

BABU SUNDER SINGH INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(Approved By AICTE, Govt. of India & Affiliated to AKTU (formally UPTU) and BTE Lucknow UP)

(AKTU Code - 648, BTE Code - 2289)

SCHOOL OF SCIENCE AND ENGINEERING

Grocery Android Application

SCHOOL OF SCIENCE AND ENGINEERING

FINAL PROJECT REPORT

Student: Jaykaran, Ravi Kumar Gautam,
Jayhind Yadav

HOD: Dr. Hira Sir

Guide By:- Mr. B.N. Sir

CSE Final Year Student 2020

GROCERY ANDROID APPLICATION

Student Statement:

I, Jay Karan, assert that I have applied ethics to the design process and in the selection of the final proposed design. I also affirm that I have held the safety of the public to be paramount and have addressed this in the presented design wherever may be applicable.

Jay Karan

student: Jay Karan

ACKNOWLEDGEMENTS

First, I would like to express my thankfulness to Dr. Hira Sir, my supervisor, who assisted me throughout the development of the project. He provided me with the necessary suggestions that guided me and saved me a lot of time. I would also like to thank him for -- accepting me as a supervisee and giving me the opportunity to work on a project that highly motivates me.

I would also like to thank BSSITM and the School of Science and Engineering for giving me the opportunity to work my capstone project, which I intend to turn into a startup, in a healthy work environment. I would also like to thank SSE for providing me with the necessary resources that allowed me to develop a successful project.

I would like to thank my Group Jay Karan, Ravi Kumar Gautam and Jayhind Yadav for the moral support and helpfulness that they provided me with throughout my journey BSSITM.

Last but not least, I would like to express my deepest gratitude to my parents who invested in me both morally and financially, for their continuous support and without whom, I would probably not be here today.

TABLE OF CONTENTS

1	ABSTRACT	5
2	INTRODUCTION.....	7
3	STEEPLE Analysis	8
4	REQUIREMENTS SPECIFICATIONS.....	9
5	APP DESIGN AND BACKEND.....	33
6	FEASIBILITY STUDY	34
7	METHODOLOGY.....	36
8	SOFTWARE ARCHITECTURE	48
9	DESIGN.....	48
	9.1 Use Case Diagram	52
	9.2 Interaction Overview Diagram	60
	9.3 Data Model.....	70
10	IMPLEMENTATION	80
11	TECHNOLOGY ENABLERS.....	98
12	TESTING.....	101
13	Conclusion & Future Perspectives:.....	102

1 ABSTRACT

Over the this years, online commerce has become very popular. In India, the culture of is, slowly but surely, kicking in. More than that, there is a smooth shift to using Android devices for shopping. Therefore, “Deshi Touch Grocery”, which is a Grocery company located in Tangier, decided to expand their business open up to online market through an system. This will get new clients from all over India to use their system.

This project, initially, started in the Database Systems course, where a desktop application was developed for the management of the database. A web application was also developed as an interface for the .

This capstone project started with the design of a more sophisticated backend, database with a web application synchronized with the Android application’s database to be implemented. Afterwards, the online shopping platform was implemented as a Android application.

The objective of this project is to have a complete, reliable, and unique system that is expected to be an added value for the company.

2 INTRODUCTION

The Capstone Design course is required by the School of Science and Engineering as partial fulfillment of the bachelor degree in BSSITM. Throughout the development of the capstone project, the student about to graduate demonstrates the knowledge acquired during his four years as an undergraduate student. The development of this project required the use of most of the learned computer science skills: Database Design, Web Development, Desktop Application Development, Software Engineering, and Android Applications Development.

In 2020, a survey was conducted by the National Telecommunications Regulatory Agency in India about the usage of Android phones in India. The survey's results show that of individuals are equipped with a smart phone, which is

million units by 2020 and it is growing fast. The number of smart phone owners is growing very fast.

Deshi Touch Grocery is a store that offers various spice and accessories for Men, Women and Children and meets the needs of the whole family daily need. As a specialist in the distribution of sporting goods in India, DESHI TOUCH GROCERY offers its customers a wide variety of sports products.

Therefore, the development of a Android application for online shopping is a very promising project since the use of smart phones is still growing and the market is not yet saturated. That is because the online shopping culture is not very old in India.

The system that I am implementing in this capstone project is a complete system, consisted in a database, web and Android application (for customers), and a desktop application (for the administrator). However, the main focus of my capstone is on the Android Application.

3 STEEPLE Analysis

Socio-cultural:

Android phones have massively changed our society. They changed the way people live. In fact, almost everybody has a smart phone and use it on a daily basis for different tasks. Now, with phones full of applications, the first thing most us do in the morning is checking our smartphone. Android applications have changed the way we use our phones. We can say that our society is rapidly changing towards a very widespread use of Android technologies. More than that, having a smart phone and using it for shopping is becoming a trend. Therefore, our Grocery shop needs to keep up with the societal changes in order to match our consumer needs and preferences.

Technological:

The market is rapidly changing, many of these changes are because of technological improvements. These advancements can create new markets and new opportunities. An online shop, as a Android application, associated with our physical store is going to be the first of its kind in the Grocery market. This will allow the company to keep up with the technological trend and run the store more effectively. It will also improve the communication between the company and its consumers. This will improve the user experience as he will be informed about all the new products and have them delivered to him.

Economical:

The Android application will be free to download for all customers. It is developed using open- source/free tools and programming languages, the profit will be mainly from the purchases made. More than that, the system developed is expected to guarantee economic growth for the company thanks to the expansion to the online market. It is also expected to gain user confidence. Also, since the will be launched in parallel with the actual physical shop, this Android application online shop will not affect the workflow of the overall company. This could also

result in improving sales of Grocery products; which might reduce prices and increase competition.

Environmental:

Developing this online store will somehow be in favor of the environment. It will save the energy needed for the customers to physically visit the shop (fuel). This will preserve energy and prevent the pollution of the local environment. Also, creating a digitized version of the company will decrease the use of papers, hence preserve the environment.

Political:

In the context, there are no direct consequences of launching a Android application online store. In fact, the government of India seem to have a positive position when it comes to new technologies.

Legal:

The Android application that is going to be developed is completely legal since it doesn't infract any law in the Indian constitution. It won't be harmful in any way and it won't publish the users' private/personal information. Also, the company will pay the taxes for the sales made through the Android application.

Ethical:

The Android application will follow the code of ethics and won't harm in any way the customers, the company, the employees and the society as a whole. The consumers' rights will be respected to the maximum extent. The marketing techniques will be telling the truth about the products offered. The employees will be informed about the application's mission and goals. Also, the information of users will be encrypted and stored in secure database.

4 REQUIREMENTS SPECIFICATIONS

4.1 Functional Requirements

Introduction:

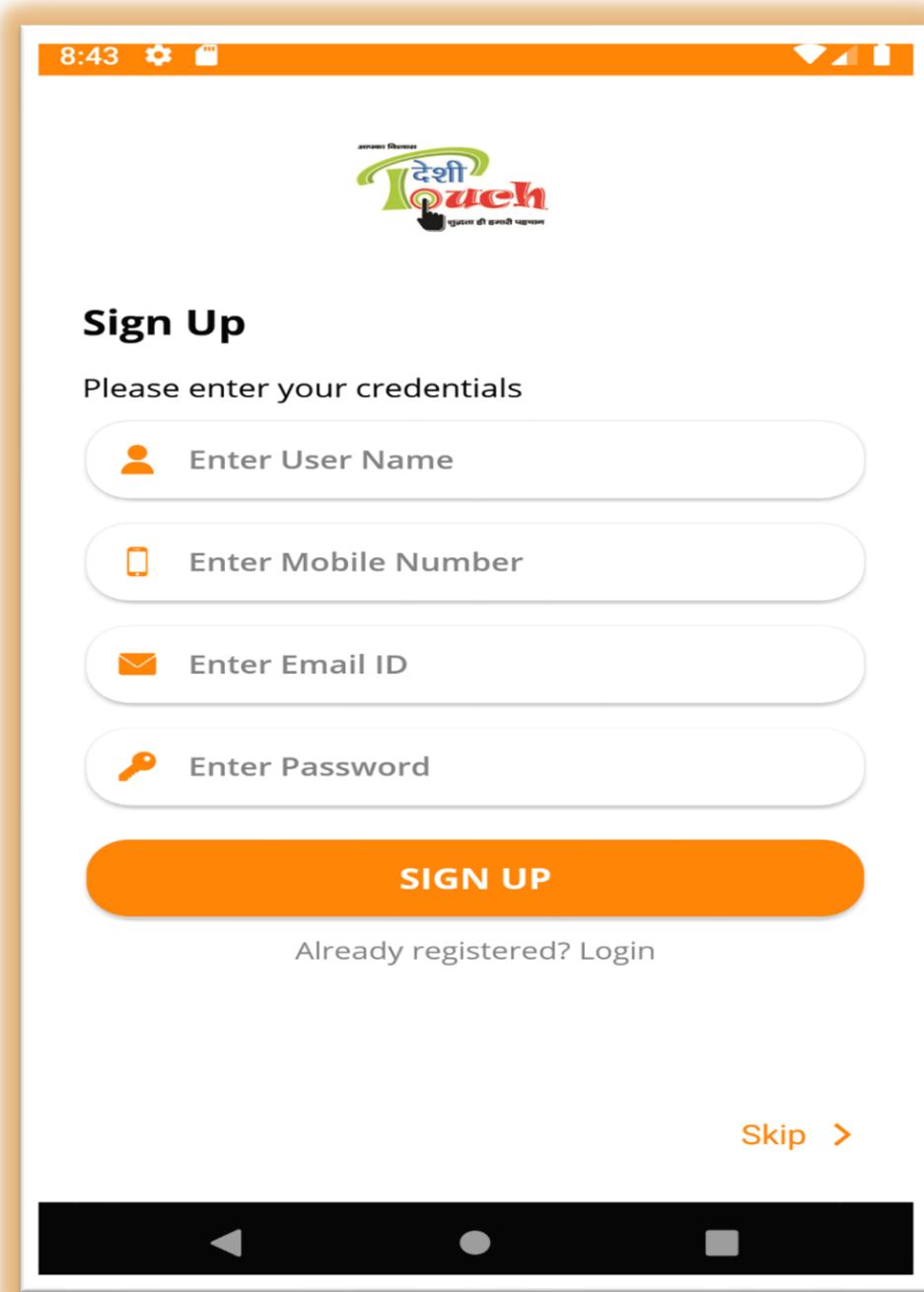
The project is composed of a Web Application (Already developed in previous work on the project), a desktop application (already developed but will be optimized and updated), these last two are linked to backend database (MySQL), and a Android application that is going to be implemented together with a backend database (Firebase) and synchronized with the MySQL one.

This functional requirements section will be mainly about the Android application but will also touch the desktop application that will take care of the management of orders, accounts and products. It is dedicated for the administrator of the system.

ANDROID APPLICATION:

Register:

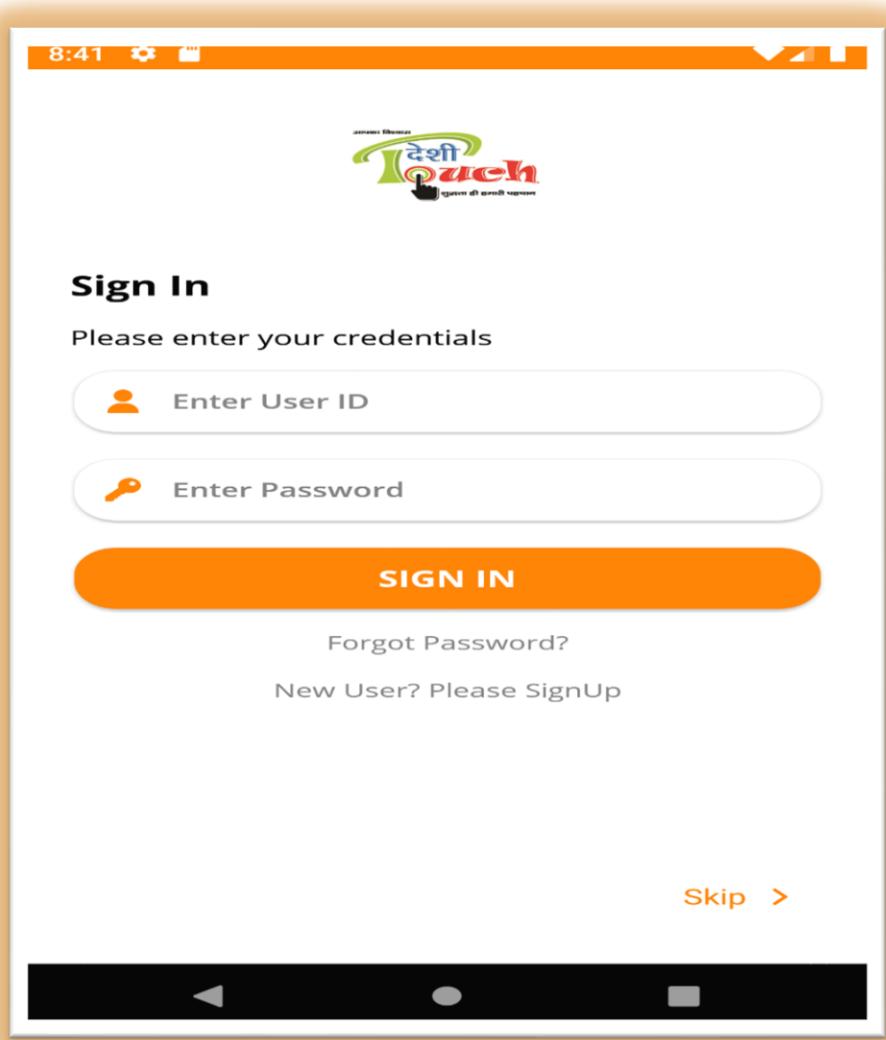
The user of the application must be able to create an account. While creating his account, he must provide personal info: Name, email address, Android number, Password, Confirm Password, He will have to confirm his Android number.



