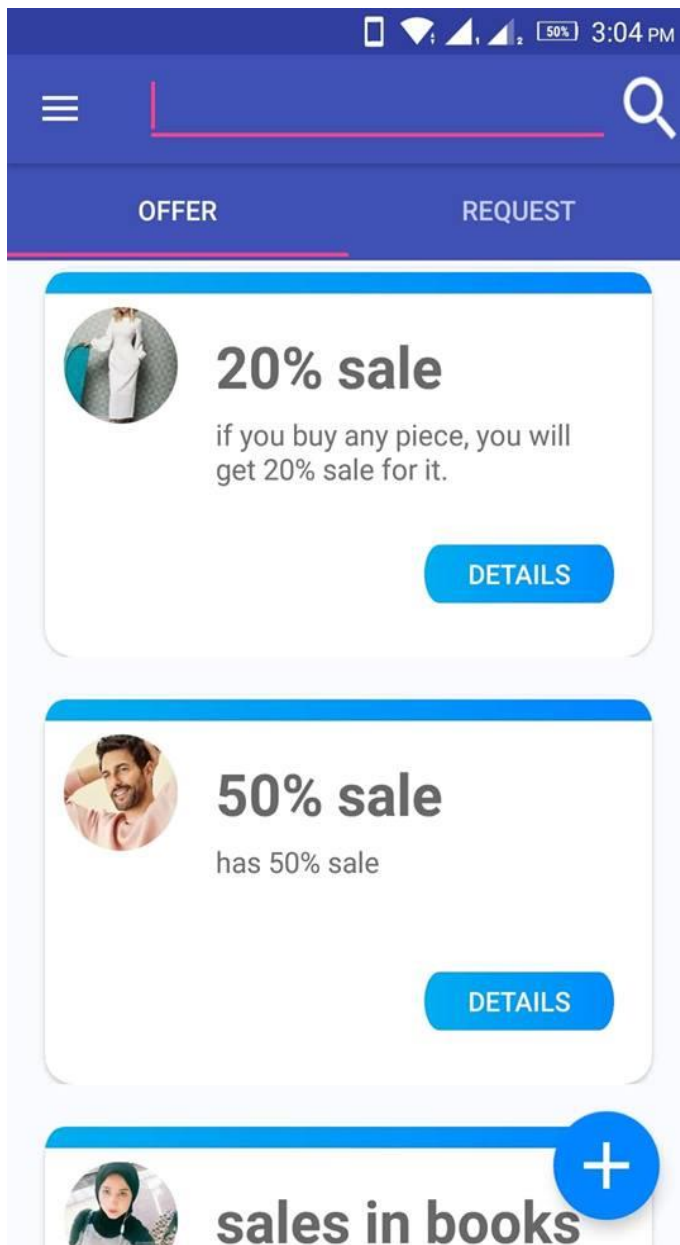


Figure 22: Offer Tab

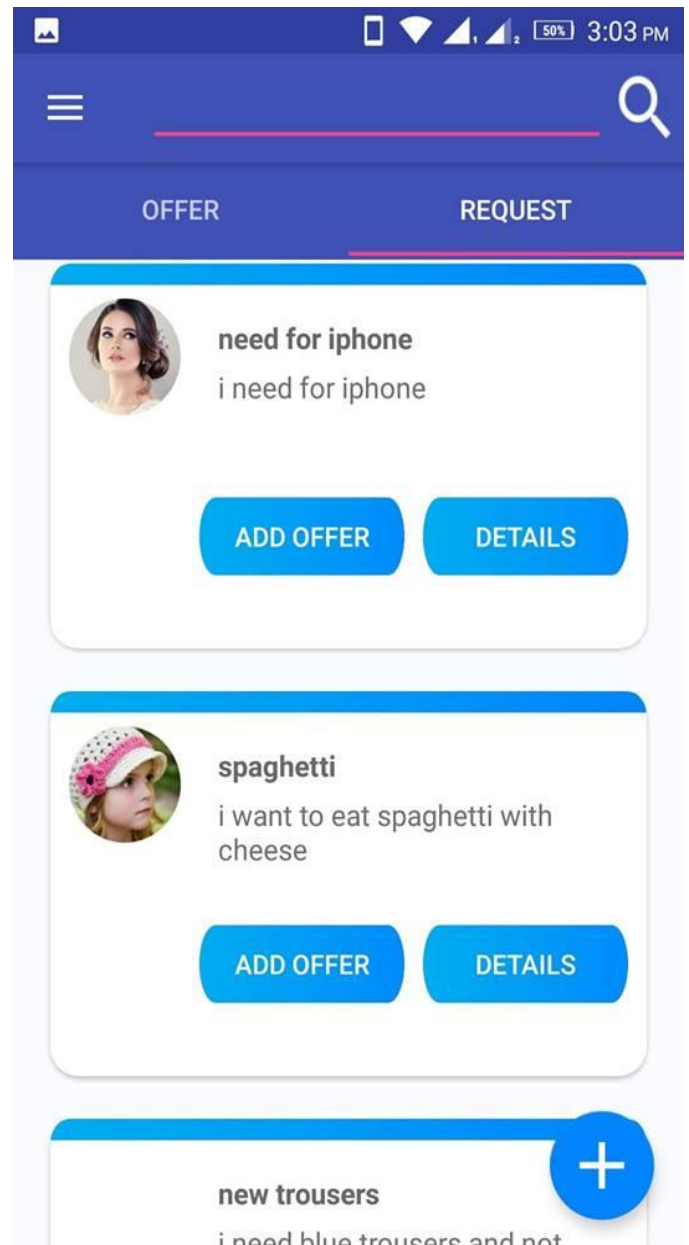


Figure 21: Request Tab

6) Alert Dialog for request description

It allows the user view details of request and join to it by click on join button

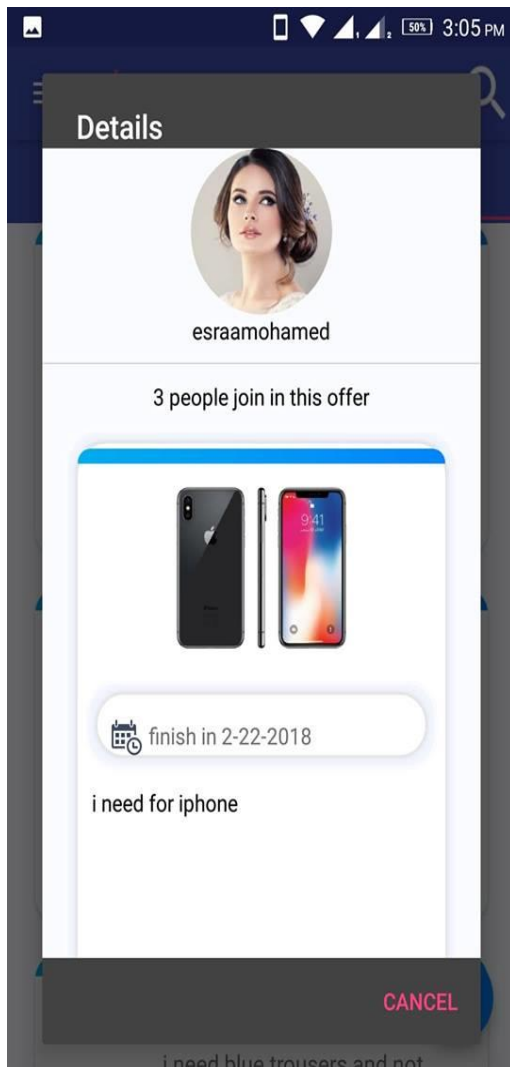


Figure 23 : Alert Dialog For Request

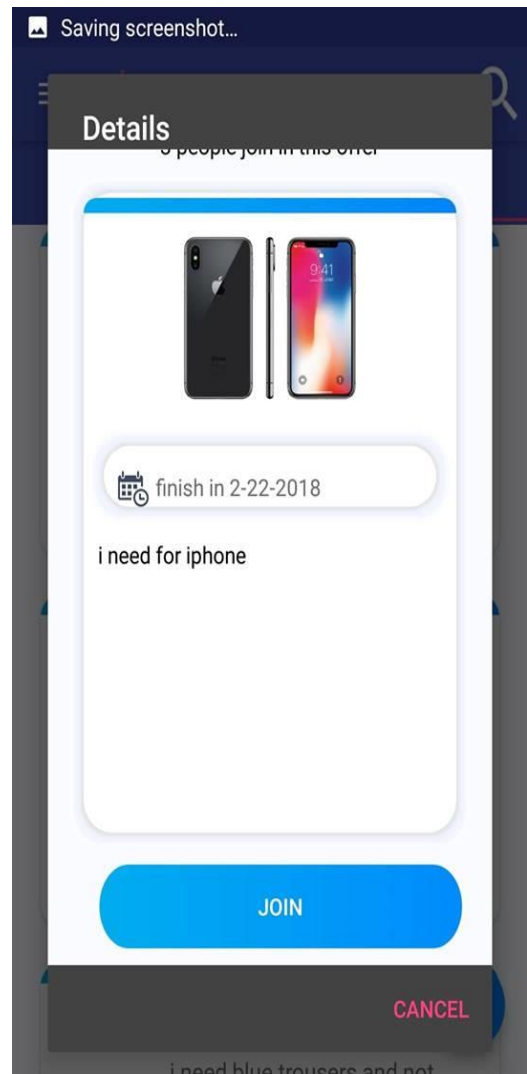
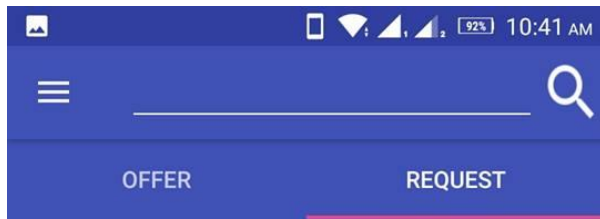


Figure 24 : Alert Dialog For Request

- 7) To insert a request, a user must insert all data required to create that request so that all users in the same location can be notified by it. These data may include, the request's title, category, city, attachment and the date of the request's availability.



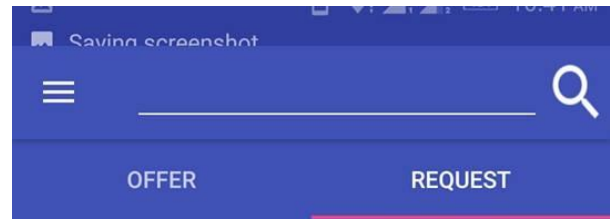
Request Title

Category

Note

city

Figure 25: Request Creation 1



Select Location

Current Location

15 Kilo Way

Select Validate Date

Set date

Date Text

Add attachment

Figure 26: Request Creation 2

8) Alert Dialog for offer's description

It allows to the user to see offers details and can

- Like the offer
- share offer by any social media app
- Contact with producer
- Rate the offer

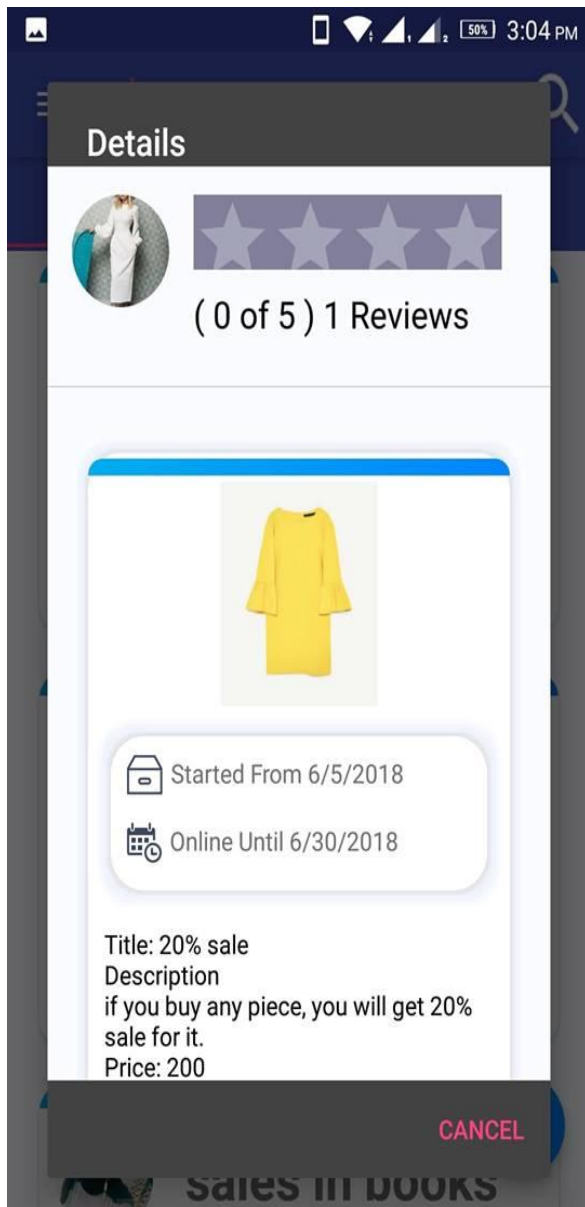


Figure 27 : Offer Detials 1

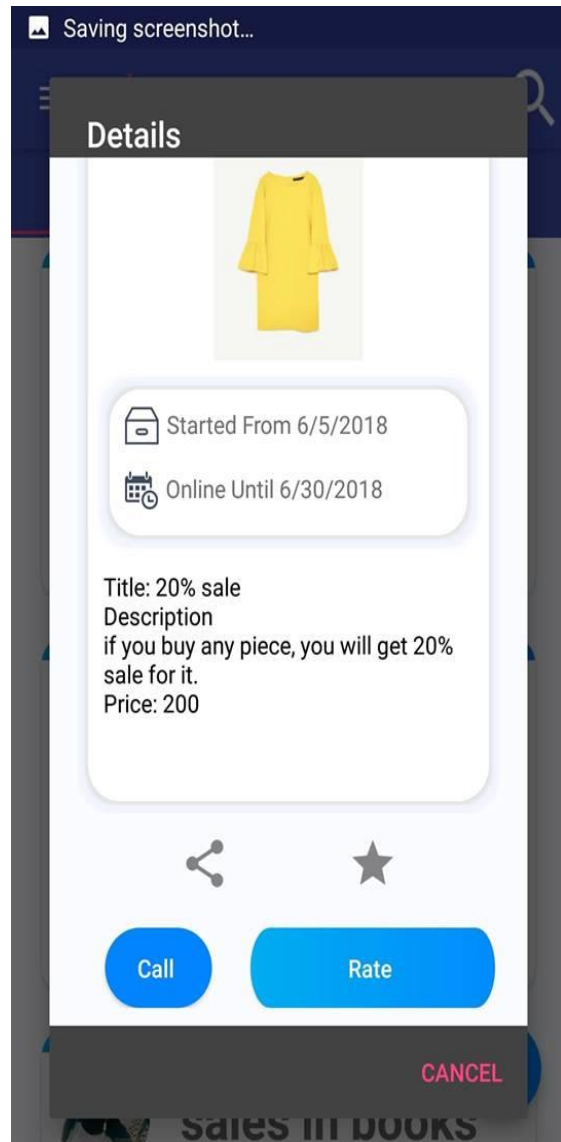


Figure 28 : Offer Detials 2

Offer Title

price

city

Note

Category

Figure 30: Offer Creation Step 1

Category

product description

Add attachment

Figure 29: Offer Creation Step 2

Mobile app interface for Offer Creation Step 3. The screen displays a map with a red dot for 'Current Location' and a green dot for '15 Kilo Way'. Below the map, there are two date selection fields: 'From' with a 'SET DATE' button and 'To' with a 'SET DATE' button. A large blue 'Submit offer' button is at the bottom.