Login:

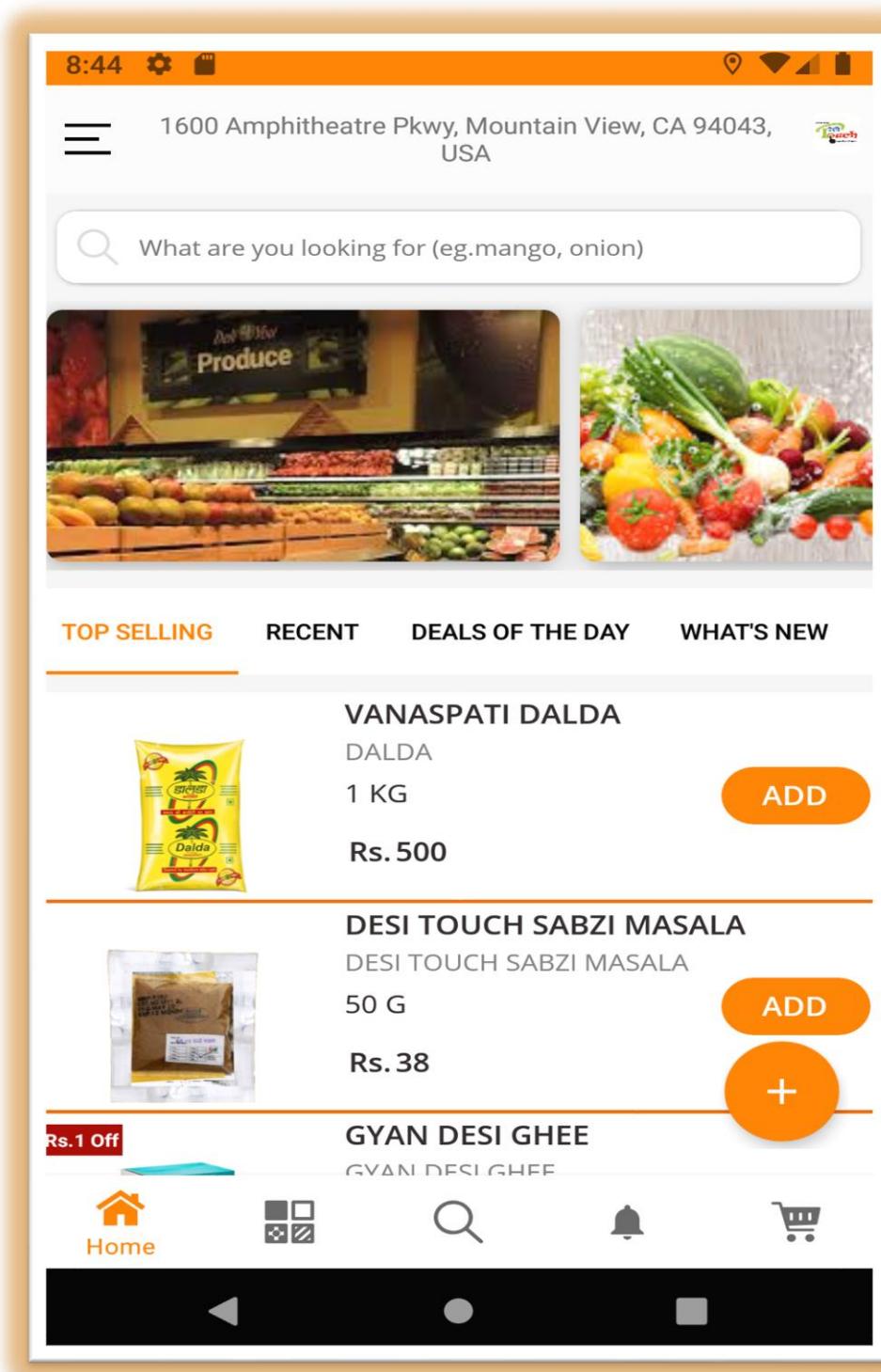
The Android application must have login activity with a field for a username and a password. The user will be able to login if he confirmed his email address. After logging in, the user will be able to:

- view/modify his personal info.
- view latest products.
- add products to a cart.
- remove product from a cart.
- view the cart.
- confirm an order.



Home:

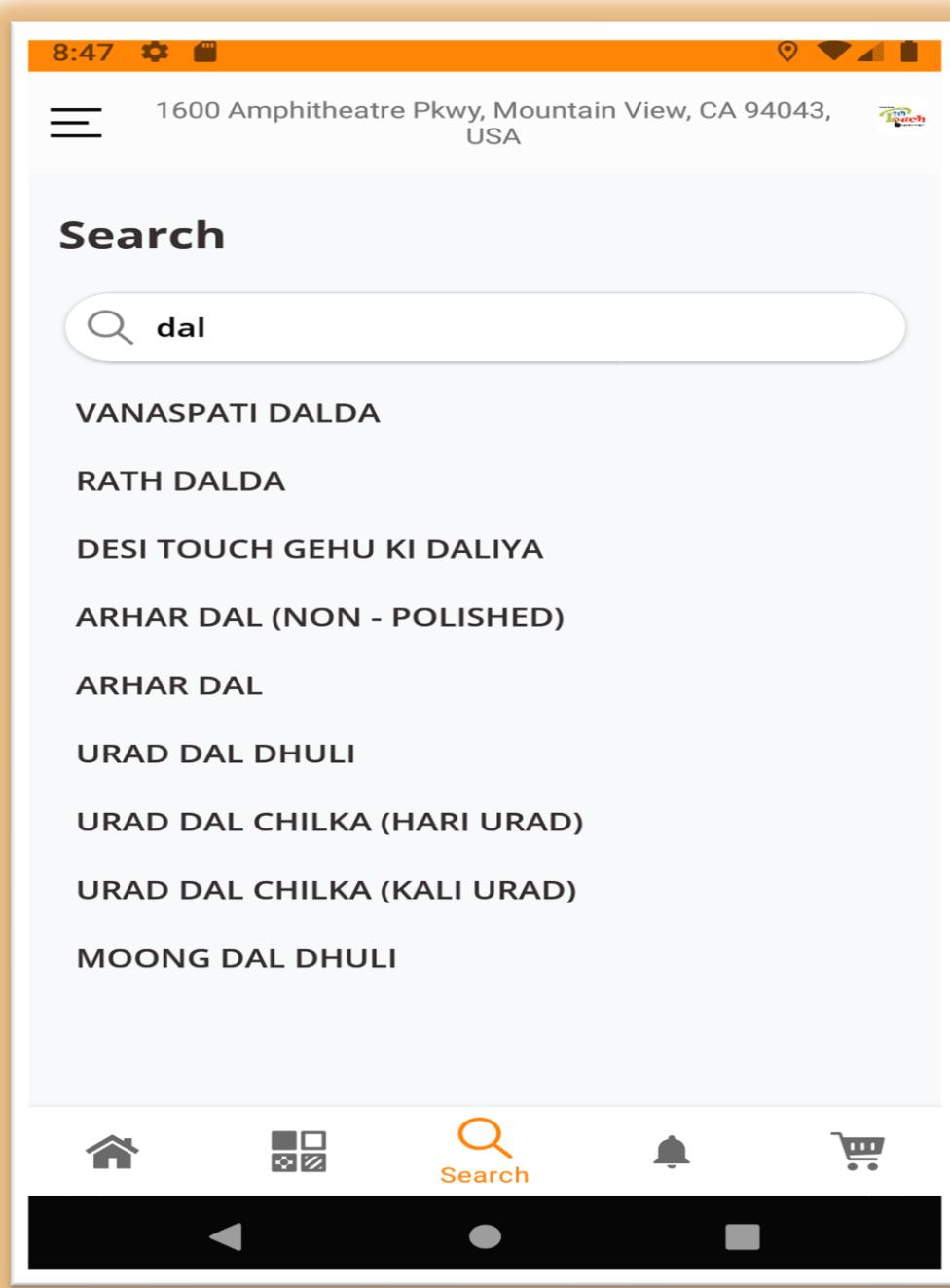
The Android application must have a home screen visible for all the user of the application (whether logged in or not). The home screen should contain the latest products added to the database.



Search:

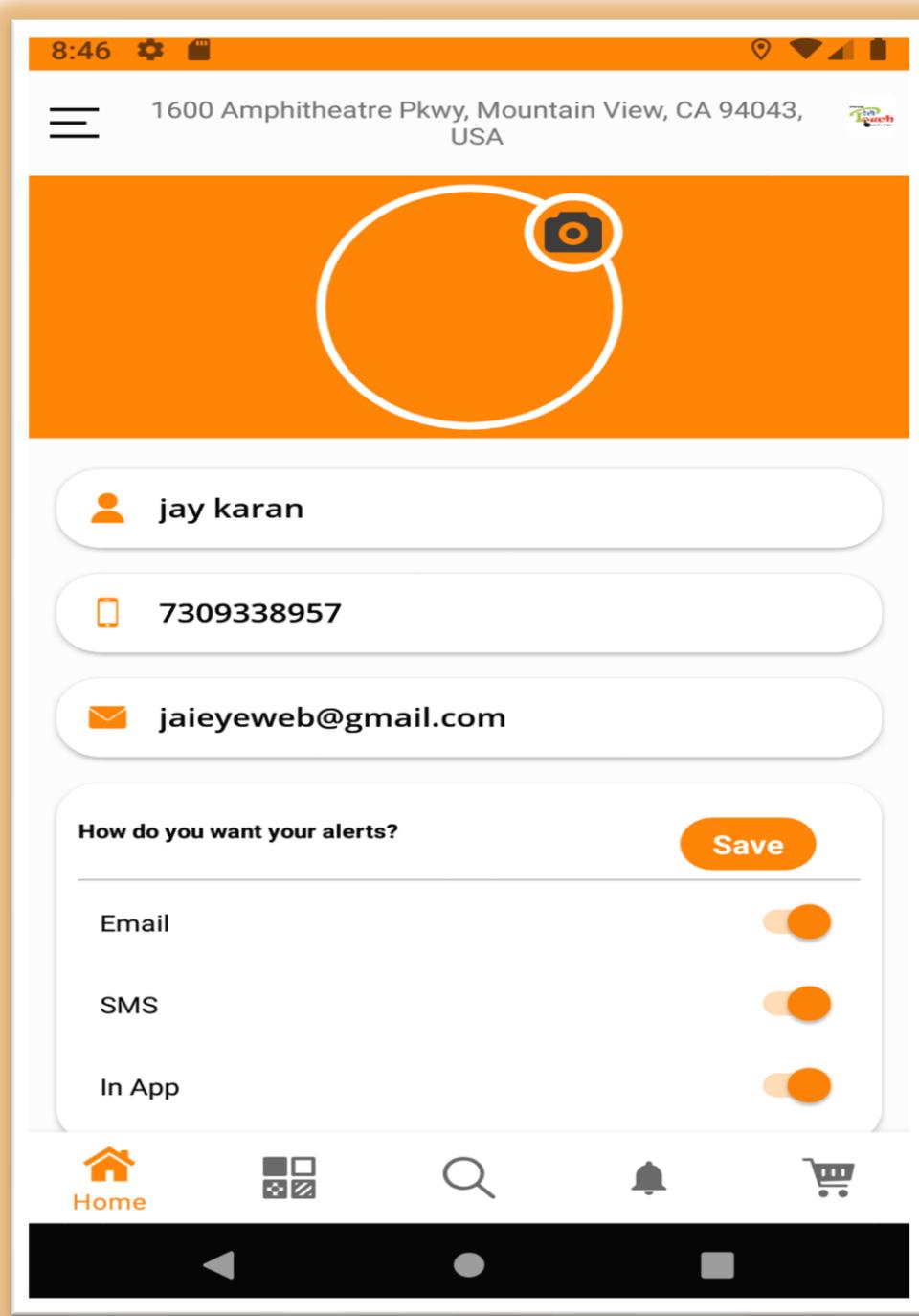
The Android application should have a screen for product search based on some criteria:

- Price
- Name
- Brand



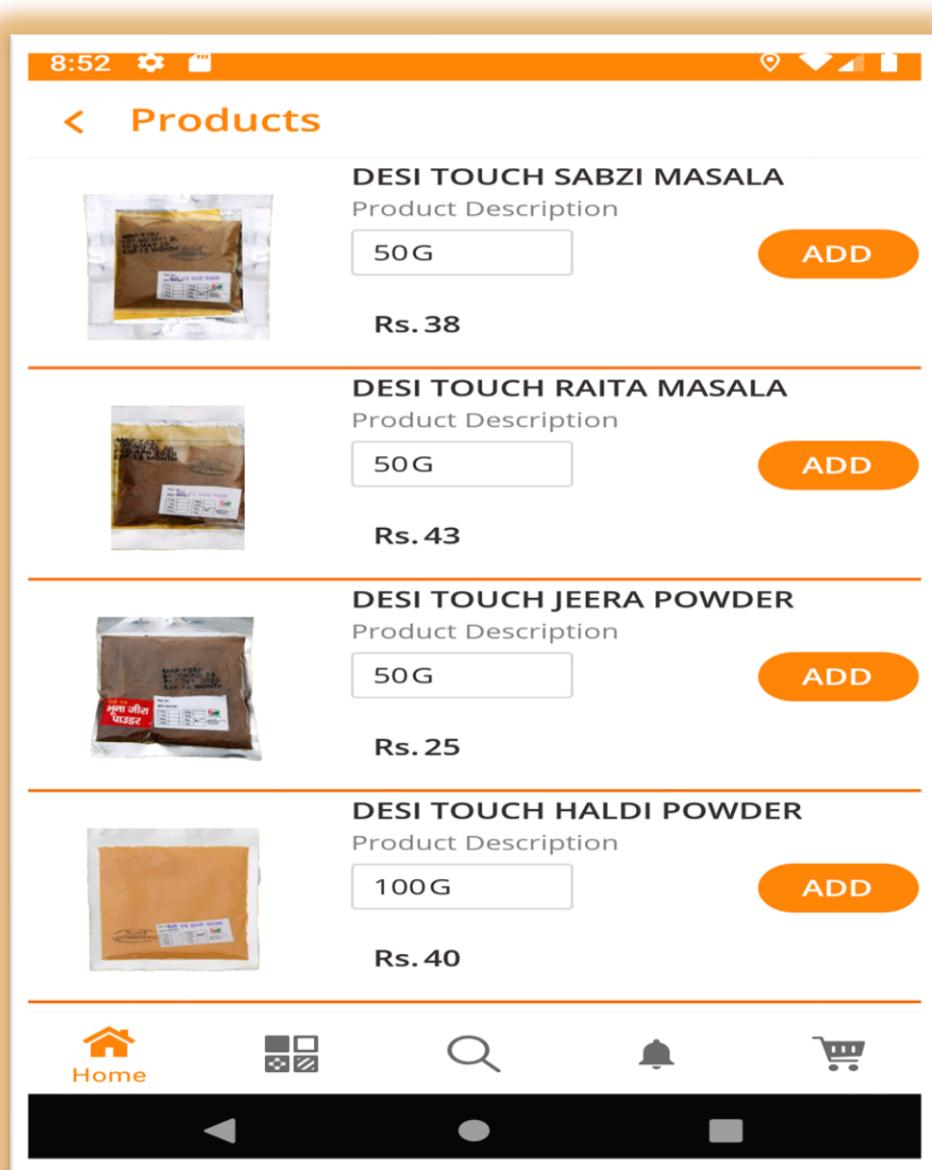
User Account:

The user must be able to modify his personal information and terminate his account if he has no pending orders.



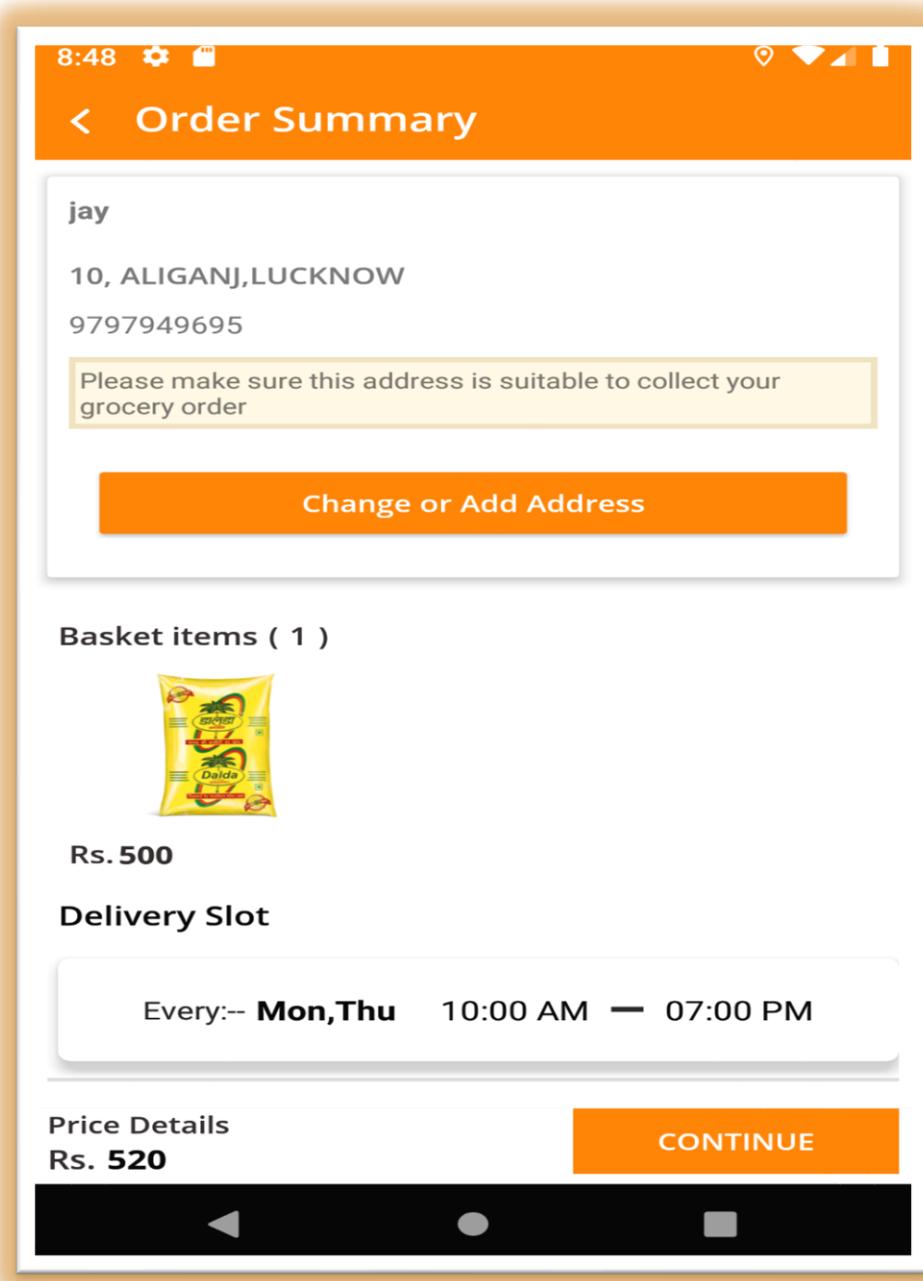
Products:

The user of the application must be able to view a product list available for sale. He must be able to view their details and add them to his cart if he is logged in.



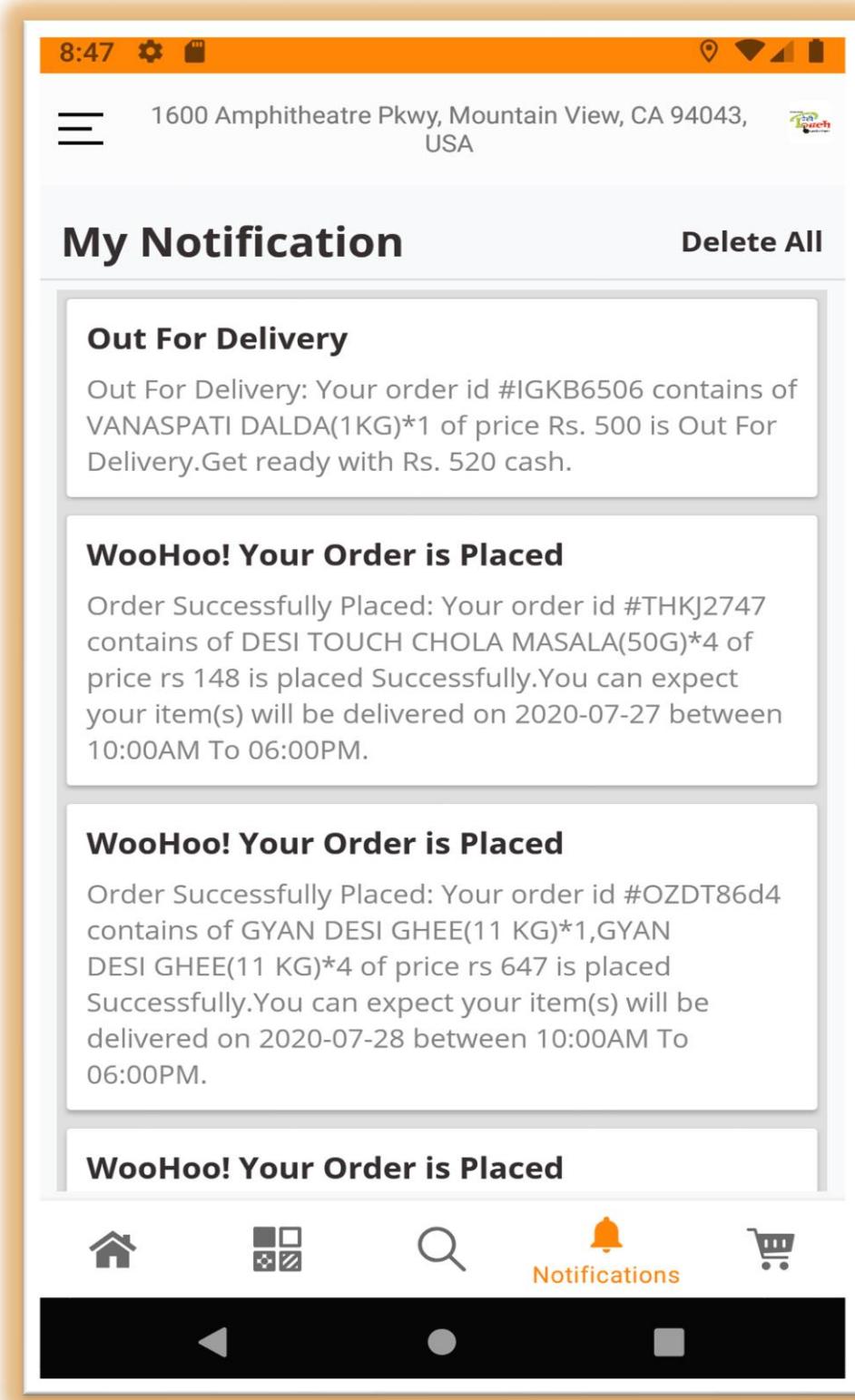
Orders:

After adding products to his cart, the user must be able to confirm his order and provide a valid address and a valid Android phone for the products to be shipped.



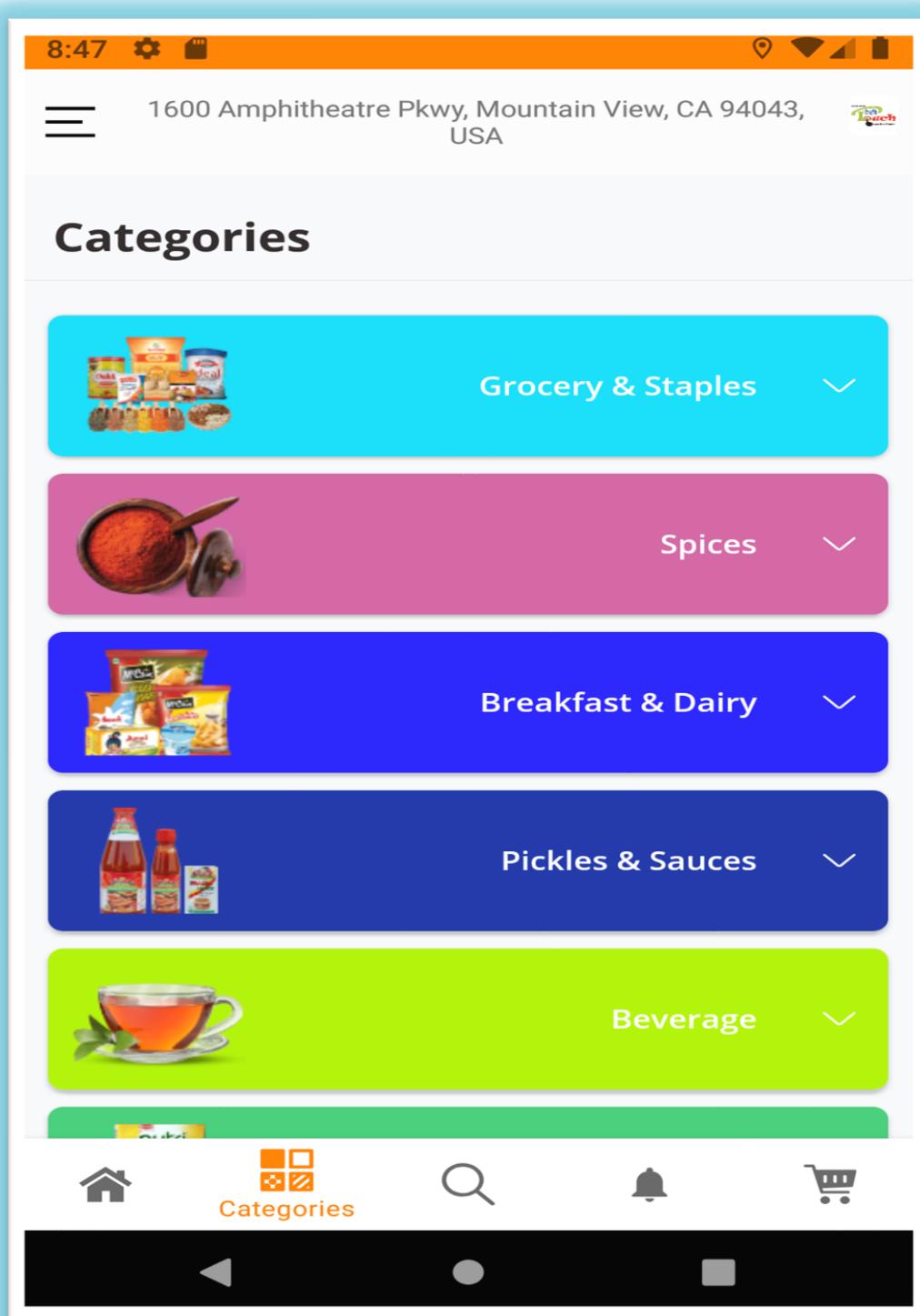
Push Notifications:

The Android application could have a push notifications functionality. It will be used to send promotions to users. It could also be used to inform users about new products.