Figure 31: Offer Creation Step 3

- 9) To insert an offer, a user must insert all data required to create an offer so that all users in the same location can be notified by it. These data include, the offer's title, category, city, attachment and the date of the offer's availability.

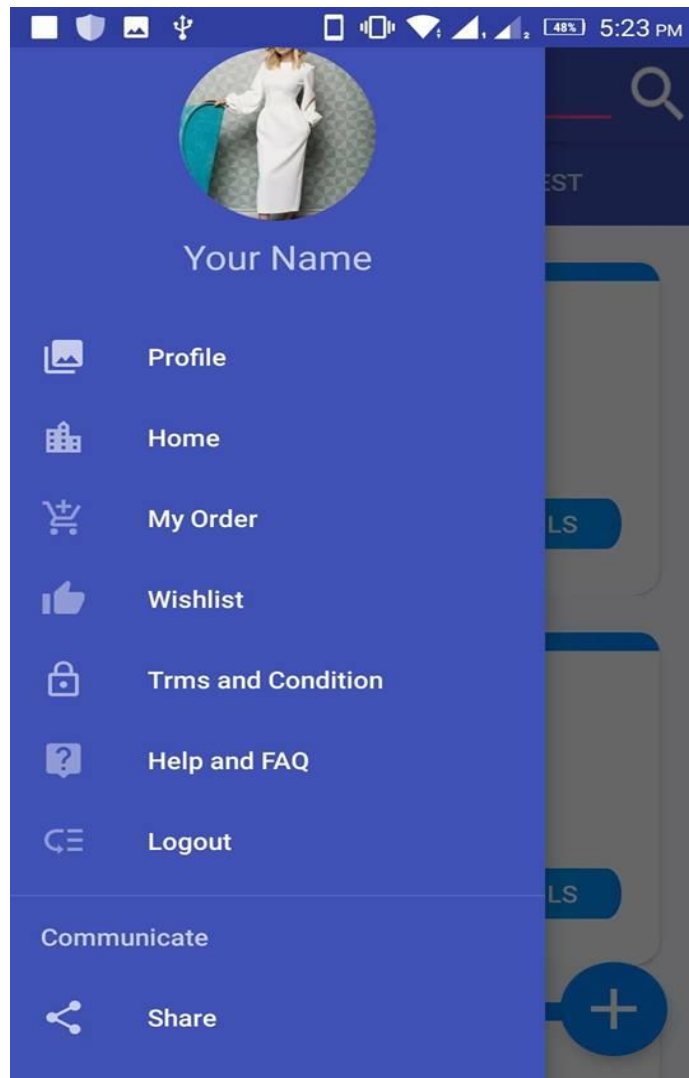


Figure 32 : Navigation Bar for menu

10) Navigation Bar

It contains

- 1- Profile and it allos the user to update his information
- 2- Wishlist :it contatins offers liked by him
- 3- MyOrder : it contains his requests
- 4- Home (discover page)
- 5- Help
- 6- Terms and condition
- 7- Log out

- 11) A user can turn on the location feature to discover offers and deals within the area he is in.

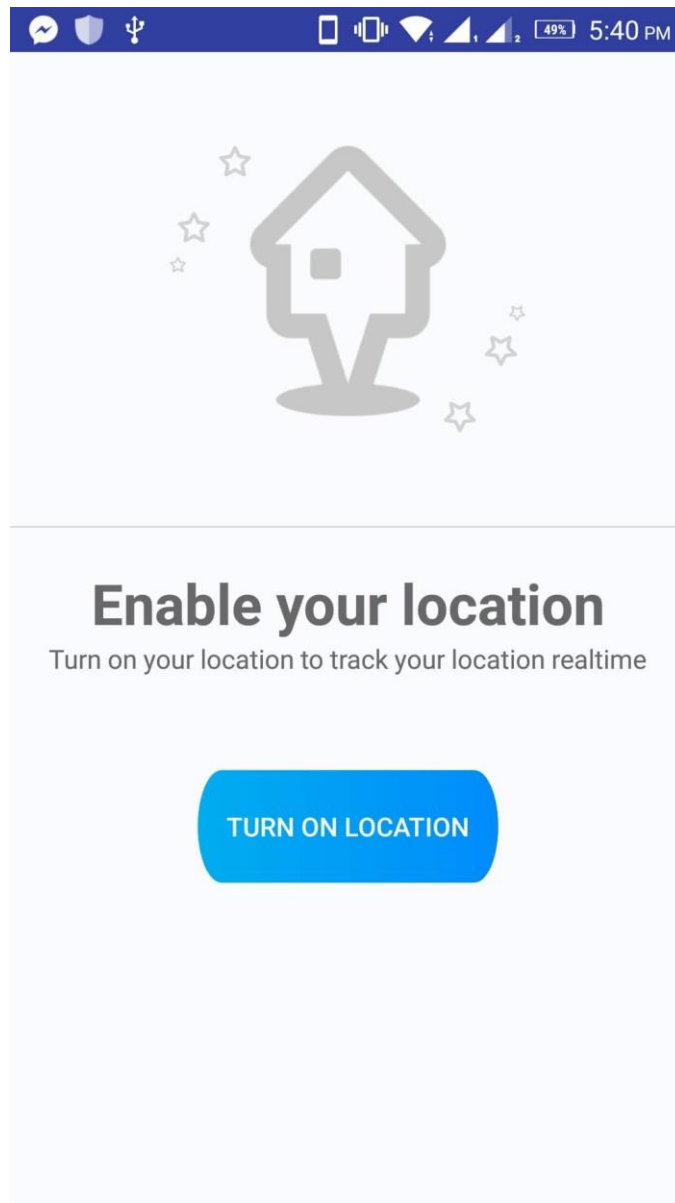


Figure 33: Location Feature

4. Implementation

4.1. Project plane

4.1.1. Schedule

- This consist of the following
- Defining the project scope
- Preforming the system design
- Learning required course
- Develop the entire software except the ML code
- Coding ML
- Writing the documentation

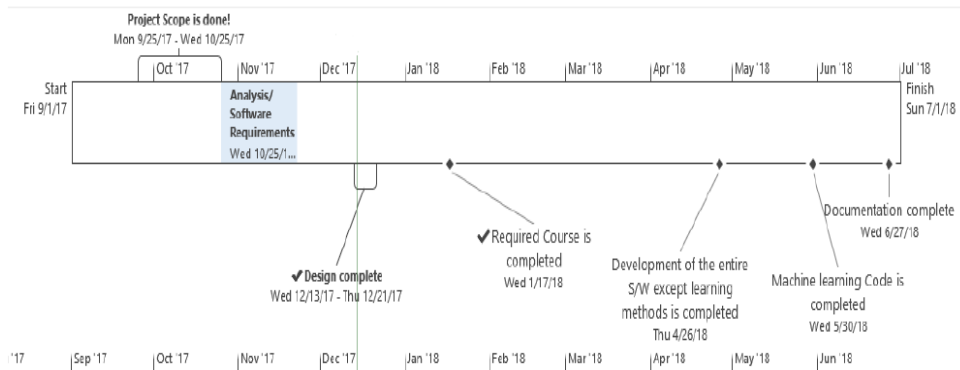


Figure 34 : Schedule

4.2. Hardware, software, tools and languages

They can list as the following:

1. You must have Android Mobile and Laptop.
2. App is based on three fields :
 - Mobile apps.
 - Web application.
 - Machine learning.
3. For the mobile application must be familiar with these tools (Java, Android, Android Studio).
4. For the web application must be familiar with these tools (PHP, HTML , CSS JAVASCRIPT, brackets editor).
5. For the ML must be familiar with these tools (Anaconda platform, Python programming language).
6. We use php programming for web services , MySQL database (php my admin) and “https://ooowebhost.com” as server

4.3. Android App Code

Here, you will show the screens and their code for the whole system

1. splash screen
2. login and sign up
3. upload image and select category
4. location
5. home (request and offers)
6. submit offer
7. submit Request
8. Navigation menu

1. Splash Screens

1.1. XML code

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/BackgroudColor"
    tools:context="com.example.hp.offeromagnet.Splash1">

    <android.support.v4.view.ViewPager
        android:id="@+id/viewPagerSlider"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="14dp"
        android:background="@color/BackgroudColor">

    </android.support.v4.view.ViewPager>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:id="@+id/dots"
        android:layout_marginTop="400dp"
        android:orientation="horizontal"
        android:layout_centerHorizontal="true"
        android:background="@color/BackgroudColor"

        >

    </LinearLayout>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/dots"
        android:layout_marginBottom="30dp"
        android:layout_centerHorizontal="true"
        android:background="@drawable/back_slider_button"
        android:id="@+id/butnSplash"
        android:textColor="@color/colorFocus"
        android:text="SKIP"/>

</RelativeLayout>
```

1.2. Java Code

```
package com.example.hp.offer magnet;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.Html;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

public class Splash1 extends AppCompatActivity {
    private ViewPager mSliderViewPager;
    private LinearLayout maDotsLayout;
    private slideAdapter slideAdapter1;
    private TextView[] mDots;
    private Button bSlider;
    private int mcurrent;

    SharedPreferences appPreferences;

    Boolean logged=false;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash1);
        mSliderViewPager =(ViewPager) findViewById(R.id.viewPagerSlider);
        maDotsLayout=(LinearLayout)findViewById(R.id.dots);
        bSlider=(Button)findViewById(R.id.butnSplash);
        appPreferences= Splash1.this.getSharedPreferences("AppPreferences",
Context.MODE_PRIVATE);
        logged=appPreferences.getBoolean("logged",false);
        if(logged){
            Intent intent = new Intent(Splash1.this, NavDrawer.class);
            startActivity(intent);
        }

        bSlider.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(Splash1.this ,UserTab.class);
                startActivity(intent);
            }
        });
        slideAdapter1=new slideAdapter(this);
        mSliderViewPager.setAdapter(slideAdapter1);
```

```

addDotsIndicator(0);
mSliderViewPager.addOnPageChangeListener(viewListener);
}
public void addDotsIndicator(int position){
    mDots=new TextView[4];

maDotsLayout.removeAllViews();
    for (int i=0;i<mDots.length;i++){
        mDots[i]=new TextView(this);
        mDots[i].setText(Html.fromHtml("&#8226;"));
        mDots[i].setTextSize(35);
        mDots[i].setTextColor(getResources().getColor(R.color.colorGray));
        maDotsLayout.addView(mDots[i]);

    }
    if(mDots.length>0){
        mDots[position].setTextColor(getResources().getColor(R.color.colorFocus));
    }
}
ViewPager.OnPageChangeListener viewListener=new
ViewPager.OnPageChangeListener() {
    @Override
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {

    }

    @Override
    public void onPageSelected(int position) {
addDotsIndicator(position);

        mcurrent=position;
        if(position==mDots.length-1){

            bSlider.setText("Get start");
        }
        else {

            bSlider.setText("SKIP");
        }
    }

    @Override
    public void onPageScrollStateChanged(int state) {
    }
}
};}

```

2. User Tab

XML code

```
<?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:id="@+id/main_content"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.example.hp.offer magnet.UserTab">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/appbar_padding_top"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:layout_weight="1"
            android:background="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|enterAlways"
            app:popupTheme="@style/AppTheme.PopupOverlay"
            app:title="@string/app_name">

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

        <android.support.design.widget.TabLayout
            android:id="@+id/tabs"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <android.support.design.widget.TabItem
                android:id="@+id/tabItem"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/tab_text_1" />

            <android.support.design.widget.TabItem
                android:id="@+id/tabItem2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="@string/tab_text_2" />

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

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

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

```

        <android.support.design.widget.TabItem
            android:id="@+id/tabItem3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tab_text_3" />

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

<android.support.v4.view.ViewPager
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior" />

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

```

Java code

```

<?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:id="@+id/main_content"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:context="com.example.hp.offermagnet.UserTab">

    <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/appbar_padding_top"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:layout_weight="1"
            android:background="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|enterAlways"
            app:popupTheme="@style/AppTheme.PopupOverlay"
            app:title="@string/app_name">

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

        <android.support.design.widget.TabLayout

```

```

        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <android.support.design.widget.TabItem
            android:id="@+id/tabItem"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tab_text_1" />

        <android.support.design.widget.TabItem
            android:id="@+id/tabItem2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tab_text_2" />

        <android.support.design.widget.TabItem
            android:id="@+id/tabItem3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/tab_text_3" />

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

<android.support.v4.view.ViewPager
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior" />

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

```

2.1. Login_tab

2.1.1. Log in xml Code

```

<ScrollView 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/constraintLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/BackgroudColor"
    tools:context="com.example.hp.offeromagnet.UserTab">

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

            android:fontFamily="Gilroy"
            android:id="@+id/phoneEditText"
            android:layout_width="match_parent"

```



```

        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="phone"
        android:layout_centerHorizontal="true"
        android:background="@drawable/shap_login_signup"
        android:layout_marginBottom="18dp"
        android:layout_marginTop="109dp"
        android:hint="+20|"
        android:textColor="#000"
        android:drawableLeft="@drawable/cellphoneiphone"
        android:paddingLeft="20dp"
        android:drawablePadding="@dimen/padding_drawable"
        android:paddingRight="20dp"
        android:paddingTop="10dp"
        android:maxLength="11"
        android:textColorHint="@color/colorGray"
    android:paddingBottom="10dp"
        android:layout_marginRight="50dp"
        android:layout_marginLeft="50dp"/>

```

<EditText

```

        android:fontFamily="Gilroy"
        android:drawablePadding="@dimen/padding_drawable"
        android:paddingRight="20dp"
        android:id="@+id/passEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:cursorVisible="true"
        android:hint="Password"
        android:ems="10"
        android:inputType="textPassword"
        android:textColorHint="@color/colorGray"
        android:background="@drawable/shap_login_signup"
        android:layout_marginBottom="21dp"
        android:textColor="@color/colorBlack"
        android:drawableLeft="@drawable/lock_reset"
        android:paddingLeft="20dp"
        android:layout_below="@id/phoneEditText"
        android:layout_centerHorizontal="true"
        android:maxLength="20"
        android:paddingTop="10dp"
        android:paddingBottom="10dp"
        android:layout_marginRight="50dp"
        android:layout_marginLeft="50dp"

```

/>

<Button

```

        android:id="@+id/btnlogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="@color/BackgroudColor"
        android:background="@drawable/sign_back_button"
        android:text="log in"
        android:layout_marginBottom="50dp"

```

```

        android:layout_below="@id/passEditText"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="50dp"
        android:layout_marginRight="50dp"
        android:layout_centerHorizontal="true"
    />
    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:src="@drawable/sign"
        android:layout_below="@id/btnlogin"
        android:layout_centerHorizontal="true" />

    <ImageView
        android:layout_below="@id/imageView3"
        android:id="@+id/imageView4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/logosignin"
        android:layout_marginBottom="33dp"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
    />
</RelativeLayout>

</ScrollView>

```

2.1.2. Login Java Code

```

package com.example.hp.offeromagnet;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

```

```

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.HashMap;
import java.util.Map;

public class LogIn extends Fragment {
    Button btn;
    EditText phone;
    EditText pass;
    String var;
    boolean phonev, passv;
    Database database;
    SharedPreferences.Editor prefEditor;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        final View rootView = inflater.inflate(R.layout.tab_signin, container, false);
        phone = (EditText) rootView.findViewById(R.id.phoneEditText);
        pass = (EditText) rootView.findViewById(R.id.passEditText);
        prefEditor = getActivity().getSharedPreferences("AppPreferences",
Context.MODE_PRIVATE).edit();

        phone.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
                if (phone.getText().length() <= 10) {
                    phone.setError("Required at least 11 number");
                    phonev = false;
                } else
                    phonev = true;
            }

            @Override
            public void afterTextChanged(Editable editable) {

            }
        });
        pass.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
                if (pass.getText().length() <= 6) {

```

```

        pass.setError("Required at least 7 digits");
        passv = false;
    } else
        passv = true;
    }

    @Override
    public void afterTextChanged(Editable editable) {

    }

});

btn = (Button) rootView.findViewById(R.id.btnlogin);
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (phonev && passv) {
            login();
            // Intent intent = new Intent(getApplicationContext(), NavDrawer.class);

            // startActivity(intent);
            Toast.makeText(getApplicationContext(), "sent",
Toast.LENGTH_LONG).show();

        }
    }
});
return rootView;

}

@Override
public void onStart() {
    super.onStart();
    database = new Database(getActivity());
}

private void login() {

    StringRequest postRequest = new StringRequest(Request.Method.POST,
"https://offer-system.000webhostapp.com/Login.php",

        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

                try {
                    JSONObject jsonObject = new JSONObject(response);
                    JSONArray jsonArray = jsonObject.getJSONArray("response");
                    Toast.makeText(getActivity(), response, Toast.LENGTH_SHORT).show();
                    for (int x = 0; x < jsonArray.length(); x++) {
                        JSONObject jsonObject1 = jsonArray.getJSONObject(x);
                        if (jsonObject1.getString("state").equals("OK")) {

```

```

        database.UpdateData("1",jsonObject1.getString("id"),
            jsonObject1.getString("user_name"),
            jsonObject1.getString("phone"),
            jsonObject1.getString("password"),
            jsonObject1.getString("city"),
            jsonObject1.getString("gender"),
            "1",
            jsonObject1.getString("birthdate"),
            jsonObject1.getString("profile_picture")

        );

        Intent intent = new Intent(getContext(), NavDrawer.class);
        prefEditor.putBoolean("logged",true);
        prefEditor.apply();
        startActivity(intent);

    } else if (jsonObject1.getString("state").equals("NO"))

        Toast.makeText(getContext().getApplicationContext(), "error",
Toast.LENGTH_LONG).show();
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

            if (error instanceof ServerError)
                Toast.makeText(getContext().getApplicationContext(), "Server Error",
Toast.LENGTH_SHORT).show();
            else if (error instanceof TimeoutError)
                Toast.makeText(getContext().getApplicationContext(), "Connection Timed
Out", Toast.LENGTH_SHORT).show();
            else if (error instanceof NetworkError)
                Toast.makeText(getContext().getApplicationContext(), "Bad Network
Connection", Toast.LENGTH_SHORT).show();
        }

    })

    {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            Map<String, String> param = new HashMap<String, String>();
            param.put("phone", phone.getText().toString());
            param.put("password", pass.getText().toString());
            return param;
        }
    }

```

```

    };
    Volley.newRequestQueue(getContext()).add(postRequest);
}
}

```

2.2. Signup_tab

2.2.1. Sign up XML Code

```

<ScrollView 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/constraintLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/BackgroudColor"
    tools:context="com.example.hp.offeromagnet.UserTab">

```

```

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

```

```

<ImageView
    android:layout_centerHorizontal="true"
    android:id="@+id/imageView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:layout_alignParentTop="true"

    android:layout_marginLeft="11dp"
    android:layout_marginStart="11dp"
    android:layout_marginTop="44dp"
    android:src="@drawable/logosignin" />

```

```

<EditText
    android:id="@+id/firstNme"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/imageView4"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginTop="28dp"
    android:background="@drawable/shap_login_signup"
    android:drawableLeft="@drawable/account"
    android:drawablePadding="@dimen/padding_drawable"

```

```

android:ems="10"
android:fontFamily="Gilroy"
android:hint="User Name"
android:inputType="text"
android:maxLength="15"
android:paddingBottom="11dp"
android:paddingLeft="20dp"
android:paddingRight="20dp"
android:paddingTop="11dp"
android:textColor="@color/colorBlack"
android:textColorHint="@color/colorGray" />

```

```

<EditText
    android:id="@+id/phone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/firstNme"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
    android:layout_marginTop="10dp"
    android:background="@drawable/shap_login_signup"
    android:drawableLeft="@drawable/cellphoneiphone"
    android:drawablePadding="@dimen/padding_drawable"
    android:ems="10"
    android:hint="+20|"
    android:inputType="phone"
    android:maxLength="11"
    android:paddingBottom="10dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:paddingTop="10dp"
    android:textColor="@color/colorBlack"
    android:textColorHint="@color/colorGray" />

```

```

<EditText
    android:id="@+id/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/phone"
    android:layout_marginTop="10dp"
    android:background="@drawable/shap_login_signup"
    android:drawableLeft="@drawable/lock_reset"
    android:drawablePadding="@dimen/padding_drawable"
    android:ems="10"
    android:fontFamily="Gilroy"
    android:hint="Password"
    android:inputType="textPassword"
    android:maxLength="20"
    android:paddingBottom="11dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:paddingTop="11dp"

```

```

        android:layout_marginRight="50dp"
        android:layout_marginLeft="50dp"
        android:textColor="@color/colorBlack"
        android:textColorHint="@color/colorGray" />

```

```

<EditText
    android:id="@+id/cpassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/password"
    android:layout_alignStart="@+id/password"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/password"
    android:layout_marginTop="10dp"
    android:background="@drawable/shap_login_signup"
    android:drawableLeft="@drawable/lock_reset"
    android:drawablePadding="@dimen/padding_drawable"
    android:ems="10"
    android:fontFamily="Gilroy"
    android:hint="Confirm password"
    android:inputType="textPassword"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:layout_marginRight="50dp"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:layout_centerHorizontal="true"
    android:textColor="@color/colorBlack"
    android:textColorHint="@color/colorGray" />

```

```

<Spinner
    android:entries="@array/cities"
    style="@style/spinner_style"
    android:pointerIcon="all_scroll"
    android:spinnerMode="dropdown"
    android:id="@+id/Capital"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/cpassword"
    android:layout_marginTop="15dp"
    android:drawablePadding="@dimen/padding_drawable"
    android:ems="10"
    android:paddingLeft="10dp"
    android:paddingRight="20dp"
    android:paddingBottom="10dp"
    android:paddingTop="10dp"
    android:layout_marginLeft="50dp"
    android:layout_marginRight="50dp"
/>

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```



```

android:id="@+id/age"
android:layout_below="@id/Capital"
android:layout_marginLeft="50dp"
android:layout_marginRight="50dp"
android:layout_marginTop="10dp">

```

<TextView

```

android:id="@+id/TextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/monthSpinner"
android:layout_alignBottom="@+id/monthSpinner"
android:layout_gravity="center_vertical"
android:paddingTop="5dp"
android:text="BirthDate:"
android:textColor="@color/colorGray"
android:textSize="18sp" />

```

<Button

```

android:id="@+id/btnSetDateOfBirth"
android:layout_width="wrap_content"
android:layout_height="30dp"
android:layout_gravity="center_vertical"
android:background="@drawable/button_shape"
android:layout_marginLeft="10dp"
android:hint="select"
android:textColorHint="@color/colorWhite"/>

```

<TextView

```

android:id="@+id/textDateOfBirth"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="20dp"
android:layout_marginTop="10dp"

android:textColor="@color/colorBlack"

```

/>

</LinearLayout>

<Spinner

```

android:id="@+id/gender"
style="@style/spinner_style"

android:entries="@array/gender"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_below="@id/age"
android:paddingBottom="10dp"
android:paddingTop="10dp"
android:layout_marginTop="10dp"
android:layout_marginRight="50dp"

```

```

        android:layout_marginLeft="50dp"
        android:pointerIcon="all_scroll"
        android:spinnerMode="dropdown" />
    <Button
        android:id="@+id/signup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/gender"
        android:layout_alignStart="@+id/gender"
        android:layout_below="@+id/gender"
        android:layout_marginBottom="30dp"
        android:layout_marginRight="50dp"
        android:layout_marginTop="27dp"
        android:background="@drawable/sign_back_button"
        android:ems="10"
        android:fontFamily="Gilroy"
        android:text="Sign Up"
        android:textColor="@color/BackgroudColor" />

</RelativeLayout>
</ScrollView>

```

2.2.2. Signup Java Code

```

package com.example.hp.offer magnet;

import android.app.DatePickerDialog;
import android.content.Intent;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.os.Message;
import android.support.v4.app.DialogFragment;
import android.support.v4.app.Fragment;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.ServerError;

```

```

import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class SignUp extends Fragment {
    TextView txtDate;
    Button btn,btnSetDate;
    EditText pass,fname,cpass,phone;;
    Spinner spnGinder, Capital;

    double a=0.0;
    boolean vn=false;
    boolean vphone=false;

    boolean pc=false;
    boolean vpc=false;
    int age;
    private DatePickerDialog.OnDateSetListener DatePicker1;
    User user=new User();

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

        final View rootView = inflater.inflate(R.layout.tab_signup, container, false);
        final Calendar today=Calendar.getInstance();
        pass=(EditText)rootView.findViewById(R.id.password);
        cpass=(EditText)rootView.findViewById(R.id.cpassword);
        phone=(EditText) rootView.findViewById(R.id.phone);
        fname=(EditText) rootView.findViewById(R.id.firstNme);
        spnGinder =(Spinner)rootView.findViewById(R.id.gender);
        btnSetDate=(Button)rootView.findViewById(R.id.btnSetDateOfBirth);
        txtDate =(TextView) rootView.findViewById(R.id.textDateOfBirth);
        Capital=(Spinner)rootView.findViewById(R.id.Capital);

        fname.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            }
        }
    }

```

```

@Override
public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    if(fname.getText().length()<=0)
    {
        fname.setError("please enter a valid name");
    }
    else {
        vn=true;
    }
}