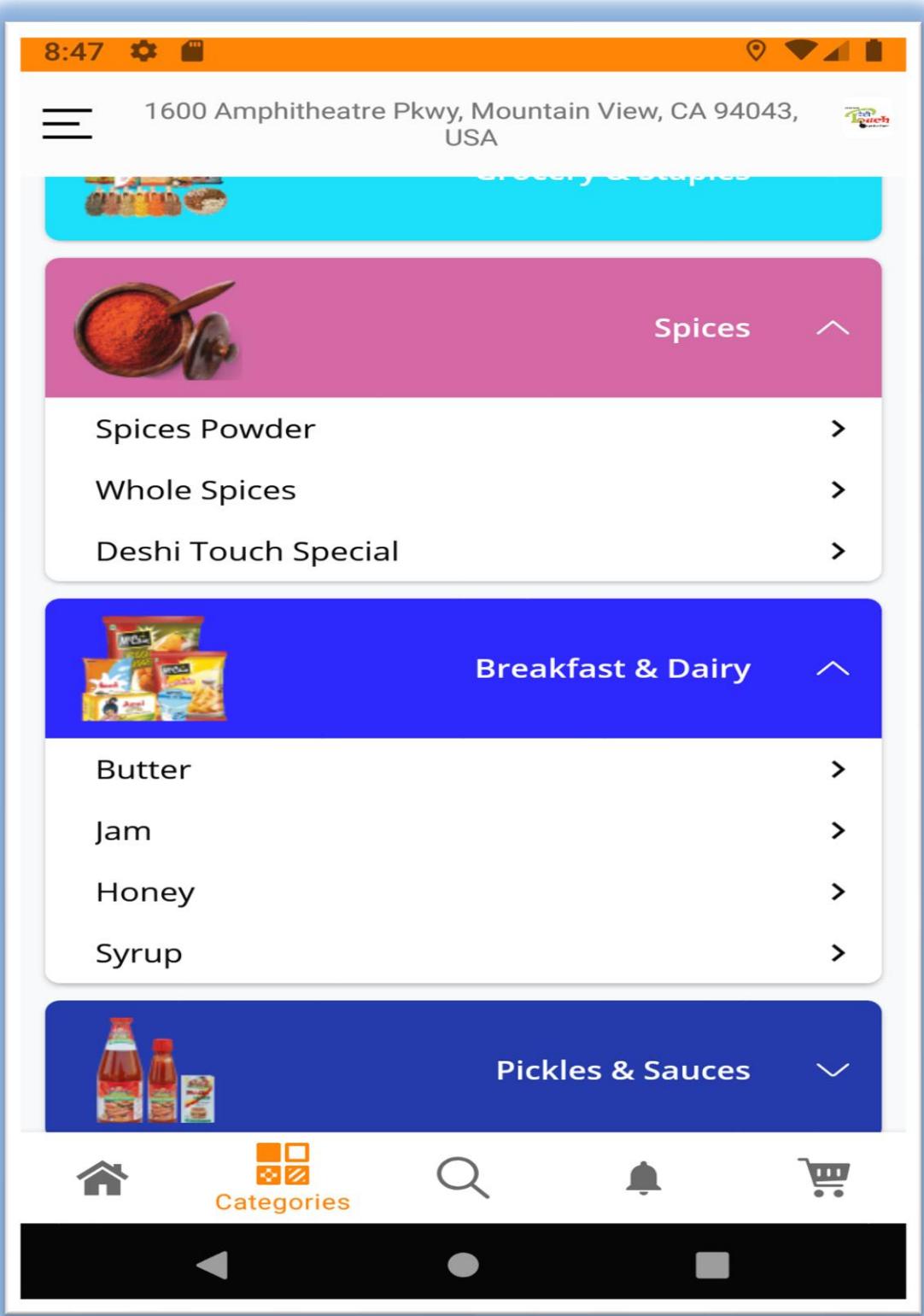
Category:

All Category list all product in this application



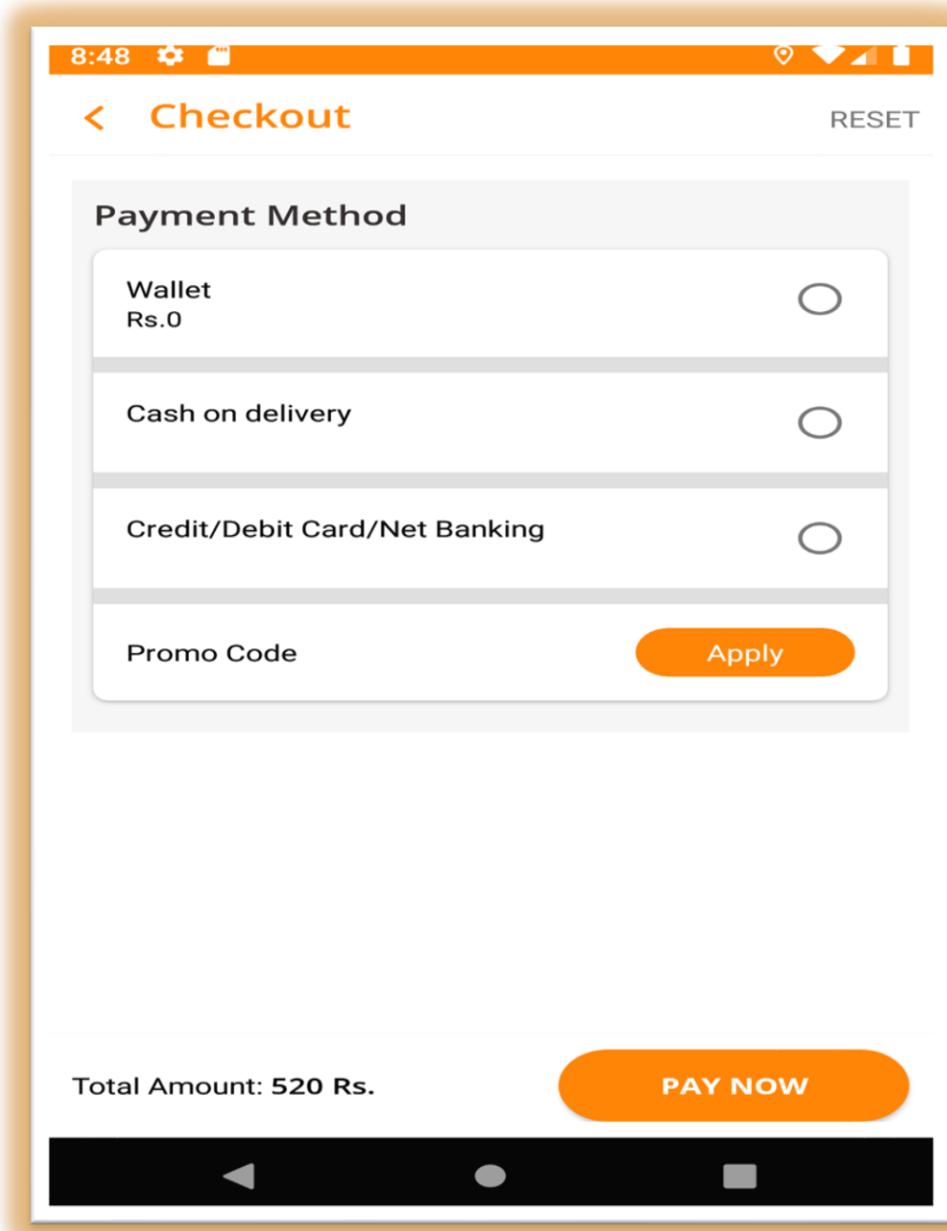
Category View Product:

All category view individual product

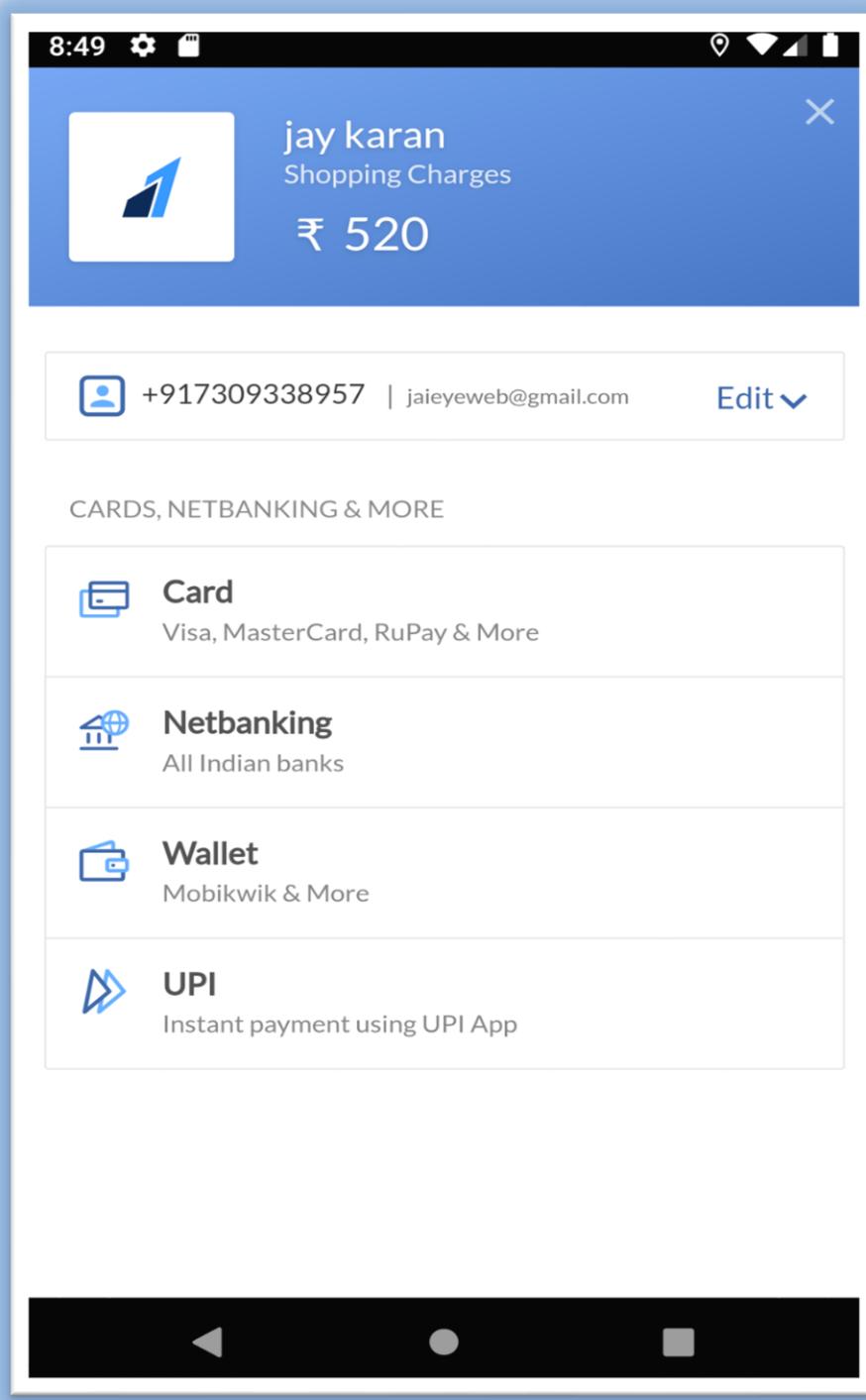


Payment:-

- 1. Cash On Delivery (COD)**
- 2. Online Payment (Net banking , UPI, Debit Card, Credit Card)**