@Override
public void afterTextChanged(Editable editable) {
}
});

phone.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
        if(phone.getText().length()==0)
        {
            phone.setError("Required");
        }
        else if(phone.getText().length()<=10){
            phone.setError("Required at least 11 number");
        }
        else
        if(!phone.getText().toString().substring(0,3).equals("010")&&!phone.getText().toString().substri
ng(0,3).equals("011")
&&!phone.getText().toString().substring(0,3).equals("012")&&!phone.getText().toString().substri
ng(0,3).equals("015")){
            phone.setError("invalid phone number");
        }
        else {
            vphone=true;
        }
    }
}

@Override
public void afterTextChanged(Editable editable) {
}
});
pass.addTextChangedListener(new TextWatcher() {
    @Override

```

```

    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

        if(pass.getText().length()==0)
        {
            pass.setError("Required");
        }
        else if(pass.getText().length()<8){
            pass.setError("required at least 8 character");
        }
        else
        {
            pc=true;
        }

    }

    @Override
    public void afterTextChanged(Editable editable) {

    }
});
cpass.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
        if(cpass.getText().length()==0)
        {
            cpass.setError("Required");
        }

        else if(! cpass.getText().toString().equals(pass.getText().toString()))
        {
            cpass.setError("mismatch password");
        }
        else {
            vpc=true;
        }
    }

    @Override
    public void afterTextChanged(Editable editable) {

    }
});

```

```

final int yearx=today.get(Calendar.YEAR);
int month= today.get(Calendar.MONTH)+1;
int year =today.get(Calendar.YEAR);
;
btnSetDate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar cal=Calendar.getInstance();

        final int yearU=cal.get(Calendar.YEAR);
        int monthU=cal.get(Calendar.MONTH);
        int dayU=cal.get(Calendar.DAY_OF_MONTH);
        DatePickerDialog dialog=new DatePickerDialog(
            rootView.getContext(),
            android.R.style.Theme_Holo_Light_Dialog_MinWidth
            , DatePicker1
            ,yearU,monthU,dayU);
        dialog.getWindow().setBackgroundDrawable(new
        ColorDrawable(Color.TRANSPARENT));
        dialog.show();

    }

});
DatePicker1=new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
        month=month+1;
        String date=month+"/"+ dayOfMonth + "/" +year;
        txtDate.setText(date);
        age=yearx-year;
    }
};

ArrayAdapter<CharSequence> dataAdapter
=ArrayAdapter.createFromResource(rootView.getContext(),R.array.gender,android.R.layout.simple_spinner_item);

dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spnGinder.setAdapter(dataAdapter);

btn=(Button) rootView.findViewById(R.id.signup);
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

if(vn&&pc&&vpc&&vphone&&!spnGinder.getSelectedItem().toString().equals("gender")&&!Ca
pital.getSelectedItem().toString().equals("city"))
        if(age>18){
            trySignUp();
        }
        else
        {

```

```

        Toast.makeText(getApplicationContext(), "you can't register you are under age"+age,
        Toast.LENGTH_LONG).show();
    }
    else
        Toast.makeText(getApplicationContext(), "please fill info", Toast.LENGTH_LONG).show();

    }
    });

    return rootView;
}

protected void trySignUp(){

    StringRequest postRequest = new StringRequest(Request.Method.POST,"https://offer-
    system.000webhostapp.com/Register.php",
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                Toast.makeText(getApplicationContext(), response,
                Toast.LENGTH_LONG).show();
                try {

                    JSONObject jsonObject = new JSONObject(response);

                    if (!jsonObject.getString("id").equals("NULL") ) {

                        Toast.makeText(getApplicationContext(), "connect",
                        Toast.LENGTH_LONG).show();

                        Intent intent = new Intent(getApplicationContext(), uploadImage.class);
                        intent.putExtra("user_id",jsonObject.getString("id") );
                        startActivity(intent);

                    } else

                        Toast.makeText(getApplicationContext(), "error",
                        Toast.LENGTH_LONG).show();

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

            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

                if (error instanceof ServerError)
                    Toast.makeText(getApplicationContext(), "Server Error",

```

```

Toast.LENGTH_SHORT).show();
    else if (error instanceof TimeoutError)
        Toast.makeText(getApplicationContext(), "Connection Timed Out",
Toast.LENGTH_SHORT).show();
    else if (error instanceof NetworkError)
        Toast.makeText(getApplicationContext(), "Bad Network Connection",
Toast.LENGTH_SHORT).show();
    }

})

{
    @Override
    protected Map<String, String> getParams() throws AuthFailureError
    {
        Map <String,String> param=new HashMap<String, String>() ;
        param.put("name",fname.getText().toString());
        param.put("user_phone",phone.getText().toString());
        param.put("password",pass.getText().toString());
        param.put("age", String.valueOf(age));
        param.put("gender",spnGinder.getSelectedItem().toString());
        param.put("lon",String.valueOf(a));
        param.put("lat",String.valueOf(a));
        param.put("captial",Capital.getSelectedItem().toString());

        return param;
    }

};
Volley.newRequestQueue(getApplicationContext()).add(postRequest);

}
}

```

3. Upload Image & Select Category

3.1. Upload Image

3.1.1. Upload Image XML Code

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.hp.offermagnet.uploadImage"

```



```
android:background="@color/BackgroudColor">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_centerVertical="true"  
    android:orientation="vertical">
```

```
<de.hdodenhof.circleimageview.CircleImageView
```

```
    android:id="@+id/imageView"  
    app:civ_border_color="@color/colorPrimary"  
    app:civ_border_overlay="true"  
    android:src="@drawable/account"  
    android:layout_margin="20dp"  
    android:textColor="@color/colorBlack"  
    android:padding="20dp"  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:layout_gravity="center_horizontal"/>
```

```
<Button
```

```
    android:id="@+id/buttonUploadImage"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@drawable/sign_back_button"  
    android:layout_margin="30dp"  
    android:layout_gravity="center"  
    android:text="Upload Image" />
```

```
</LinearLayout>
```

```
</RelativeLayout>
```

3.1.2. Upload Image JAVA Code

```
package com.example.hp.offeromagnet;  
  
import android.Manifest;  
import android.app.AlertDialog;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.database.Cursor;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.media.MediaScannerConnection;
```

```

import android.net.Uri;
import android.os.Environment;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;

import android.widget.ImageView;
import android.widget.Toast;

import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Calendar;
import java.util.UUID;

import de.hdodenhof.circleimageview.CircleImageView;

public class uploadImage extends AppCompatActivity {

    private Button btn;
    private CircleImageView imageview;
    private static final String IMAGE_DIRECTORY = "/demonuts";
    private int GALLERY = 1, CAMERA = 2;
    private int PICK_IMAGE_REQUEST = 1;
    private static final int STORAGE_PERMISSION_CODE = 123;
    private Bitmap bitmap;
    private Uri filePath;
    Bundle extras;
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_upload_image);
        requestStoragePermission();

        btn = (Button) findViewById(R.id.buttonUploadImage);
        imageview = findViewById(R.id.imageView);
        extras = getIntent().getExtras();
        imageview.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Intent intent = new Intent();
                intent.setType("image/*");
                intent.setAction(Intent.ACTION_GET_CONTENT);
                startActivityForResult(Intent.createChooser(intent, "Select Picture"),

```

```

PICK_IMAGE_REQUEST);

    }
});

btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        String path = getPath(filePath);

        try {
            String uploadId = UUID.randomUUID().toString();

            //Creating a multi part request
            new MultipartUploadRequest(uploadImage.this, uploadId, "https://offer-
system.000webhostapp.com/AddImage.php")
                .addFileToUpload(path, "image") //Adding file
                .addParameter("user_id", extras.getString("user_id")) //Adding text
parameter to the request
                .setNotificationConfig(new UploadNotificationConfig())
                .setMaxRetries(2)
                .startUpload(); //Starting the upload
            Intent intent = new Intent(uploadImage.this, Category.class);
            intent.putExtra("user_id", extras.getString("user_id"));
            startActivity(intent);

        } catch (Exception exc) {
            Toast.makeText(uploadImage.this, exc.getMessage(),
Toast.LENGTH_SHORT).show();
        }

    }
});

}

//Requesting permission
private void requestStoragePermission() {
    if (ContextCompat.checkSelfPermission(this,
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED)
        return;

    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.READ_EXTERNAL_STORAGE)) {

        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE,
STORAGE_PERMISSION_CODE};
    }
}

```

```

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

    //Checking the request code of our request
    if (requestCode == STORAGE_PERMISSION_CODE) {

        //If permission is granted
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            //Displaying a toast
            Toast.makeText(this, "Permission granted now you can read the storage",
Toast.LENGTH_LONG).show();
        } else {
            //Displaying another toast if permission is not granted
            Toast.makeText(this, "Oops you just denied the permission",
Toast.LENGTH_LONG).show();
        }
    }
}

//handling the image chooser activity result
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data !=
null && data.getData() != null) {
        filePath = data.getData();
        try {
            bitmap = MediaStore.Images.Media.getBitmap(getContentResolver(), filePath);
            imageView.setImageBitmap(bitmap);

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

//method to get the file path from uri
public String getPath(Uri uri) {
    Cursor cursor = getContentResolver().query(uri, null, null, null, null);
    cursor.moveToFirst();
    String document_id = cursor.getString(0);
    document_id = document_id.substring(document_id.lastIndexOf(":") + 1);
    cursor.close();

    cursor = getContentResolver().query(
        android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
        null, MediaStore.Images.Media._ID + " = ? ", new String[]{document_id}, null);
    cursor.moveToFirst();
    String path = cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
    cursor.close();
}

```

```

        return path;
    }

    private void showPictureDialog() {
        AlertDialog.Builder pictureDialog = new AlertDialog.Builder(this);
        pictureDialog.setTitle("Select Action");
        String[] pictureDialogItems = {
            "Select photo from gallery",
            "Capture photo from camera"};
        pictureDialog.setItems(pictureDialogItems,
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    switch (which) {
                        case 0:
                            choosePhotoFromGallary();
                            break;
                        case 1:
                            takePhotoFromCamera();
                            break;
                    }
                }
            });
        pictureDialog.show();
    }

    public void choosePhotoFromGallary() {
        Intent galleryIntent = new Intent(Intent.ACTION_PICK,
            android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);

        startActivityForResult(galleryIntent, GALLERY);
    }

    private void takePhotoFromCamera() {
        Intent intent = new Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent, CAMERA);
    }
}

```

3.2. Select Category

3.2.1. Select Category XML code

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/BackgroudColor"
    tools:context="com.example.hp.offermagnet.Category">

    <ImageView
        android:id="@+id/category"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"
        android:background="@drawable/category" />

    <RelativeLayout
        android:id="@+id/cat"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/category"
        android:gravity="center">

        <TableLayout
            android:id="@+id/table1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="20dp"
            android:layout_marginTop="40dp">

            <Button
                android:id="@+id/books"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:background="@color/BackgroudColor"
                android:drawableBottom="@drawable/books" />

            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"

                android:text="Books"
                android:textAlignment="center"
                android:textColor="@color/colorFocus" />

            <Button
                android:id="@+id/computer"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginTop="5dp"
                android:background="@color/BackgroudColor"
                android:drawableBottom="@drawable/computer" />

            <TextView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="computer"
        android:textAlignment="center"
        android:textColor="@color/colorFocus" />

```

<Button

```

        android:id="@+id/shoses"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"
        android:background="@color/BackgroudColor"
        android:drawableBottom="@drawable/shoses" />

```

<TextView

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Shoses"
        android:textAlignment="center"
        android:textColor="@color/colorFocus" />

```

</TableLayout>

<TableLayout

```

        android:id="@+id/table2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/category"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="40dp"
        android:layout_toEndOf="@id/table1"
        android:layout_toRightOf="@id/table1">

```

<Button

```

        android:id="@+id/foods"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/BackgroudColor"
        android:drawableBottom="@drawable/foods" />

```

<TextView

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="Foods"
        android:textAlignment="center"
        android:textColor="@color/colorFocus" />

```

<Button

```

        android:id="@+id/cellphone"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="5dp"

```

```

        android:background="@color/BackgroudColor"
        android:drawableBottom="@drawable/cellphone" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Cellphones"

    android:textAlignment="center"
    android:textColor="@color/colorFocus" />

<Button

    android:id="@+id/fashions"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:background="@color/BackgroudColor"
    android:drawableBottom="@drawable/fashions" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Fashions"
    android:textAlignment="center"
    android:textColor="@color/colorFocus" />

</TableLayout>

<TableLayout
    android:id="@+id/Table3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/category"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="40dp"
    android:layout_toEndOf="@id/table2"
    android:layout_toRightOf="@id/table1">

    <Button
        android:id="@+id/electronics"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/BackgroudColor"
        android:drawableBottom="@drawable/electronics" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="Electronics"
        android:textAlignment="center"
        android:textColor="@color/colorFocus" />

```



```

<Button
    android:id="@+id/officies"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:background="@color/BackgroudColor"
    android:drawableBottom="@drawable/officies" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Offices"

    android:textAlignment="center"
    android:textColor="@color/colorFocus" />

<Button

    android:id="@+id/collections"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="5dp"
    android:background="@color/BackgroudColor"
    android:drawableBottom="@drawable/collections" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Collections"
    android:textAlignment="center"
    android:textColor="@color/colorFocus" />

</TableLayout>

</RelativeLayout>

<Button

    android:id="@+id/btncontinue"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/cat"
    android:layout_centerHorizontal="true"
    android:layout_marginRight="40dp"
    android:layout_marginTop="40dp"
    android:background="@drawable/continuebutton" />

</RelativeLayout>

```

3.2.2. Select Category JAVA code

```

package com.example.hp.offeromagnet;

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

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import java.util.HashMap;
import java.util.Map;

public class Category extends AppCompatActivity {

    Button btn1;
    Button btn2;
    Button btn3;
    Button btn4;
    Button btn5;
    Button btn6;
    Button btn7;
    Button btn8;
    Button btn9;
    Button btncontinue;
    Bundle extras;

    final HashMap<String, String> category = new HashMap<String, String>();

    boolean check[] = {false, false, false, false, false, false, false, false, false};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_category);
        extras= getIntent().getExtras();
        btn1 = (Button) findViewById(R.id.foods);
        btn2 = (Button) findViewById(R.id.books);
        btn3 = (Button) findViewById(R.id.fashions);
        btn5 = (Button) findViewById(R.id.electronics);
        btn6 = (Button) findViewById(R.id.collections);
        btn7 = (Button) findViewById(R.id.shoses);
        btn8 = (Button) findViewById(R.id.cellphone);
        btn9 = (Button) findViewById(R.id.computer);
        btn4 = (Button) findViewById(R.id.officies);
    }

```

```

btn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        check[0] = !check[0];
        if (check[0] == true) {
            btn1.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat1", "1");
        } else {
            btn1.setBackgroundColor(Color.rgb(249, 251, 255));
        }

    }
});

btn2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[1] = !check[1];
        if (check[1] == true) {
            btn2.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat2", "2");
        } else {
            btn2.setBackgroundColor(Color.rgb(249, 251, 255));
        }

    }
});

btn3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[2] = !check[2];
        if (check[2] == true) {
            btn3.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat3", "3");
        } else {
            btn3.setBackgroundColor(Color.rgb(249, 251, 255));
        }

    }
});

btn4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[3] = !check[3];
        if (check[3] == true) {
            btn4.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat4", "4");
        } else {
            btn4.setBackgroundColor(Color.rgb(249, 251, 255));
        }

    }
});

```

```

    }
});
btn5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[4] = !check[4];
        if (check[4] == true) {
            btn5.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat5", "5");
        } else {
            btn5.setBackgroundColor(Color.rgb(249, 251, 255));
        }
    }
});
btn6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[5] = !check[5];
        if (check[5] == true) {
            btn6.setBackgroundColor(Color.rgb(81, 171, 255));
            // category.put("cat6", "collections");
        } else {
            btn6.setBackgroundColor(Color.rgb(249, 251, 255));
        }
    }
});
btn7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[6] = !check[6];
        if (check[6] == true) {
            btn7.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat3", "3");
        } else {
            btn7.setBackgroundColor(Color.rgb(249, 251, 255));
        }
    }
});
btn8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        check[7] = !check[7];
        if (check[7] == true) {
            btn8.setBackgroundColor(Color.rgb(81, 171, 255));
            category.put("cat5", "5");
        } else {
            btn8.setBackgroundColor(Color.rgb(249, 251, 255));
        }
    }
});
btn9.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View view) {
    check[8] = !check[8];
    if (check[8] == true) {
        btn9.setBackgroundColor(Color.rgb(81, 171, 255));
        category.put("cat5", "5");
    } else {
        btn9.setBackgroundColor(Color.rgb(249, 251, 255));
    }
}

});

btncontinue = (Button) findViewById(R.id.btncontinue);
btncontinue.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        selectCategory();
    }
});
}

void selectCategory() {
    StringRequest postRequest = new StringRequest(Request.Method.POST,
    "https://offer-system.000webhostapp.com/Category.php",
    new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if (!response.contains("No parameters !")) {
                Toast.makeText(Category.this, "connect",
                Toast.LENGTH_LONG).show();
                Intent intent = new Intent(Category.this, UserTab.class);
                startActivity(intent);
            } else
                Toast.makeText(Category.this, "error", Toast.LENGTH_LONG).show();

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

            if (error instanceof ServerError)
                Toast.makeText(Category.this, "Server Error",
                Toast.LENGTH_SHORT).show();
            else if (error instanceof TimeoutError)
                Toast.makeText(Category.this, "Connection Timed Out",
                Toast.LENGTH_SHORT).show();
            else if (error instanceof NetworkError)
                Toast.makeText(Category.this, "Bad Network Connection",
                Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

```

    }
  }) {
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
      HashMap<String, String> select_category = new HashMap<String, String>();
      select_category = category;
      select_category.put("user_id", extras.getString("user_id"));
      return select_category;
    }
  };
  Volley.newRequestQueue(this).add(postRequest);
}
}

```

4. GPS Location

4.1. Location xml Code

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.hp.offermagnet.GPSlocation"
    android:background="@color/BackgroudColor">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@drawable/location_image"
            android:layout_gravity="center"
            android:layout_marginTop="50dp"
            android:layout_centerHorizontal="true"/>

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:background="@drawable/back_location"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="50dp"
            android:id="@+id/back"

```

```

/>
<ImageView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@drawable/divider"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"
    android:id="@+id/divider"
    android:layout_below="@+id/back"
/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorGrayloc"
    android:layout_below="@+id/divider"
    android:textAlignment="center"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="20dp"
    android:textSize="30dp"
    android:textStyle="bold"
    android:text="Enable your location"
    android:id="@+id/txtLOc"
    android:fontFamily="@string/fontUsed"
/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorGrayloc"
    android:layout_below="@+id/txtLOc"
    android:textAlignment="center"
    android:layout_centerHorizontal="true"
    android:fontFamily="@string/fontUsed"
    android:gravity="center"
    android:text="Turn on your location to track yo bus realtime"
    android:id="@+id/txt2loc"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/txt2loc"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="50dp"
    android:background="@drawable/sign_back_button"
    android:text="Turn on location"
    android:id="@+id/turnOnL"
    android:textAlignment="center"
    android:textColor="@color/BackgroudColor"
    android:layout_marginRight="200dp"/>
</RelativeLayout>

</RelativeLayout>

```

4.2. Location Java Code

```
package com.example.hp.offer magnet;

import android.annotation. SuppressLint;
import android.content. Context;
import android.content. Intent;
import android.location. Address;
import android.location. Geocoder;
import android.location. Location;
import android.location. LocationListener;
import android.location. LocationManager;
import android.os. Bundle;
import android.provider. Settings;
import android.support.annotation. Nullable;
import android.support.v4.app. Fragment;
import android.support.v7.app. AppCompatActivity;
import android.util. Log;
import android.view. LayoutInflater;
import android.view. View;
import android.view. ViewGroup;
import android.widget. Button;
import android.widget. TextView;
import android.widget. Toast;

import java.io. UnsupportedEncodingException;
import java.net. URLDecoder;
import java.net. URLEncoder;
import java.util. HashMap;
import java.util. Iterator;
import java.util. List;
import java.util. Locale;
import android. Manifest;
import android.annotation. SuppressLint;
import android.content. Context;
import android.content. Intent;
import android.content. pm. PackageManager;
import android.location. Address;
import android.location. Geocoder;
import android.location. Location;
import android.location. LocationListener;
import android.location. LocationManager;
import android.os. Build;
import android.os. Bundle;
import android.provider. Settings;
import android.support.annotation. NonNull;
import android.support.annotation. Nullable;
import android.support.v4.app. ActivityCompat;
import android.support.v4.app. Fragment;
import android.support.v4.app. FragmentManager;
import android.util. Log;
import android.view. LayoutInflater;
import android.view. View;
import android.view. ViewGroup;
import android.widget. Button;
```



```

import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import java.util.Iterator;
import java.util.List;
import java.util.Locale;
import java.util.Map;

import static android.content.Context.LOCATION_SERVICE;

public class GPSlocation extends AppCompatActivity {

    double MyLat, MyLong;
    String CityName;
    Button btn;
    TextView txt;
    private Location location;
    private LocationListener locationListener;
    Context context;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_gps);
        btn = (Button) findViewById(R.id.turnOnL);
        ActivityCompat.requestPermissions(GPSlocation.this, new String[]
        {Manifest.permission.ACCESS_FINE_LOCATION}, 123);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                getMyCurrentLocation();
                if(MyLat==0.0&&MyLong==0.0){
                    Toast.makeText(GPSlocation.this, "Turn On Location",
                    Toast.LENGTH_SHORT).show();
                }
                else {

                    StringRequest stringRequest = new StringRequest(Request.Method.POST,
                    "https://offer-system.000webhostapp.com/GPSI.php",
                    new Response.Listener<String>() {
                        @Override
                        public void onResponse(String response) {
                            try {
                                String s = URLEncoder.encode(response, "ISO-8859-1");
                                response = URLDecoder.decode(s, "UTF-8");

```

```

        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
        }
        Log.i("tagconvertstr", "[" + response + "]");
        if (response.contains("Done")) {

            Toast.makeText(GPSlocation.this, response,
                Toast.LENGTH_SHORT).show();
        } else if (response.contains(" not Done")) {

            Toast.makeText(GPSlocation.this, response,
                Toast.LENGTH_SHORT).show();
        } else
            Toast.makeText(GPSlocation.this, response,
                Toast.LENGTH_SHORT).show();

    }

    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

            Toast.makeText(GPSlocation.this, error.getMessage(),
                Toast.LENGTH_SHORT).show();
        }
    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            HashMap<String, String> stringStringHashMap = new HashMap<>();
            stringStringHashMap.put("user_id", "45");
            stringStringHashMap.put("city", cityName);
            stringStringHashMap.put("longitude", String.valueOf(MyLong));
            stringStringHashMap.put("latitude", String.valueOf(MyLat));
            return stringStringHashMap;
        }
    };

    Volley.newRequestQueue(GPSlocation.this).add(stringRequest);
}

});

}

}

boolean gps_enabled, network_enabled;
@SuppressLint("MissingPermission")
void getMyCurrentLocation() {

    LocationManager locManager = (LocationManager) getSystemService(
        Context.LOCATION_SERVICE);
    LocationListener locListener = new MyLocationListener();

```

```

        try {
            gps_enabled = locationManager.isProviderEnabled(
LocationManager.GPS_PROVIDER);
        } catch (Exception ex) {
        }
        try {
            network_enabled = locationManager.isProviderEnabled(
LocationManager.NETWORK_PROVIDER);
        } catch (Exception ex) {
        }

        if (gps_enabled) {
            locationManager.requestLocationUpdates( LocationManager.GPS_PROVIDER, 0, 0,
locListener );
        }

        if (gps_enabled) {
            location = locationManager.getLastKnownLocation(
LocationManager.GPS_PROVIDER);

        }

        if (network_enabled && location == null) {
            locationManager.requestLocationUpdates(
LocationManager.NETWORK_PROVIDER, 0, 0, locListener );
        }

        if (network_enabled && location == null) {
            location = locationManager.getLastKnownLocation(
LocationManager.NETWORK_PROVIDER);
        }

        if (location != null) {

            MyLat = location.getLatitude();
            MyLong = location.getLongitude();

        } else {
            Location loc = getLastKnownLocation( this );
            if (loc != null) {

                MyLat = loc.getLatitude();
                MyLong = loc.getLongitude();
            }
        }
    }
}

```

```

    }
}
locManager.removeUpdates( locListener );
try {
    Geocoder geocoder;

    List<Address> addresses;
    geocoder = new Geocoder( this, Locale.getDefault() );
    addresses = geocoder.getFromLocation( MyLat, MyLong, 1 );
    CityName = addresses.get( 0 ).getLocality();

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

Toast.makeText( this, "" + MyLat + "\n" + MyLong + "" + CityName,
Toast.LENGTH_LONG ).show();
Log.e( "Lat: ", MyLat + "" );
Log.e( "Long: ", MyLong + "" );
}

// Location listener class. to get location.
public class MyLocationListener implements LocationListener {
    public void onLocationChanged(Location location) {
        if (location != null) {
        }
    }

    public void onProviderDisabled(String provider) {
        Intent b = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
        startActivity(b);
    }

    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {
        // TODO Auto-generated method stub
    }
}

public static Location getLastKnownLocation(Context context) {
    Location location = null;
    @SuppressWarnings("WrongConstant") LocationManager locationManager =
(LocationManager) context.getSystemService( "location" );
    List list = locationManager.getAllProviders();
    boolean i = false;
    Iterator iterator = list.iterator();

```

```

do {
    //System.out.println("-----");
    if (!iterator.hasNext())
        break;
    String s = (String) iterator.next();
    if (i != false && !locationmanager.isProviderEnabled( s ))
        continue;
    // System.out.println("provider ==> "+s);
    @SuppressWarnings("MissingPermission") Location location1 =
locationmanager.getLastKnownLocation( s );
    if (location1 == null)

        continue;
    if (location != null) {

        float f = location.getAccuracy();
        float f1 = location1.getAccuracy();
        if (f >= f1) {
            long l = location1.getTime();
            long l1 = location.getTime();
            if (l - l1 <= 600000L)
                continue;
        }
    }
    location = location1;

} while (true); return location;}}

```

5. Home (Request and Offer)

TabPageAdapter.java

```

package com.example.hp.offer magnet;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentStatePagerAdapter;
import android.widget.Toast;

public class tabPageAdapter extends FragmentStatePagerAdapter {
    String[] tabArray=new String[]{"Offer", "Request"};
    Integer tabnumber=2;
    String searchText="";
    public tabPageAdapter(FragmentManager fm, String txt)
    {
        super(fm);
        searchText= txt;
    }

    @Override

```

```

    public CharSequence getPageTitle(int position) {

        return tabArray[position];
    }

    @Override
    public Fragment getItem(int position) {
        switch (position){
            case 0:
                //one is an offer fragment
                One f=new One();
                Bundle bundle = new Bundle();
                bundle.putString("srchTxt", searchTxt);
                f.setArguments(bundle);
                return f;
            case 1:
                //two is a Request fragment
                Two r=new Two();
                return r;

        }

        return null;
    }

    @Override
    public int getCount() {

        return tabnumber;
    }
}

```

6. Submit Offer

6.1. Submit Offer XML code

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/offerFrame"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorWhite"
    tools:context="com.example.hp.offer magnet.SubmitOfferFragment">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"