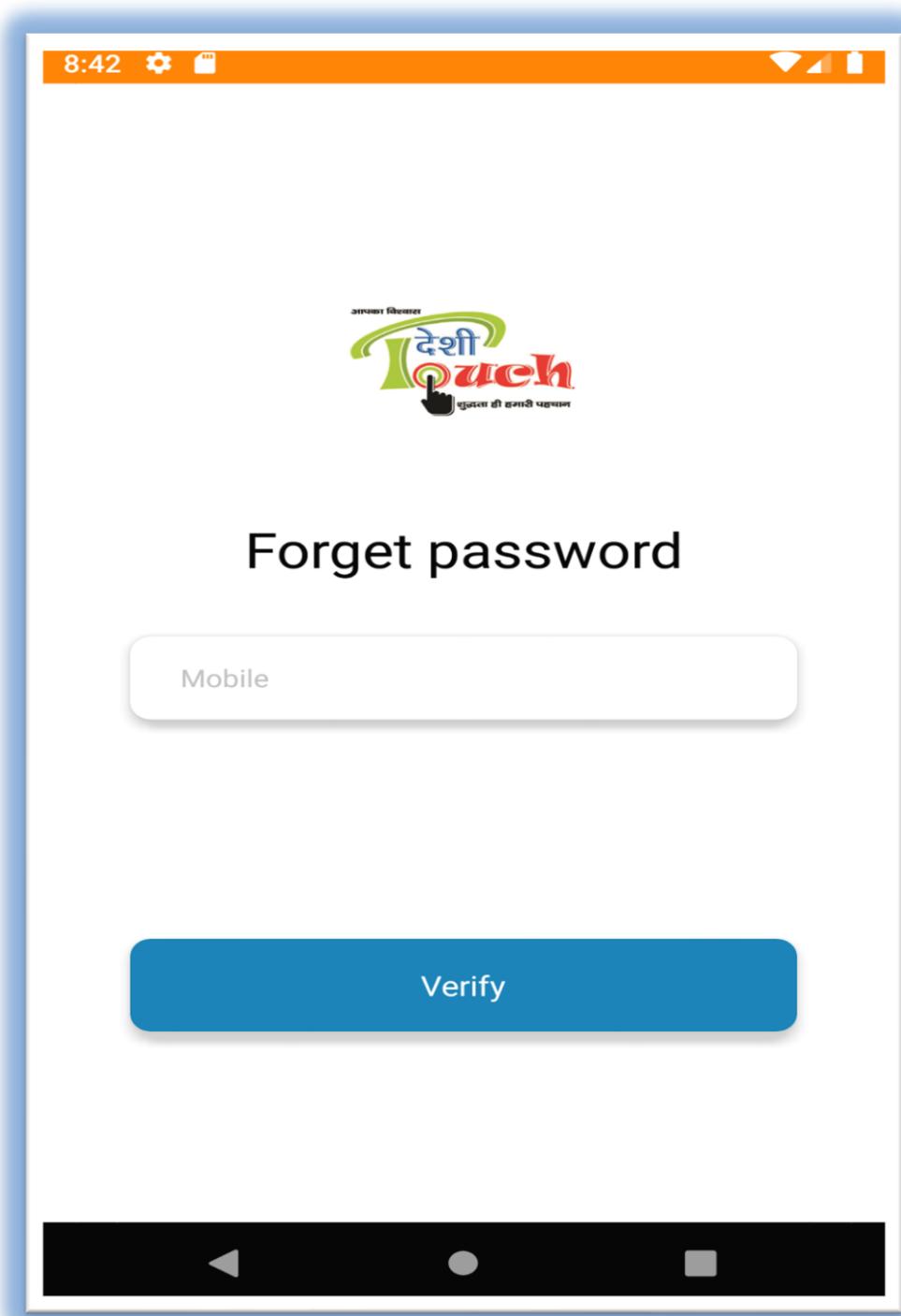


Online Payment:-



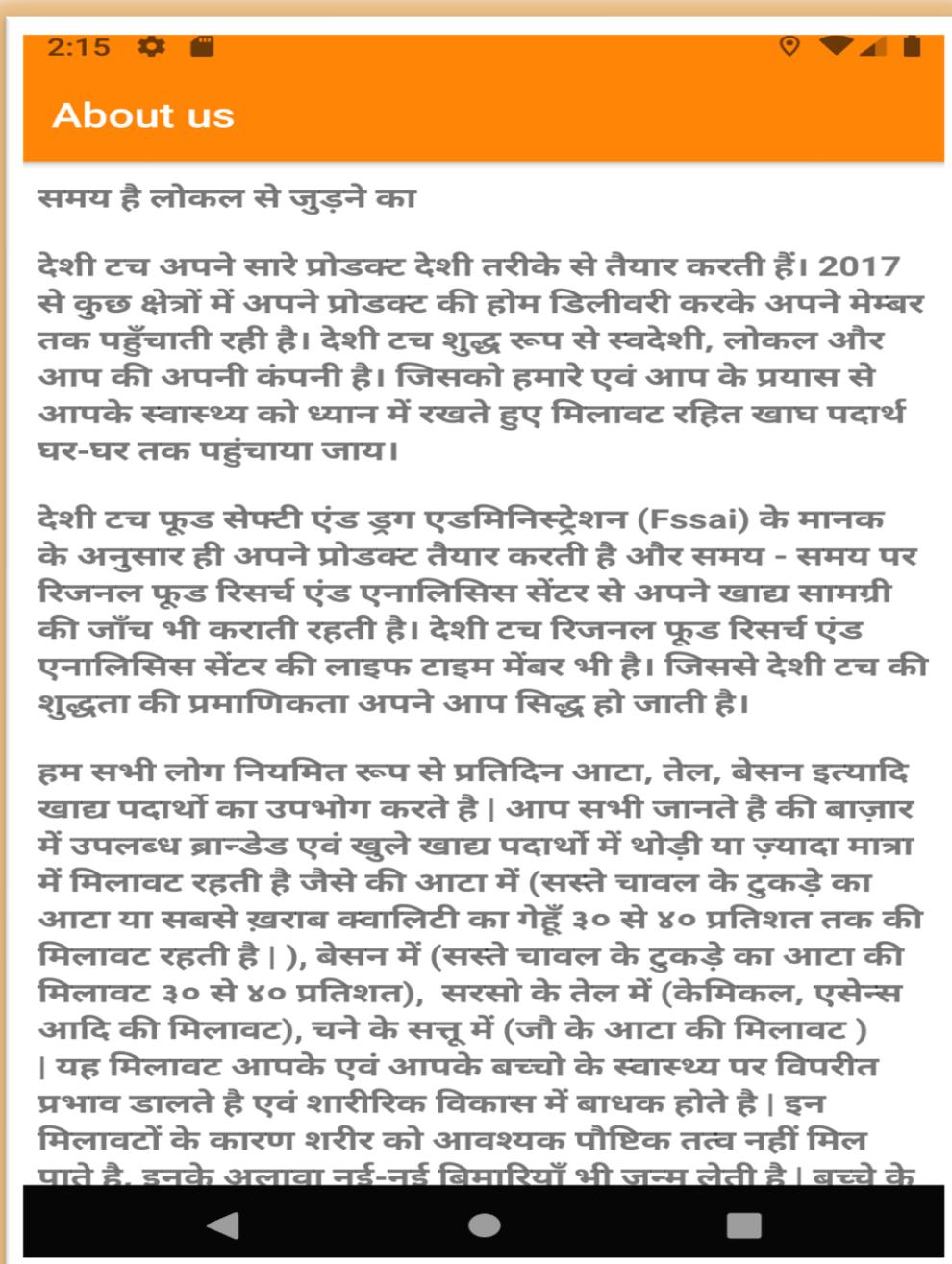
Forget Password:-

Login Password forget then reset password throw Android number OTP



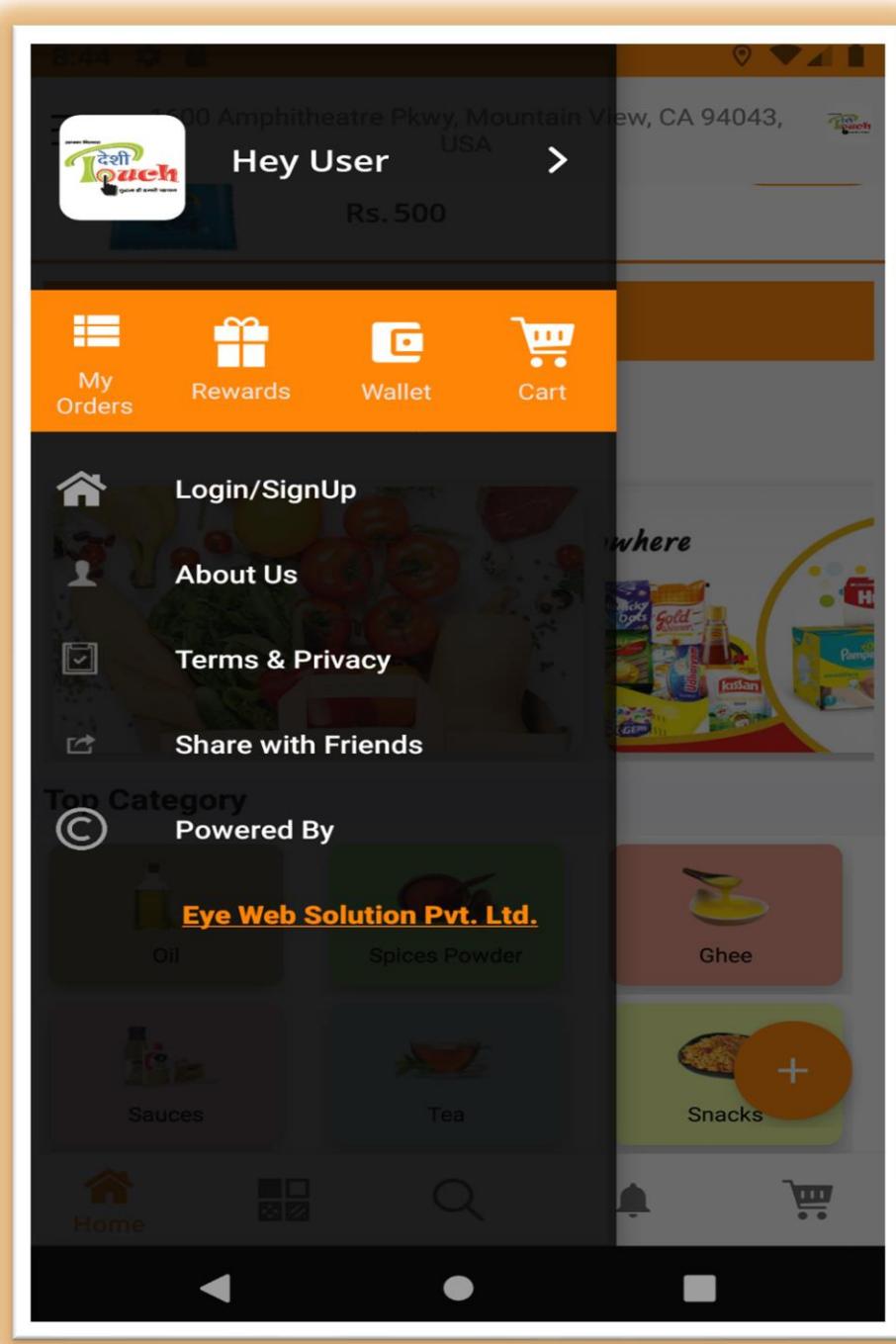
About:

The Android application should have a screen dedicated for general information about the company.