```

```
android:background="@color/colorWhite">
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content">
```

```
<RelativeLayout
```

```
    android:id="@+id/layout1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_marginLeft="16dp"  
    android:layout_marginRight="16dp"  
    android:layout_marginTop="16dp"  
    android:background="@drawable/layout_shape">
```

```
<EditText
```

```
    android:id="@+id/title"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_marginLeft="25dp"  
    android:layout_marginTop="25dp"  
    android:background="@drawable/layout_shape"  
    android:hint="Offer Title"  
    android:inputType="text"  
    android:maxLength="20"  
    android:paddingBottom="10dp"  
    android:paddingLeft="23dp"  
    android:paddingRight="23dp"  
    android:paddingTop="10dp"  
    android:textColor="@color/colorGray"  
    android:textColorHint="#253041" />
```

```
<EditText
```

```
    android:id="@+id/productPrice"  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:layout_below="@id/title"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="10dp"  
    android:background="@drawable/layout_shape"  
    android:hint="price"  
    android:inputType="number"  
    android:maxLength="10"  
    android:paddingLeft="20dp"  
    android:paddingRight="20dp"  
    android:textColor="@color/colorGray"  
    android:textColorHint="#253041" />
```

```
<Spinner
```

```
    android:id="@+id/spinnerCity"
```

```

style="@style/spinner_style"
android:entries="@array/cities"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_below="@id/productPrice"
android:layout_centerHorizontal="true"
android:layout_marginLeft="23dp"
android:layout_marginRight="23dp"
android:layout_marginTop="25dp">

</Spinner>

<EditText
    android:id="@+id/description"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_below="@id/spinnerCity"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="23dp"
    android:layout_marginRight="23dp"
    android:layout_marginTop="25dp"
    android:background="@drawable/layout_shape"
    android:gravity="top"
    android:hint="Note"
    android:inputType="textMultiLine"
    android:maxLength="100"
    android:paddingLeft="23dp"
    android:paddingRight="23dp"
    android:textColor="@color/colorGray"
    android:textColorHint="#253041" />

<RelativeLayout

    android:id="@+id/productlayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/description"
    android:layout_margin="10dp"
    android:background="@drawable/layout_shape">

    <Spinner
        android:id="@+id/spinnerCategory"
        style="@style/spinner_style"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginLeft="25dp"
        android:layout_marginTop="10dp"
        android:entries="@array/Category"
        android:paddingLeft="18dp"
        android:paddingRight="18dp"
        android:pointerIcon="all_scroll"
        android:spinnerMode="dropdown">

</Spinner>

```



```

<Spinner
    android:id="@+id/productName"
    style="@style/spinner_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_below="@id/spinnerCategory"
    android:layout_marginTop="20dp"
    android:hint="Product name"
    android:maxLength="20"
    android:paddingBottom="10dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:paddingTop="10dp" />

```

<EditText

```

    android:id="@+id/productDescription"
    android:layout_width="match_parent"
    android:layout_height="135dp"
    android:layout_below="@id/productName"
    android:layout_marginLeft="23dp"
    android:layout_marginRight="23dp"
    android:layout_marginTop="25dp"
    android:background="@drawable/layout_shape"
    android:gravity="top"
    android:hint="product description"
    android:inputType="textMultiLine"
    android:maxLength="500"
    android:paddingLeft="23dp"
    android:paddingRight="23dp"
    android:textColor="@color/colorGray"
    android:textColorHint="#253041" />

```

<Button

```

    android:id="@+id/addAttachment"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/productDescription"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="17dp"
    android:layout_marginTop="35.49dp"
    android:background="@drawable/sign_back_button"
    android:hint="Add attachment"
    android:textColorHint="@color/colorWhite" />

```

<ImageView

```

    android:id="@+id/show_image"
    android:layout_width="500px"
    android:layout_height="200px"
    android:layout_below="@id/addAttachment"

```

```
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="37dp"
        android:layout_marginTop="15dp" />
</RelativeLayout>
```

```
</RelativeLayout>
```

```
<RelativeLayout
    android:id="@+id/layout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/layout1"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="16dp"
    android:background="@drawable/layout_shape">
```

```
<TextView
    android:id="@+id/textLocation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="21dp"
    android:text="Select Location"
    android:textColor="@color/colorText"
    android:textSize="20sp" />
```

```
<LinearLayout
    android:id="@+id/linear1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textLocation"
    android:layout_marginTop="21dp">
```

```
<ImageView
    android:layout_width="8dp"
    android:layout_height="8dp"
    android:layout_gravity="center"
    android:layout_marginLeft="28dp"
    android:src="@drawable/active_location" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="25dp"
    android:gravity="center"
    android:text="Current Location"
    android:textColor="#253041"
    android:textSize="20sp" />
</LinearLayout>
```

```
<LinearLayout
    android:id="@+id/linear2"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/linear1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="21dp">

        <SeekBar
            android:id="@+id/barLocation"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="25dp"
            android:layout_weight="3"

            android:max="30"
            android:progress="15"

        />

        <ImageButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:layout_weight="1"
            android:background="@drawable/layout_shape"
            android:src="@drawable/swap_btn" />
    </LinearLayout>

    <LinearLayout
        android:id="@+id/linear3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/linear2"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="21dp"

    >

        <ImageView
            android:layout_width="8dp"
            android:layout_height="8dp"
            android:layout_gravity="center"
            android:layout_marginLeft="28dp"
            android:src="@drawable/active_location" />

        <TextView
            android:id="@+id/textKilo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="25dp"
            android:text="15 Kilo Way"
            android:textColor="#253041"
            android:textSize="20sp" />
    </LinearLayout>

    <TextView

```

```

        android:id="@+id/FromDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/linear3"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="45dp"
        android:text="From"
        android:textColor="@color/colorText" />

<LinearLayout
    android:id="@+id/linearFromDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/FromDate"
    android:layout_centerHorizontal="true"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnFromDate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_vertical"
        android:background="@drawable/button_shape"
        android:text="set Date" />

    <TextView
        android:id="@+id/textFromDate"
        android:layout_width="wrap_content"
        android:layout_height="47dp"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="20dp"
        android:text="FromDate"
        android:textColor="@color/colorBlack"
        android:textSize="20sp"

    />
</LinearLayout>

<TextView
    android:id="@+id/ToDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/linearFromDate"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="25dp"
    android:text="To"
    android:textColor="@color/colorText"
    android:textSize="20sp" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/ToDate"

```

```

        android:layout_centerHorizontal="true"
        android:layout_marginBottom="10dp">

        <Button
            android:id="@+id/btnToDate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/ToDate"
            android:layout_gravity="center_vertical"
            android:layout_weight="1"
            android:background="@drawable/button_shape"
            android:text="set Date"

        />

        <TextView
            android:id="@+id/textToDate"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="30dp"
            android:layout_marginTop="20dp"
            android:layout_weight="2"
            android:text="todate"
            android:textColor="@color/colorBlack"
            android:textSize="20sp"

        />
    </LinearLayout>

</RelativeLayout>

<Button
    android:id="@+id/btnOffer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/layout2"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="37dp"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="38dp"
    android:background="@drawable/sign_back_button"
    android:hint="Submit offer"
    android:textColorHint="@color/colorWhite" />
</RelativeLayout>

</ScrollView>
</FrameLayout>

```

6.2. Submit Offer JAVA code

```
package com.example.hp.offer magnet;

import android.Manifest;
import android.app.DatePickerDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.content.ContextCompat;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.NetworkError;
import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.ServerError;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;

import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.IOException;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.HashMap;
```

```

import java.util.Map;
import java.util.UUID;

import static android.app.Activity.RESULT_OK;

public class SubmitOfferFragment extends Fragment {

    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    private DatePickerDialog.OnDateSetListener DatePicker1, DatePicker2;

    public SubmitOfferFragment() {
        // Required empty public constructor
    }

    EditText txtTitle, txtDesc, txtPrice, productDesc;
    Spinner spnCategory, producTitle, spinnerCity;
    Button btnFrom, btnTo;
    TextView txtFrom, txtTo;
    Button btnOffer, addAttach;
    String dateFrom, dateTo;
    ArrayList<String> arraySpinner;
    ImageView show_image;
    private int PICK_IMAGE_REQUEST = 1;
    private static final int STORAGE_PERMISSION_CODE = 123;
    private Bitmap bitmap;
    private Uri filePath;
    HashMap<String,Integer> cate ;
    Database db;

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

        // Inflate the layout for this fragment
        final View view = inflater.inflate(R.layout.fragment_submit_offer, container, false);
        db=new Database(getActivity());
        txtTitle = (EditText) view.findViewById(R.id.title);
        txtDesc = (EditText) view.findViewById(R.id.description);
        producTitle = (Spinner) view.findViewById(R.id.productName);
        txtPrice = (EditText) view.findViewById(R.id.productPrice);
        productDesc = (EditText) view.findViewById(R.id.productDescription);
        spnCategory = (Spinner) view.findViewById(R.id.spinnerCategory);
        txtFrom = (TextView) view.findViewById(R.id.textFromDate);
        txtTo = (TextView) view.findViewById(R.id.textToDate);
        btnFrom = (Button) view.findViewById(R.id.btnFromDate);
        btnTo = (Button) view.findViewById(R.id.btnToDate);
        btnOffer = (Button) view.findViewById(R.id.btnOffer);
        addAttach = (Button) view.findViewById(R.id.addAttachment);
        show_image = (ImageView) view.findViewById(R.id.show_image);
        spinnerCity = (Spinner) view.findViewById(R.id.spinnerCity);

```

```

        final DatePickerDialog.OnDateSetListener datePicker1, datePicker2;

        return view;
    }

    @Override
    public void onStart() {
        super.onStart();
        requestStoragePermission();
        cate = new HashMap<>();
        cate.put("Books",2);
        cate.put("Foods",1);
        cate.put("Electronics",4);
        cate.put("Fashions",3);
        cate.put("Offices",5);
        addAttach.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setType("image/*");
                intent.setAction(Intent.ACTION_GET_CONTENT);
                startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST);
            }
        });

        txtPrice.addTextChangedListener(new TextWatcher() {
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
                txtPrice.setError("your price should be at most one billion");
            }

            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
                if (txtPrice.getText().length() <= 0) {
                    txtPrice.setError("required,please enter your price");
                }
            }
        });

        @Override
        public void afterTextChanged(Editable editable) {
        }
    });

    productDesc.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            productDesc.setError("Enter your description in 100 letter, please be
specific.");
        }

        @Override

```



```

        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            if (productDesc.getText().length() == 0) {
                productDesc.setError("required");
            }
        }

        @Override
        public void afterTextChanged(Editable editable) {
            if (productDesc.getText().length() < 100) {
                productDesc.setError("description must be at least 100 letter, please be
specific");
            }
        }
    });

    txtTitle.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            txtTitle.setError("enter your title in 20 letters");
        }

        @Override
        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            if (txtTitle.getText().length() <= 0) {
                txtTitle.setError("required");
            }
        }

        @Override
        public void afterTextChanged(Editable editable) {
            if (txtTitle.getText().length() <= 20) {
                txtTitle.setError("title must be at least 20 letter");
            }
        }
    });

    txtDesc.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {

        }

        @Override
        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            if (txtDesc.getText().length() <= 0)
                txtDesc.setError("required");
        }

        @Override
        public void afterTextChanged(Editable editable) {

        }
    });
    if (spnCategory.getSelectedItem().toString().trim().equals("Category")) {
        spnCategory.setFocusable(true);
    }

```

```

    }

    btnFrom.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Calendar cal = Calendar.getInstance();
            int year = cal.get(Calendar.YEAR);
            int month = cal.get(Calendar.MONTH);
            int day = cal.get(Calendar.DAY_OF_MONTH);
            DatePickerDialog dialog = new DatePickerDialog(
                getContext(),
                android.R.style.Theme_Holo_Light_Dialog_MinWidth,
                DatePicker1,
                year, month, day);
            dialog.getWindow().setBackgroundDrawable(new
            ColorDrawable(Color.TRANSPARENT));
            dialog.show();

        }

    });
    DatePicker1 = new DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
            month = month + 1;
            dateFrom = month + "/" + dayOfMonth + "/" + year;
            txtFrom.setText(dateFrom);
        }
    };
    btnTo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Calendar cal = Calendar.getInstance();
            int year = cal.get(Calendar.YEAR);
            int month = cal.get(Calendar.MONTH);
            int day = cal.get(Calendar.DAY_OF_MONTH);
            DatePickerDialog dialog = new DatePickerDialog(
                getContext(),
                android.R.style.Theme_Holo_Light_Dialog_MinWidth,
                DatePicker2,
                year, month, day);
            dialog.getWindow().setBackgroundDrawable(new
            ColorDrawable(Color.TRANSPARENT));
            dialog.show();

        }

    });
    DatePicker2 = new DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
            month = month + 1;
            dateTo = month + "/" + dayOfMonth + "/" + year;
            txtTo.setText(dateTo);
        }
    };

```

```

    }
};

spnCategory.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
    {
        loadProductNames(cate.get(spnCategory.getSelectedItem().toString())+"");
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
});

btnOffer.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
//        insertOffer();
        String path = getPath(filePath);

        //Uploading code
        try {
            String uploadId = UUID.randomUUID().toString();

            //Creating a multi part request
            new MultipartUploadRequest(getActivity(), uploadId, "https://offer-
system.000webhostapp.com/InsertOffer.php")
                .addFileToUpload(path, "image") //Adding file
                .addParameter("user_id", db.getId())//Adding text parameter to the request
                .addParameter("title", txtTitle.getText().toString())
                .addParameter("city", spinnerCity.getSelectedItem().toString())
                .addParameter("date_from", dateFrom)
                .addParameter("date_to", dateTo)
                .addParameter("description", txtDesc.getText().toString())
                .addParameter("product_name",
producTitle.getSelectedItem().toString())
                .addParameter("price", txtPrice.getText().toString())
                .addParameter("product_desc", productDesc.getText().toString())
                .addParameter("user_id", db.getId())
                .setNotificationConfig(new UploadNotificationConfig())
                .setMaxRetries(2)
                .startUpload();//Starting the upload

        } catch (Exception exc) {
            Toast.makeText(getActivity(), exc.getMessage(),
Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

```

    public void insertOffer() {
        StringRequest postRequest = new StringRequest(Request.Method.POST,
            "https://offer-system.000webhostapp.com/InsertOffer.php",
            new Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    // Toast.makeText(getContext().getApplicationContext(), response,
                    Toast.LENGTH_LONG).show();
                    try {
                        JSONObject jsonObject = new JSONObject(response);

                        if (jsonObject.getString("response").contains("User Added
Successfully!")) {

                            Toast.makeText(getContext().getApplicationContext(), "connect",
                                Toast.LENGTH_LONG).show();
                            Fragment productDetailFragment = new Offer_Done();
                            FragmentManager ft = getFragmentManager();
                            ft.beginTransaction().replace(R.id.offerFrame,
                                productDetailFragment).commit();

                        } else if (jsonObject.getString("response").equals("OooPS ... something
went wrong!"))

                            Toast.makeText(getContext().getApplicationContext(), "error",
                                Toast.LENGTH_LONG).show();
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                }, new Response.ErrorListener() {
                    @Override
                    public void onErrorResponse(VolleyError error) {

                        if (error instanceof ServerError)
                            Toast.makeText(getContext().getApplicationContext(), "Server Error",
                                Toast.LENGTH_SHORT).show();
                        else if (error instanceof TimeoutError)
                            Toast.makeText(getContext().getApplicationContext(), "Connection Timed
Out", Toast.LENGTH_SHORT).show();
                        else if (error instanceof NetworkError)
                            Toast.makeText(getContext().getApplicationContext(), "Bad Network
Connection", Toast.LENGTH_SHORT).show();
                    }
                })
            }
            @Override
            protected Map<String, String> getParams() throws AuthFailureError {
                Map<String, String> param = new HashMap<String, String>();
                param.put("title", txtTitle.getText().toString());
            }
        }
    }

```

```

        param.put("city", spnCategory.getSelectedItem().toString());
        param.put("date_from", dateFrom);
        param.put("date_to", dateTo);
        param.put("description", txtDesc.getText().toString());
        param.put("product_name", producTitle.getSelectedItem().toString());
        param.put("price", txtPrice.getText().toString());
        param.put("product_desc", productDesc.toString());
        param.put("user_id", String.valueOf(1));
        param.put("cat_id",
String.valueOf(cate.get(spnCategory.getSelectedItem().toString()) ));
        return param;
    }
};

Volley.newRequestQueue(getContext()).add(postRequest);
}

public void loadProductNames(final String s) {
    StringRequest postRequest = new StringRequest(Request.Method.POST,
"https://offer-system.000webhostapp.com/GetProducts.php",
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                // Toast.makeText(getApplicationContext(), response,
                Toast.LENGTH_LONG).show();
                try {
                    JSONObject jsonObject = new JSONObject(response);
                    JSONArray jsonArray = jsonObject.getJSONArray("names");
                    arraySpinner = new ArrayList<>();
                    for (int x = 0; x < jsonArray.length(); x++) {
                        JSONObject jsonObject1 = jsonArray.getJSONObject(x);
                        arraySpinner.add(jsonObject1.getString("name"));
                    }
                    ArrayAdapter<String> adapter = new ArrayAdapter<String>(getActivity(),
                        android.R.layout.simple_spinner_item, arraySpinner);

                    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
                    producTitle.setAdapter(adapter);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

                if (error instanceof ServerError)
                    Toast.makeText(getApplicationContext(), "Server Error",
                    Toast.LENGTH_SHORT).show();
                else if (error instanceof TimeoutError)
                    Toast.makeText(getApplicationContext(), "Connection Timed
                    Out", Toast.LENGTH_SHORT).show();
            }
        });
}

```

```

        else if (error instanceof NetworkError)
            Toast.makeText(getContext().getApplicationContext(), "Bad Network
Connection", Toast.LENGTH_SHORT).show();
    }

    }) {
        @Override
        protected Map<String, String> getParams() throws AuthFailureError {
            HashMap<String, String> hashMap = new HashMap<>();
            hashMap.put("cat_id", s);
            return hashMap;
        }
    };
    Volley.newRequestQueue(getContext()).add(postRequest);
}

private void requestStoragePermission() {
    if (ContextCompat.checkSelfPermission(getActivity(),
Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED)
        return;

    if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),
Manifest.permission.READ_EXTERNAL_STORAGE)) {

    }
    ActivityCompat.requestPermissions(getActivity(), new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE,
STORAGE_PERMISSION_CODE});
}
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {

    if (requestCode == STORAGE_PERMISSION_CODE) {

        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED)
            Toast.makeText(getActivity(), "Permission granted now you can read the
storage", Toast.LENGTH_LONG).show();
        } else {
            //Displaying another toast if permission is not granted
            Toast.makeText(getActivity(), "Oops you just denied the permission",
Toast.LENGTH_LONG).show();
        }
    }
}

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

```

```

        if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data
            != null && data.getData() != null) {
            filePath = data.getData();
            try {
                bitmap = MediaStore.Images.Media.getBitmap(getActivity().getContentResolver(),
filePath);
                show_image.setImageBitmap(bitmap);

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

    public String getPath(Uri uri) {
        Cursor cursor = getActivity().getContentResolver().query(uri, null, null, null, null);
        cursor.moveToFirst();
        String document_id = cursor.getString(0);
        document_id = document_id.substring(document_id.lastIndexOf(":") + 1);
        cursor.close();

        cursor = getActivity().getContentResolver().query(
            android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
            null, MediaStore.Images.Media._ID + " = ? ", new String[]{document_id}, null);
        cursor.moveToFirst();
        String path =
cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
        cursor.close();

        return path;
    }
}

```

7. Submit Request

7.1. Submit Request XML code

```

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/frameRequest"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/colorWhite"
    tools:context="com.example.hp.offermagnet.SubmitRequestFragment">

    <ScrollView

```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
```

```
>
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

```
<RelativeLayout
```

```
    android:id="@+id/layout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="20dp"
    android:layout_marginRight="20dp"
    android:layout_marginTop="16dp"
    android:background="@drawable/layout_shape">
```

```
<EditText
```

```
    android:id="@+id/title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="25dp"
    android:layout_marginRight="25dp"
    android:layout_marginTop="25dp"
    android:background="@drawable/layout_shape"
    android:fontFamily="Gilroy"
    android:gravity="center_vertical"
    android:hint="Request Title"
    android:inputType="text"
    android:maxLength="50"
    android:paddingBottom="10dp"
    android:paddingLeft="23dp"
    android:paddingRight="23dp"
    android:paddingTop="10dp"
    android:textColor="@color/colorGray"
    android:textColorHint="#253041" />
```

```
<Spinner
```

```
    android:id="@+id/spinnerCategory"
    style="@style/spinner_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/title"
    android:layout_centerHorizontal="true"
    android:layout_marginLeft="25dp"
    android:layout_marginRight="25dp"
    android:layout_marginTop="25dp"
    android:entries="@array/Category"
    android:fontFamily="Gilroy"
    android:paddingBottom="10dp"
```



```

        android:paddingLeft="18dp"
        android:paddingRight="23dp"
        android:paddingTop="10dp"
        android:pointerIcon="all_scroll"
        android:spinnerMode="dropdown"
        android:textColor="@color/colorGray"

    >

</Spinner>

<EditText
    android:id="@+id/description"
    android:layout_width="match_parent"
    android:layout_height="200dp"
    android:layout_below="@+id/spinnerCategory"
    android:layout_marginBottom="25dp"
    android:layout_marginLeft="23dp"
    android:layout_marginRight="23dp"
    android:layout_marginTop="25dp"
    android:background="@drawable/layout_shape"
    android:fontFamily="Gilroy"
    android:gravity="top"
    android:hint="Note"
    android:inputType="textMultiLine"
    android:maxLength="500"
    android:paddingLeft="23dp"
    android:paddingRight="23dp"
    android:textColor="@color/colorGray"
    android:textColorHint="#253041" />
<Spinner
    android:id="@+id/spnCity"
    android:layout_marginBottom="25dp"
    android:layout_marginLeft="23dp"
    android:layout_marginRight="23dp"
    android:layout_centerHorizontal="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:entries="@array/cities"
    style="@style/spinner_style"
    android:pointerIcon="all_scroll"
    android:spinnerMode="dropdown"
    android:layout_below="@id/description"

    ></Spinner>

</RelativeLayout>

<RelativeLayout
    android:id="@+id/layout2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/layout1"
    android:layout_centerHorizontal="true"

```

```
android:layout_marginLeft="20dp"
android:layout_marginRight="20dp"
android:layout_marginTop="16dp"
android:background="@drawable/layout_shape">
```

```
<TextView
    android:id="@+id/textLocation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="21dp"
    android:text="Select Location"
    android:textColor="@color/colorText" />
```

```
<LinearLayout
    android:id="@+id/linear1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textLocation"
    android:layout_marginTop="21dp">
```

```
<ImageView
    android:layout_width="8dp"
    android:layout_height="8dp"
    android:layout_gravity="center"
    android:layout_marginLeft="28dp"
    android:src="@drawable/active_location" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="25dp"
    android:gravity="center"
    android:text="Current Location"
    android:textColor="#253041" />
</LinearLayout>
```

```
<LinearLayout
    android:id="@+id/linear2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/linear1"
    android:layout_marginTop="21dp">
```

```
<SeekBar
    android:id="@+id/barLocation"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="25dp"
    android:layout_weight="3"
    android:max="30"
    android:progress="15"
```

```
/>
```

```

<ImageButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginRight="25dp"
    android:layout_weight="1"
    android:background="@drawable/layout_shape"
    android:src="@drawable/swap_btn" />
</LinearLayout>

<LinearLayout
    android:id="@+id/linear3"
    android:layout_width="280dp"
    android:layout_height="wrap_content"
    android:layout_below="@id/linear2"
    android:layout_marginTop="21dp"

    >

    <ImageView
        android:layout_width="8dp"
        android:layout_height="8dp"
        android:layout_gravity="center"
        android:layout_marginLeft="28dp"
        android:src="@drawable/active_location" />

    <TextView
        android:id="@+id/textKilo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="25dp"
        android:text="15 Kilo Way"
        android:textColor="#253041" />
</LinearLayout>

<TextView
    android:id="@+id/textDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/linear3"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="45dp"
    android:text="Select Validate Date"
    android:textColor="@color/colorText" />

<LinearLayout
    android:id="@+id/LinearDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textDate"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="15dp"
    android:orientation="horizontal">

```

```

<Button
    android:id="@+id/btnSetDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/textDate"
    android:layout_centerHorizontal="true"
    android:layout_weight="1"
    android:background="@drawable/button_shape"
    android:fontFamily="Gilroy"
    android:hint="Set date"
    android:textColor="@color/colorWhite"
    android:textColorHint="@color/colorWhite"></Button>

<TextView
    android:id="@+id/txtDate"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="50dp"
    android:layout_weight="3"
    android:text="Date Text"
    android:textColor="@color/colorBlack"
    android:textSize="25sp"

/>
</LinearLayout>

<Button

    android:id="@+id/addAttachment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/LinearDate"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="36.45dp"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:layout_marginTop="35.49dp"
    android:background="@drawable/sign_back_button"
    android:hint="Add attachment"
    android:textColorHint="@color/colorWhite"/>

<ImageView
    android:id="@+id/show_image2"
    android:layout_width="500px"
    android:layout_height="200px"
    android:layout_below="@id/addAttachment"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="37dp"
    android:layout_marginTop="15dp" />

</RelativeLayout>

<Button
    android:id="@+id/submitRequest"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/layout2"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="37dp"
        android:layout_marginLeft="25dp"
        android:layout_marginRight="25dp"
        android:layout_marginTop="38dp"
        android:background="@drawable/sign_back_button"
        android:text="Submit request"
        android:textColorHint="@color/colorWhite" />
    </RelativeLayout>

</ScrollView>
</FrameLayout>

```

7.2. Submit Request JAVA code

```

package com.example.hp.offer magnet;

import android.Manifest;
import android.app.DatePickerDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.content.ContextCompat;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

```

```

import net.gotev.uploadservice.MultipartUploadRequest;
import net.gotev.uploadservice.UploadNotificationConfig;

import java.io.IOException;
import java.util.Calendar;
import java.util.HashMap;
import java.util.UUID;

import static android.app.Activity.RESULT_OK;

public class SubmitRequestFragment extends Fragment {

    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";
    private DatePickerDialog.OnDateSetListener datePicker;
    Button btnSetDate;
    TextView txtDate;
    Spinner spnCity;
    private String mParam1;
    private String mParam2;
    HashMap<String,Integer> cate ;

    public SubmitRequestFragment() {
        // Required empty public constructor
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    TextView txtMeters;
    SeekBar seekbar;
    EditText txtTitle;
    Spinner spnCategory;
    EditText txtDesc;
    Button btnSubmit, addAttach;
    String date;
    int a = 1;
    ImageView show_image;
    private int PICK_IMAGE_REQUEST = 1;
    private static final int STORAGE_PERMISSION_CODE = 123;
    private Bitmap bitmap;
    private Uri filePath;
    Database db;