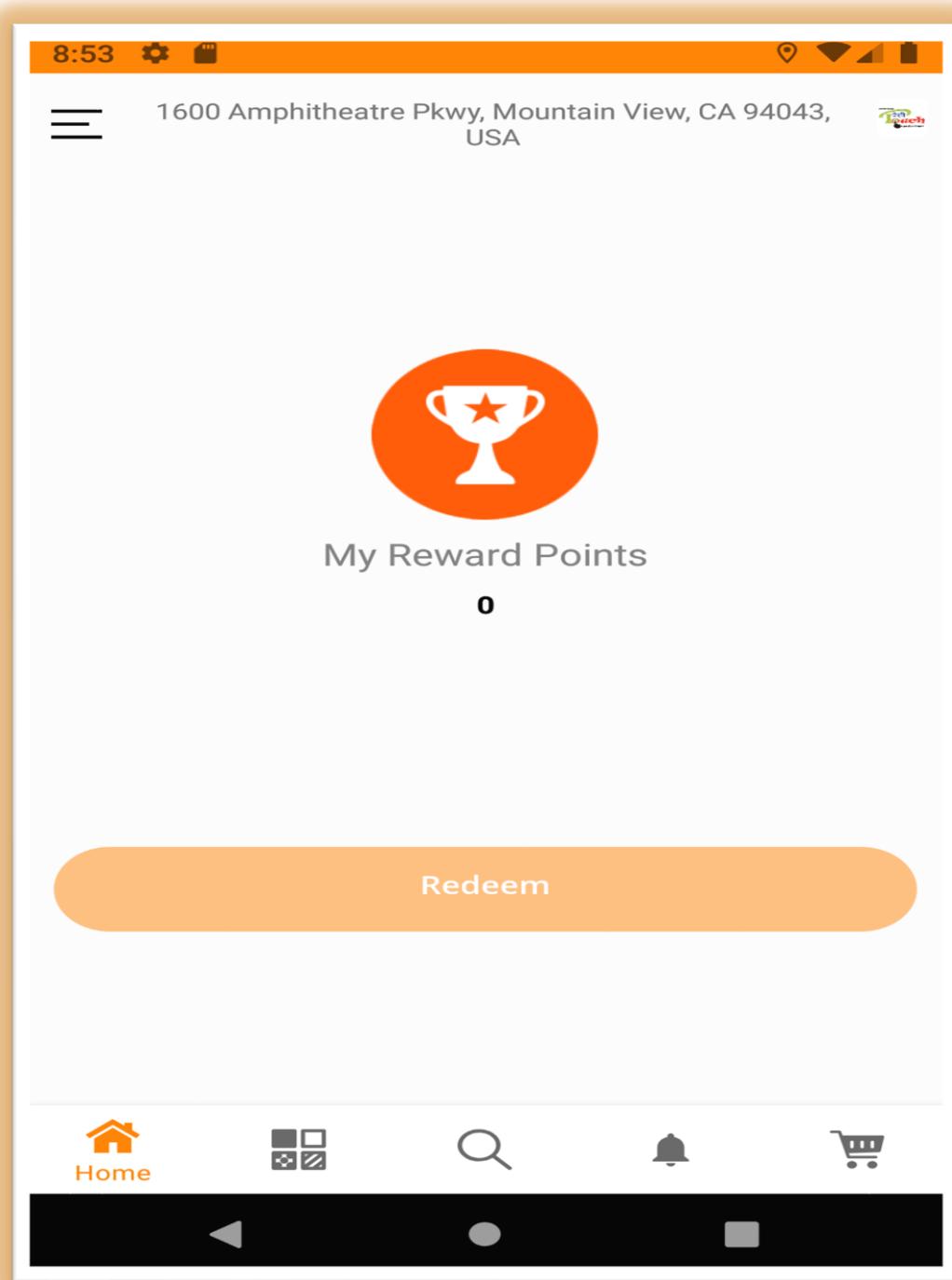
Navigation Drawer:-

Navigation Drawer in this application



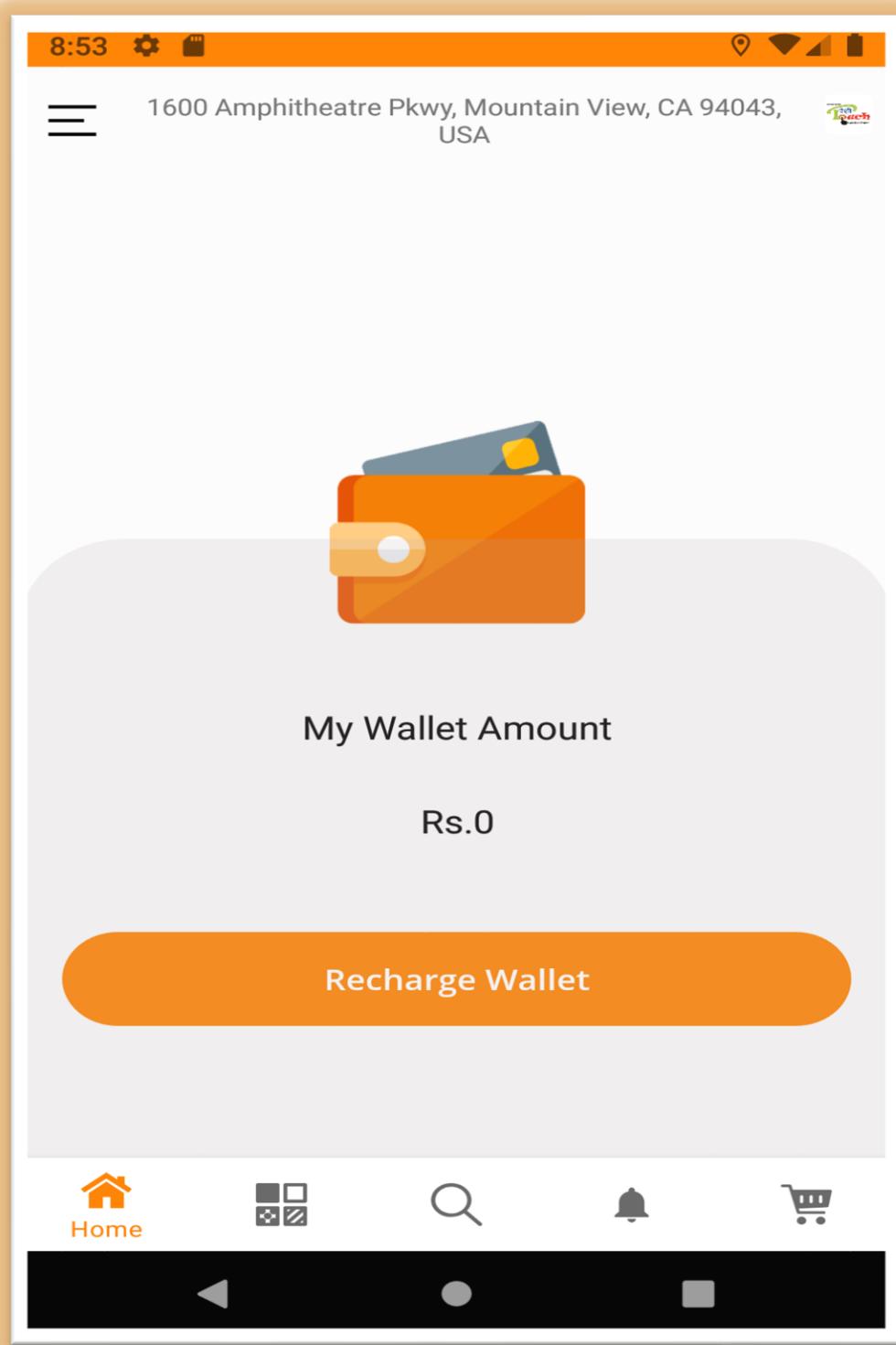
Reward Point:--

Reward point redeem any product purchase



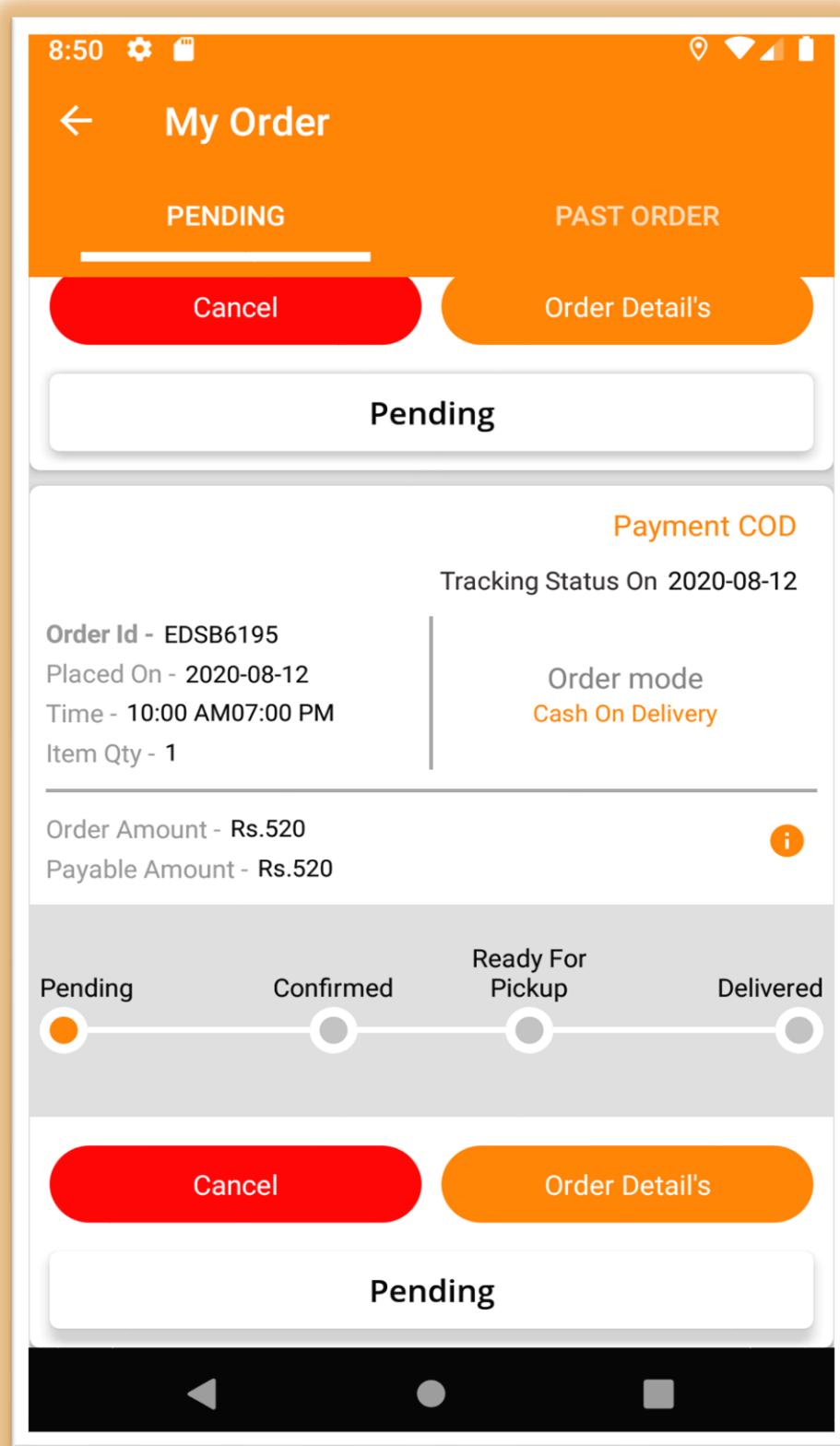
My Wallet:--

Wallet add money by Online payment



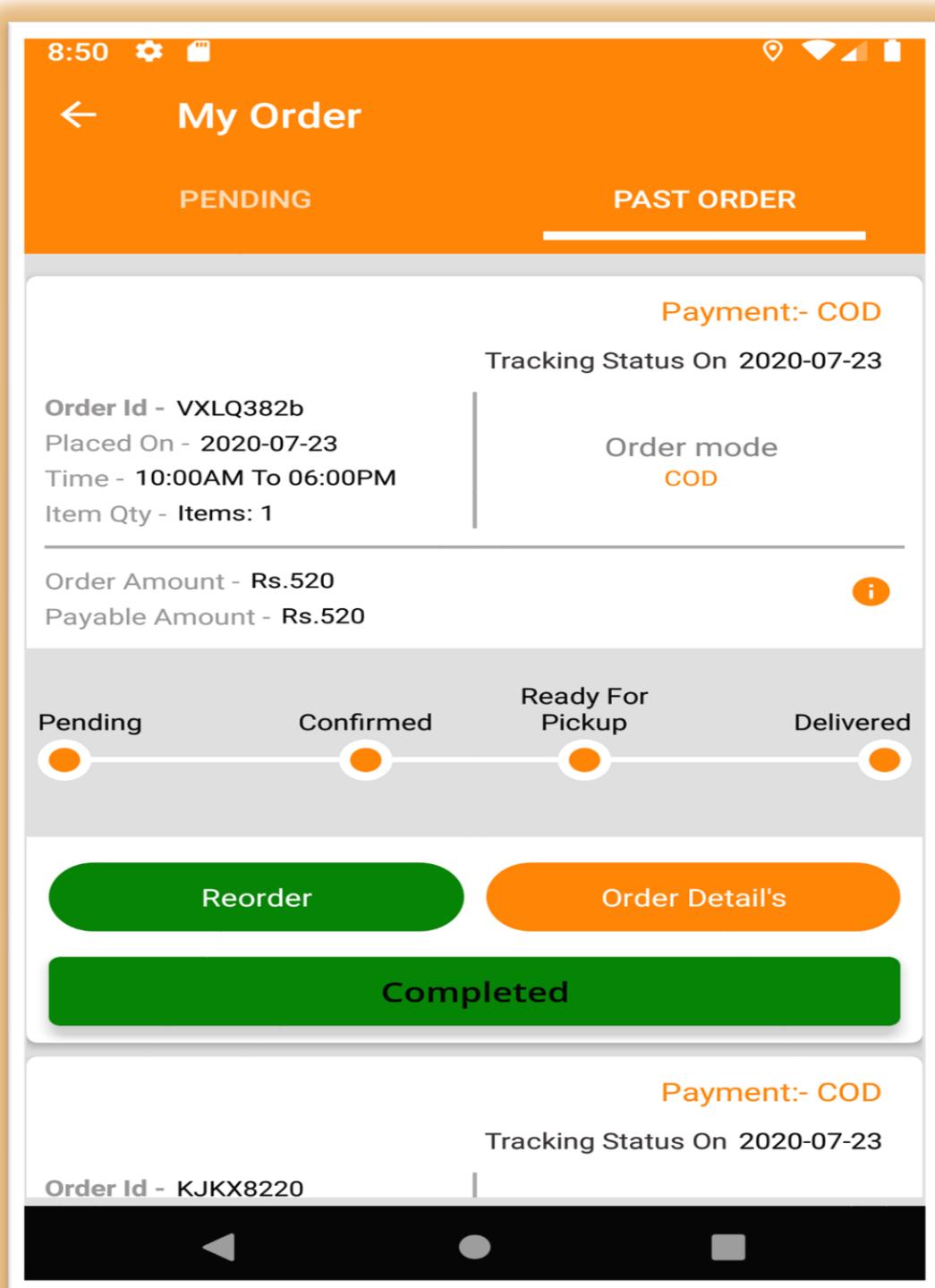
My Order Pending:--

Pending my order



Order Successful:--

Complete order deliver successfully



Desktop Application:

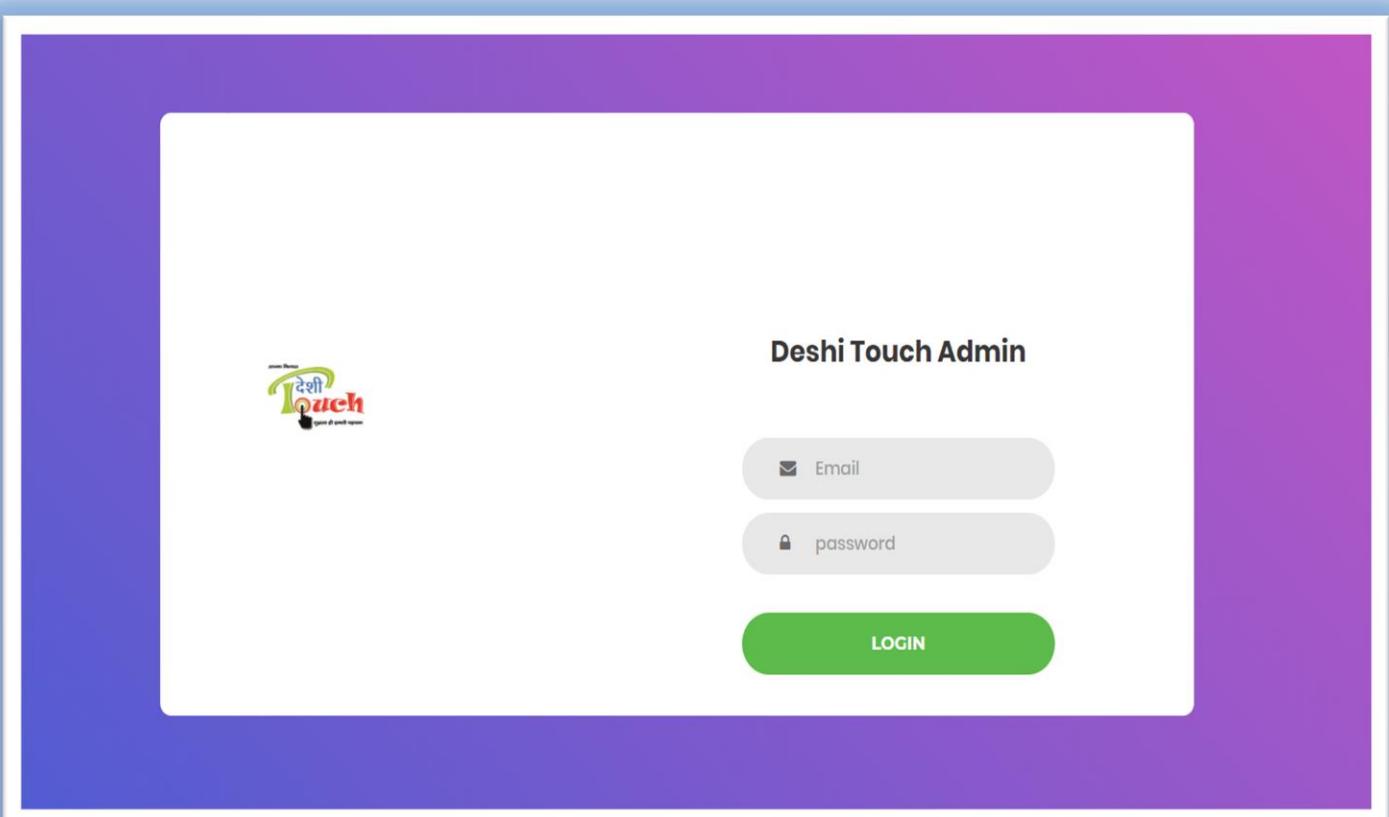
The desktop application (already implemented) is a database management system dedicated for the administrator to:

- Manage the accounts of users registered through the Android application or the web application.
- Manage the products that will be displayed in the web and Android application.
- Manage the orders made through the Android or web application.

It must have the following functional requirements:

Login:

The login frame that will allow the administrator of the system to login using a predefined login and password.



Dashboard Backend:--

The screenshot shows the DESHI TOUCH dashboard with a sidebar menu and a main dashboard area. The sidebar includes options like Dashboard, Send Notification, Settings, Category/Products, App Users, City/Area, Store Management, Orders, Payout Request/Validation, Reward, and Banner. The main dashboard displays several key statistics:

- Total Users: 8 users (Total App Users)
- Total Stores: 2 (Total Stores)
- Cities: 1 (Total Cities)
- Delivery Boys: 1 (Total Delivery Boys)
- Total Earning: 2080 (All Stores Earnings)
- Pending Orders: 48 (Total Pending Orders)
- Cancelled Orders: 5 (Total Cancelled Orders)
- Completed Orders: 4 (Completed Orders)

At the bottom right, it says "© 2020 made with ❤ by Deshi Touch".

Product Management System:

The product management system allows the admin to view a complete list of all the products offered. It will also allow the admin to search, add, update and view products. The search is done base on many criteria.

The screenshot shows a "Product List" page with a table titled "Products List". The table columns are: #, Product Name, Category, Product_Image, and Actions. The data in the table is as follows:

#	Product Name	Category	Product_Image	Actions
1	GYAN DESI GHEE	Ghee		
2	ANIK DESI GHEE	Ghee		
3	VANASPATI DALDA	Oil		
4	RATH DALDA	Oil		
5	FORTUNE REFINED	Oil		
6	SAFFOLA GOLD REFINED	Oil		

At the top right of the table, there is a "+ ADD PRODUCT" button.

All User Management List:

The management system allows the admin to view a clear list of orders made by the users of the web and Android application.

App Users						
#	User_name	User Phone	User Email	Registration Date	Is Verified	Active/Block
1	jk	9454785001	jk@gmail.com	2020-08-07 15:47:09	Not Verified	✓ ACTIVE
2	ffcf	7007580509	sj9670074765@gmail.com	2020-07-30 19:13:44	Not Verified	✓ ACTIVE
3	sadeep	8416841555	deliveryshopapp@gmail.com	2020-07-27 12:58:01	Not Verified	✓ ACTIVE
4	veer	8765530564	veereyeweb@gmail.com	2020-07-26 11:17:07	Not Verified	✓ ACTIVE
5	shiveyeweb	8127773333	shiveyeweb@gmail.com	2020-07-22 13:49:26	Verified	✓ ACTIVE
6	mahi	9935452222	mahieyeweb@gmail.com	2020-07-10 10:15:14	Verified	✓ ACTIVE
7	Ashraf Ali	8565900597	ashrafeyeweb@gmail.com	2020-07-09 17:42:32	Verified	✓ ACTIVE
8	jay karan	7309338957	jaieyeweb@gmail.com	2020-07-09 17:26:58	Verified	✓ ACTIVE

City List:--

All delivery address city list insert, update and delete

The screenshot shows a table titled "City List" with three rows. The columns are labeled "#", "City Name", and "Actions". The entries are:

#	City Name	Actions
1	LUCKNOW	
2	KANPUR	
3	GORAKHPUR	

Area List:--

All delivery address area list insert, update and delete

The screenshot shows a table titled "Area List" with nine rows. The columns are labeled "#", "Society Name", "City Name", and "Actions". The entries are:

#	Society Name	City Name	Actions
1	ALIGANJ	LUCKNOW	
2	JANKIPURAM	LUCKNOW	
3	RAJAJIPURAM	LUCKNOW	
4	INDIRA NAGAR	LUCKNOW	
5	NIRALA NAGAR	LUCKNOW	
6	VIKAS NAGAR	LUCKNOW	
7	MAHANAGAR	LUCKNOW	
8	NEW HYDERABAD	LUCKNOW	
9	GOKHLE MARG	LUCKNOW	

Product Category List:-

All category product list insert and update and delete

Category List				
#	Cat Name	Parent Category	Cat image	Actions
1	Grocery & Staples	-----		 
2	Spices	-----		 
3	Breakfast & Dairy	-----		 
4	Pickles & Sauces	-----		 
5	Beverage	-----		 
6	Chocolate & Biscuit	-----		 

Order List:- Pending Order list

Pending Orders					
#	Cart_id	Cart price	User	Delivery_Date	Cart Products
1	EDSB6195	520	jay karan (7309338957)	2020-08-12	DETAILS PRINT
2	UPQQ8200	558	Ashraf Ali (8565900597)	2020-08-10	DETAILS PRINT
3	GQFE241d	58	jay karan (7309338957)	2020-08-10	DETAILS PRINT
4	YHOW2984	58	jay karan (7309338957)	2020-08-10	DETAILS PRINT
5	ZFT06260	520	jay karan (7309338957)	2020-08-07	DETAILS PRINT
6	FGXC07b1	5560	mahi (9935452222)	2020-08-07	DETAILS PRINT
7	CYRK756b	520	jay karan (7309338957)	2020-08-07	DETAILS PRINT

Order Complete:-

Some of these functionalities of the desktop applications are already implemented in the previous work on the project. However, some changes/updates are to be made in order to adapt it to Android application that is going to be developed

Completed Orders						
#	Cart_id	Cart price	User	Store	Delivery boy	Delivery_Date
1	KJKX8220	520	jay karan (7309338957)	DT ALIGANJ (0002)	Anas (9161373746)	2020-07-23
2	CXFS9662	520	jay karan (7309338957)	DT ALIGANJ (0002)	Anas (9161373746)	2020-07-23
3	LFSN2660	520	jay karan (7309338957)	DT ALIGANJ (0002)	Anas (9161373746)	2020-07-23

4.2 Non-Functional Requirements:

Product requirements

Availability Requirements:

The system must be always available for use. Except for special cases (in case of backup), it will be announced beforehand through push notifications.

Usability requirements:

The final Android application will be entirely in French, because it is targeting the Indian market. It will be user-friendly and very easy to use.

Performance requirements:

The Android application will be developed using Android Studio which will, thanks to the good algorithms that will be used, guarantee a high execution speed and a minimized response time.

Scalability requirements:

The expected number of users of the Android application is very big. It is expected that the number of users grows on within the next few years. Therefore, the application must be highly scalable. We will achieve that thanks to some MySQL solutions such as (MySQL cluster or MySQL replication)

Maintainability requirements:

The application should be maintainable in order to allow upgrades (through the Google Play Store) in the future.

Extensibility requirements:

The Android application should be extensible. It should allow updating it and adding new features in the future.

Security requirements:

The application must be very secure because it deals with the private information of the users. This should be performed using the right encryption of data only accessed by the administrator. More than that, the system must follow

these main security rules:

- Confidentiality: Only the admin has access to user personal info and orders.
- Integrity: Only the users can modify their personal info.
- Authenticity: No one can access, modify or delete other accounts' information.

5 FEASIBILITY STUDY

The feasibility study phase is a very important phase, because it is the one that helps the software engineer to see the negative sides of his project and it helps him to refine the software process. In fact, the feasibility study can make any project a successful one if it is done in a correct way. This deliverable is the feasibility study for the project I will be working on my capstone: “Grocery Store ”

Product feasibility:

The system that will be developed will help the client achieve their objectives. Which are expanding their business and open up to the online market. This will be achieved thanks to the Android application that will make the access to the company's products, order them and communicate with the company extremely easy. Talking about communication, the Android application with the “push” feature will allow the administrator to send notifications to the users' phone about new products, promotions, or just advertising the store's products.

Technical and Operational Feasibility:

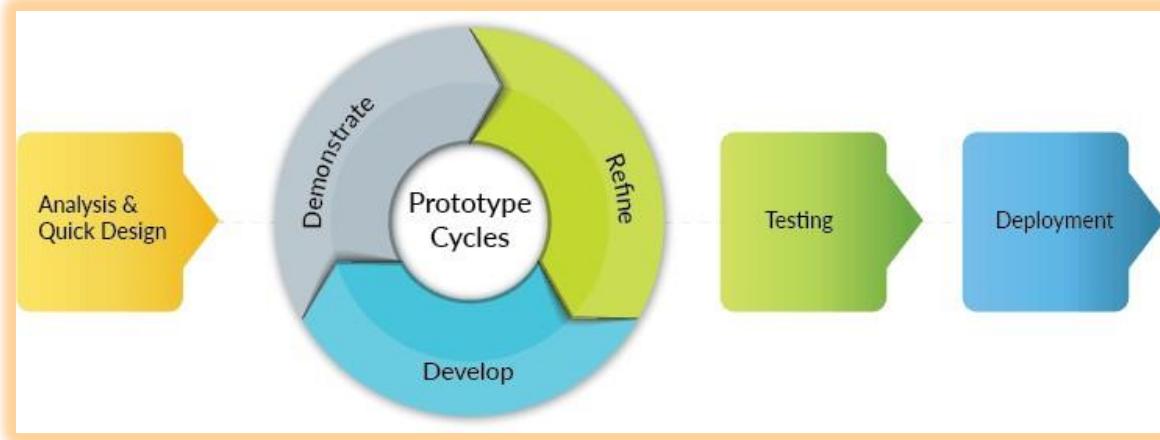
For this project, I will be using Java & XML for the Android application using Android Studio as it is the official IDE for android applications' developing. For the backend, I will be using Firebase as a database engine; as it is known for its high-reliability, full-featured, and self-contained. More than that, it is the most suitable database engine to use with an android application.