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

```

```

View view = inflater.inflate(R.layout.fragment_submit_request, container, false);
cate = new HashMap<>();
cate.put("Books",2);
cate.put("Foods",1);
cate.put("Electronics",4);
cate.put("Fashions",3);
cate.put("Offices",5);
db=new Database(getActivity());
spnCity=(Spinner)view.findViewById(R.id.spnCity);
txtMeters = (TextView) view.findViewById(R.id.textKilo);
seekbar = (SeekBar) view.findViewById(R.id.barLocation);
requestStoragePermission();
spnCategory=(Spinner)view.findViewById(R.id.spinnerCategory) ;
addAttach = (Button) view.findViewById(R.id.addAttachment);
show_image = (ImageView) view.findViewById(R.id.show_image2);
if (spnCategory.getSelectedItem().toString().trim().equals("Category")) {
    spnCategory.setFocusable(true);
}

if (spnCity.getSelectedItem().toString().trim().equals("Category")) {
    spnCity.setFocusable(true);
}

seekbar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
        txtMeters.setText(seekbar.getProgress() + " kilo maeters");
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
});
txtTitle = (EditText) view.findViewById(R.id.title);
txtDesc = (EditText) view.findViewById(R.id.description);
txtTitle.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View view, boolean b) {
        if (txtTitle.getText().length() == 0) {
            txtTitle.setError("required");
        }
    }
});
txtDesc.setOnFocusChangeListener(new View.OnFocusChangeListener() {
    @Override
    public void onFocusChange(View view, boolean b) {
        if (txtDesc.getText().length() == 0) {
            txtDesc.setError("required");
        }
    }
});

```

```

    }
}
});

addAttach.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        startActivityForResult(Intent.createChooser(intent, "Select Picture"),
PICK_IMAGE_REQUEST);
    }
});

btnSetDate = (Button) view.findViewById(R.id.btnSetDate);
txtDate = (TextView) view.findViewById(R.id.txtDate);
btnSetDate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Calendar cal = Calendar.getInstance();
        int year = cal.get(Calendar.YEAR);
        int month = cal.get(Calendar.MONTH);
        int day = cal.get(Calendar.DAY_OF_MONTH);
        DatePickerDialog dialog = new DatePickerDialog(
            v.getContext(),
            android.R.style.Theme_Holo_Light_Dialog_MinWidth
            , datePicker
            , year, month, day);
        dialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
        dialog.show();

    }
});
datePicker = new DatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
        month = month + 1;
        date = month + "-" + dayOfMonth + "-" + year;
        txtDate.setText(date);
    }
};
btnSubmit = (Button) view.findViewById(R.id.submitRequest);
btnSubmit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (txtTitle.getText().toString().equals("") || txtDesc.getText().toString().equals("")
|| date.equals("")) {
            Toast.makeText(getApplicationContext(), "Invalid",
Toast.LENGTH_SHORT).show();
        } else {

            String path = "";

```



```

        try {
            path = getPath(filePath);

        } catch (Exception e) {
            Toast.makeText(getActivity(), "Can't get Image Path",
            Toast.LENGTH_LONG).show();
        }
        //Uploading code
        try {
            String uploadId = UUID.randomUUID().toString();
            Toast.makeText(getActivity(),
            path+"\n"+txtTitle.getText().toString()+"\n"+txtDesc.getText().toString()+"\n"+txtDate.getText().toString(), Toast.LENGTH_SHORT).show();

            //Creating a multi part request
            new MultipartUploadRequest(getActivity(), uploadId, "https://offer-
            system.000webhostapp.com/InsertRequest.php")
                .addFileToUpload(path, "image") //Adding file
                .addParameter("title", txtTitle.getText().toString())
                .addParameter("dis", txtDesc.getText().toString())
                .addParameter("validate_date", txtDate.getText().toString()+"val")

            .addParameter("cat_id",String.valueOf(cate.get(spnCategory.getSelectedItem().toString())
            ))

                .addParameter("user_id", db.getId())
                .addParameter("city", spnCity.getSelectedItem().toString())
                .setNotificationConfig(new UploadNotificationConfig())
                .setMaxRetries(2)
                .startUpload();

            Fragment done = new Task_is_complete();
            FragmentManager ft = getFragmentManager();
            ft.beginTransaction().replace(R.id.frameRequest, done).commit();

        } catch (Exception exc) {
            Toast.makeText(getActivity(), exc.getMessage(),
            Toast.LENGTH_SHORT).show();
        }
    }

    }

    });
    return view;
}

private void requestStoragePermission() {
    if (ContextCompat.checkSelfPermission(getActivity(),
    Manifest.permission.READ_EXTERNAL_STORAGE) ==
    PackageManager.PERMISSION_GRANTED)
        return;

    if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),
    Manifest.permission.READ_EXTERNAL_STORAGE)) {
    }
    ActivityCompat.requestPermissions(getActivity(), new

```

```

String[] {Manifest.permission.READ_EXTERNAL_STORAGE,
STORAGE_PERMISSION_CODE};
}

//This method will be called when the user will tap on allow or deny
@Override
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {

    //Checking the request code of our request
    if (requestCode == STORAGE_PERMISSION_CODE) {

        //If permission is granted
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            //Displaying a toast
            Toast.makeText(getActivity(), "Permission granted now you can read the
storage", Toast.LENGTH_LONG).show();
        } else {
            //Displaying another toast if permission is not granted
            Toast.makeText(getActivity(), "Oops you just denied the permission",
Toast.LENGTH_LONG).show();
        }
    }
}

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

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK && data
!= null && data.getData() != null) {
        filePath = data.getData();
        try {
            bitmap = MediaStore.Images.Media.getBitmap(getActivity().getContentResolver(),
filePath);
            show_image.setImageBitmap(bitmap);

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

//method to get the file path from uri
public String getPath(Uri uri) {
    Cursor cursor = getActivity().getContentResolver().query(uri, null, null, null, null);
    cursor.moveToFirst();
    String document_id = cursor.getString(0);
    document_id = document_id.substring(document_id.lastIndexOf(":") + 1);
    cursor.close();

    cursor = getActivity().getContentResolver().query(
        android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI,
        null, MediaStore.Images.Media._ID + " = ? ", new String[]{document_id}, null);
    cursor.moveToFirst();

```

```

        String path =
        cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
        cursor.close();

        return path;
    }

}

```

8. Navigation menu

8.1. Navigation menu XML code

```

<?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

        layout="@layout/app_bar_nav_drawer"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <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:background="@color/colorPrimary"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_nav_drawer"
        app:menu="@menu/activity_nav_drawer_drawer" />

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

```

8.2. Navigation menu java code

```
package com.example.hp.offeromagnet;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import com.example.hp.offeromagnet.services.NotificationService;
import com.squareup.picasso.Picasso;

import de.hdodenhof.circleimageview.CircleImageView;

//import com.ittianyu.bottomnavigationviewex.BottomNavigationViewEx;

public class NavDrawer extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {
    CircleImageView imageView;
    Fragment fragment;
    FragmentManager fragmentManager;
    TextView name;
    Database db;
    EditText search_text;
    Button do_search;
    SharedPreferences.Editor prefEditor;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_nav_drawer);

        startService(new Intent(this, NotificationService.class));

        fragmentManager = getSupportFragmentManager();
        db = new Database(this);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        name = (TextView) findViewById(R.id.name);
```

```

search_text= findViewById(R.id.search_text);
do_search= findViewById(R.id.do_search);
do_search.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        fragment = new HomeFragment();
        Bundle bundle= new Bundle();
        bundle.putString("srchTxt", search_text.getText().toString());
        fragment.setArguments(bundle);
        fragmentManager.beginTransaction();
        FragmentTransaction transaction = fragmentManager.beginTransaction();
        prefEditor= getSharedPreferences("AppPreferences",
Context.MODE_PRIVATE).edit();
        transaction.add(R.id.container_body, fragment).addToBackStack( "home"
).commit();

    }
});

DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
drawer.addDrawerListener(toggle);
toggle.syncState();

NavigationView navigationView = (NavigationView) findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
View view = navigationView.getHeaderView(0);
name = (TextView) findViewById(R.id.yourname);

imageView = view.findViewById(R.id.imageProfilee);
//if (imageView != null)
Picasso.with(this)
    .load(db.getImage())
    .into(imageView);
//name.append(db.getName());

fragment = new HomeFragment();
Bundle bundle= new Bundle();
bundle.putString("srchTxt", "");
fragment.setArguments(bundle);
fragmentManager.beginTransaction();
FragmentTransaction transaction = fragmentManager.beginTransaction();
transaction.add(R.id.container_body, fragment).addToBackStack( "home" ).commit();

}

@Override
public void onBackPressed() {
    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    if (drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {

```

```

        super.onBackPressed();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.nav_drawer, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

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

    if (id == R.id.Home) {
        fragment = new HomeFragment();
        Bundle bundle = new Bundle();
        bundle.putString("srchTxt", "");
        fragment.setArguments(bundle);
        fragmentManager.beginTransaction();
        FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.add(R.id.container_body, fragment).addToBackStack("home")
    ).commit();

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

    } else if (id == R.id.wishlist) {
        fragment = new WishlistFragment();
        fragmentManager.beginTransaction();
        final FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.container_body, fragment).addToBackStack("wishlist")
    ).commit();
    } else if (id == R.id.myOrder) {
        fragment = new MyOrderFragment();
        fragmentManager.beginTransaction();
        final FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.container_body, fragment).addToBackStack("myorder")
    ).commit();

    } else if (id == R.id.MyProfile) {
        fragment = new ProfileFragment();
        fragmentManager.beginTransaction();

```

```

        final FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.container_body, fragment).addToBackStack( "myprofile"
    ).commit();

    }    else if (id==R.id.privacy) {
        fragment = new PrivacyFragment();
        fragmentManager.beginTransaction();
        final FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.container_body, fragment).addToBackStack( "myprofile"
    ).commit();

    }

    else if (id==R.id.logout) {

        Intent intent = new Intent(NavDrawer.this,UserTab.class );
        prefEditor.putBoolean("logged",false);
        prefEditor.apply();
        startActivity(intent);
    }
    else if (id==R.id.help) {
        fragment = new FQFragment();
        fragmentManager.beginTransaction();
        final FragmentTransaction transaction = fragmentManager.beginTransaction();
        transaction.replace(R.id.container_body, fragment).addToBackStack( "myprofile"
    ).commit();

    }

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

```

4.4. Machine learning code

4.4.1. Introduction RecommenderSystem

Recommendation systems are a type of information filtering that presents lists of items (films, songs, books, videos, images, products, web pages...) which are likely of user interest.

Amazon, Last.fm, Netflix, Pandora are the most popular recommender systems all over the world. Simply they compare user interest acquired from his/her profile with reference characteristics and predict the rating that the user would give.

Two most ubiquitous types of recommender systems are:

1) Content-Based

Content-based recommender systems focus on the attributes of the items and give you recommendations based on the similarity between them.

2) Collaborative filtering (CF)

Collaborative filtering produces recommendations based on the knowledge of users' attitude to items, that is it used to recommend items.

CF can be divided into Memory-Based Collaborative Filtering and Model-Based Collaborative filtering.

2.1) Model-based methods: Model-based algorithms make use of machine learning and data mining methods to generate predictive models that will be used to predict missing entries in the ratings matrix R

2.2) Memory-Based Collaborative Filtering approaches can be divided into two main sections: user-item filtering and item-item filtering.

2.2.1) user-item filtering "Users who are similar to you also liked" takes a particular user, find users that are similar to that user based on similarity of ratings, and recommend items that those similar users liked.

2.2.2) item-item filtering "Users who liked this item also liked ..." will take an item, find users who liked that item, and find other items that those users or similar users also liked. It takes items and outputs other items as recommendations.

Memory-Based Collaborative Filtering techniques rely heavily on simple similarity measures (Cosine similarity, Pearson correlation, etc.) to match similar people or items together. If we have a huge matrix with users on one dimension and

items on the other, with the cells containing Rating or likes, then memory-based techniques use similarity measures on two vectors (rows or columns) of such a matrix to generate a number representing similarity.

In This Project , we use Memory-Based Collaborative Filter (User-Item) by using Pearson Correlation Technique to Measure Similarity Because it suffers from computing high similarity between users with few ratings in common.

4.4.2. Implementation ML codeing

```
def pearson_correlation(person1, person2):

    both_rated = {}
    for item in dataset[person1]:
        if item in dataset[person2]:
            both_rated[item] = 1

    number_of_ratings = len(both_rated)
    if number_of_ratings == 0:
        return 0

    person1_preferences_sum = sum([int(dataset[person1][item]) for item in both_rated])
    person2_preferences_sum = sum([int(dataset[person2][item]) for item in both_rated])

    person1_square_preferences_sum = sum([pow(int(dataset[person1][item]), 2) for item in both_rated])
    person2_square_preferences_sum = sum([pow(int(dataset[person2][item]), 2) for item in both_rated])

    product_sum_of_both_users = sum([int(dataset[person1][item]) * int(dataset[person2][item]) for item in both_rated])

    numerator_value = product_sum_of_both_users - (person1_preferences_sum * person2_preferences_sum / number_of_ratings)
    denominator_value = sqrt((person1_square_preferences_sum - pow(person1_preferences_sum, 2) / number_of_ratings) *
                              (person2_square_preferences_sum - pow(person2_preferences_sum, 2) / number_of_ratings))
    if denominator_value == 0:
        return 0
    else:
        r = numerator_value / denominator_value
        return r
```

Figure 35 : ML Coding

Future Features

- Separate consumers from producers.
- Allow the usage of credit cards.
- Deal with mobile providers to verify phone numbers.
- Detect fake users who join requests.
- Give every user a wallet.
- Create group chat to have a multiway communication with the initial requestor.
- Approving best offers based on different cases.
- Use a cloud platform architecture.
- Develop ML to classify results based on gender and age.
- Use ML to improve the search result by analyzing the user behavior.