Social & Market Feasibility:

Since smart phones are being used by almost everybody, I believe that a Android application for our client “the Grocery store” will be of great benefit. Also, since there are no similar applications for other Grocery stores, this project will be one of a kind and it is expected to become the leader of Grocery stores Android applications’ .

After approaching the client « Deshi Touch Grocery » for the how they want their product to be developed and what are the minimum functions the product must perform; I decided that the project is feasible to develop and to be materialized in terms of “implementation, contribution of project to organization, cost constraints, and as per values and objectives of the organization.”

6 METHODOLOGY



Rapid Application Development Model

Concerning the system development methodology, I opted for the RAD (Rapid Application Development). Since it allows fast development and delivery of a high quality system, it was the perfect methodology for my capstone project. In fact, I had permanent contact with the client throughout the different stages of software process. I developed each part of the project alone and had the client (Deshi Touch Grocery) test it and give me feedback about what should be changed or improved. This interaction saved me a lot of time, because the feedback was instantaneous. Accordingly, the user was involved in the design. Also, since the project is an , the priority is to fulfill the business need of the company; which goes hand to hand with the main principles of the Rapid Application Development methodology. Moreover, since it is a capstone project, we have deadlines and we need to meet them. This among the basic principles of RAD. The presence of deadlines or “timeboxes” .

The main strengths of the RAD model:

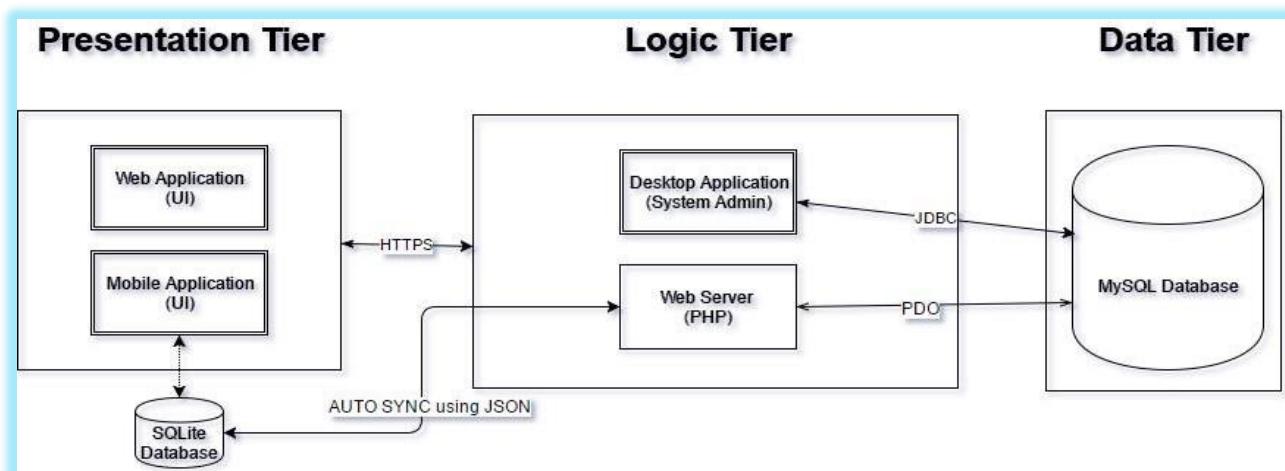
- “The operational version of an application is available much earlier than with Waterfall, Incremental, or Spiral frameworks.”
- “Because RAD produces systems more quickly and to a business focus, this approach tends to produce systems at a lower cost”

7 SOFTWARE ARCHITECTURE

The system architecture I opted for in the project is 3-tier application architecture. Because of the four main benefits it has; that most of them happen to be on the list of the non-functional requirements of the project:

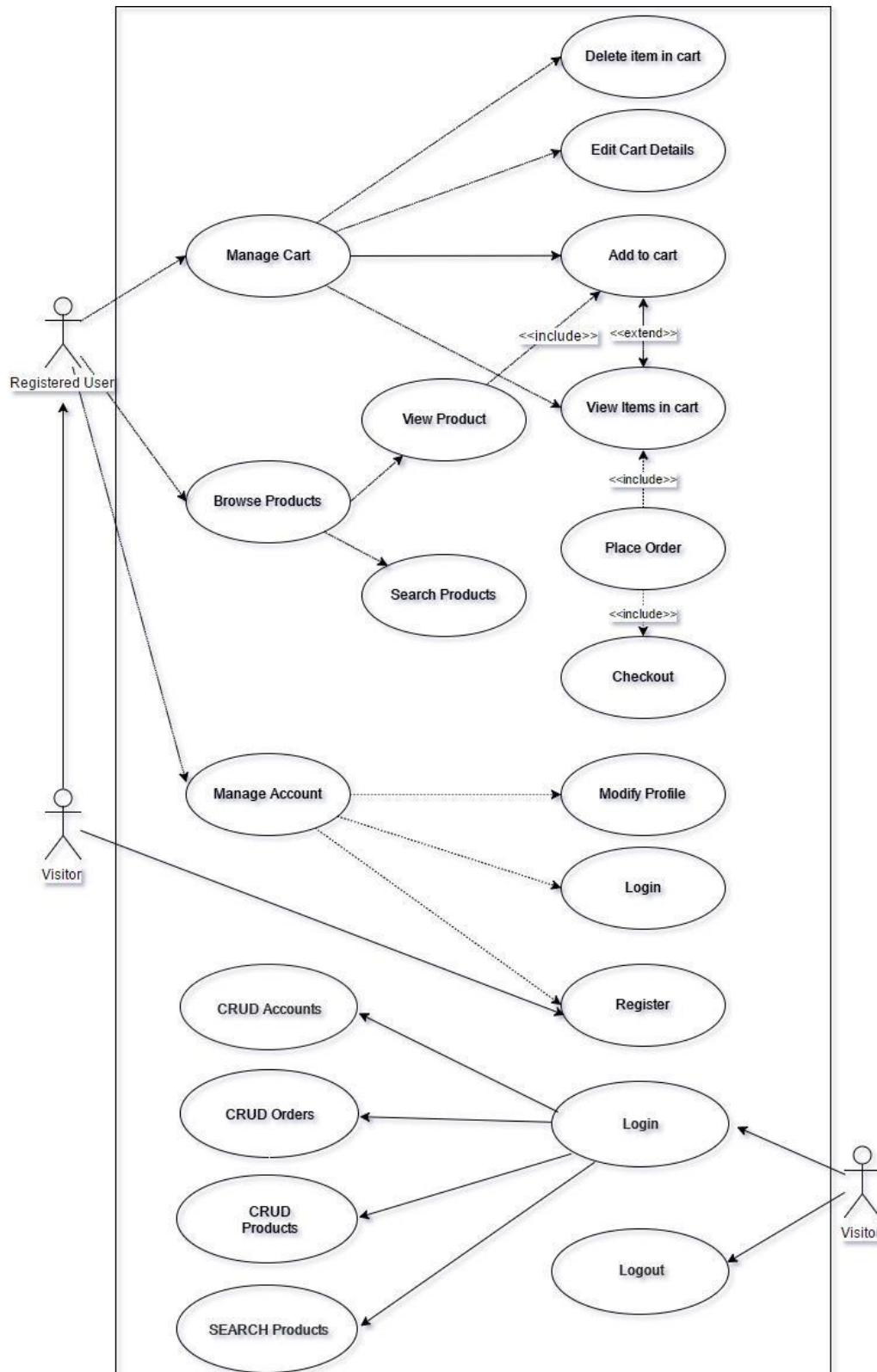
- “Maintainability. Because each tier is independent of the other tiers, updates or changes can be carried out without affecting the application as a whole.”
- “Scalability. Because tiers are based on the deployment of layers, scaling out an application is reasonably straightforward.”
- “Flexibility. Because each tier can be managed or scaled independently, flexibility is increased.”
- “Availability. Applications can exploit the modular architecture of enabling systems using easily scalable components, which increases availability.”

On the other hand, the system architecture I opted for is similar to the RESTful API architecture. The client connects to the web application or the Android application in the presentation tier and makes orders. The orders are processed in the logic tier by the admin and then stored in the data tier. There is also an auto-synchronization of the Firebase database with MySQL one to add new products to the Android application.



8 DESIGN

8.1 Use Case Diagram:



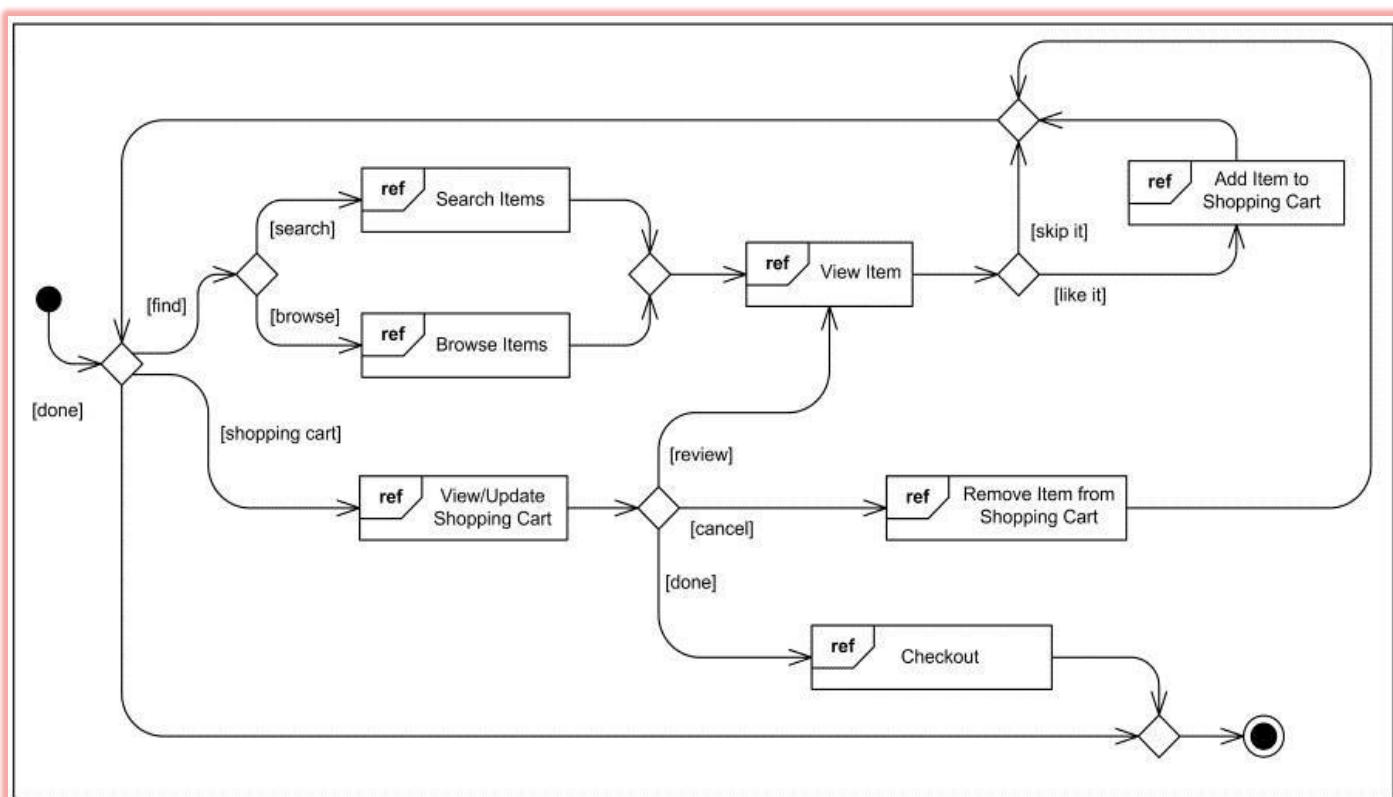
The diagram above is a use case diagram which represents the Android user interaction with the system of the Android application as well as the administrator's interaction with the desktop application. The diagram shows, depending on the user of the system, the different actions and relationship with each other. CRUD is the acronym for: Create, Read, Update, Delete, which are the 4 basic operations performed by the administrator of the system on its different components. The actors are:

- Administrator: Creates, Reads, Updates, and deletes accounts, orders, and products.
- Visitor: Registers to be able to search, view, and order products.
- Registered User: He needs to login to be able to create a cart and add products to it.

Then he can place an order. He can also modify his profile.

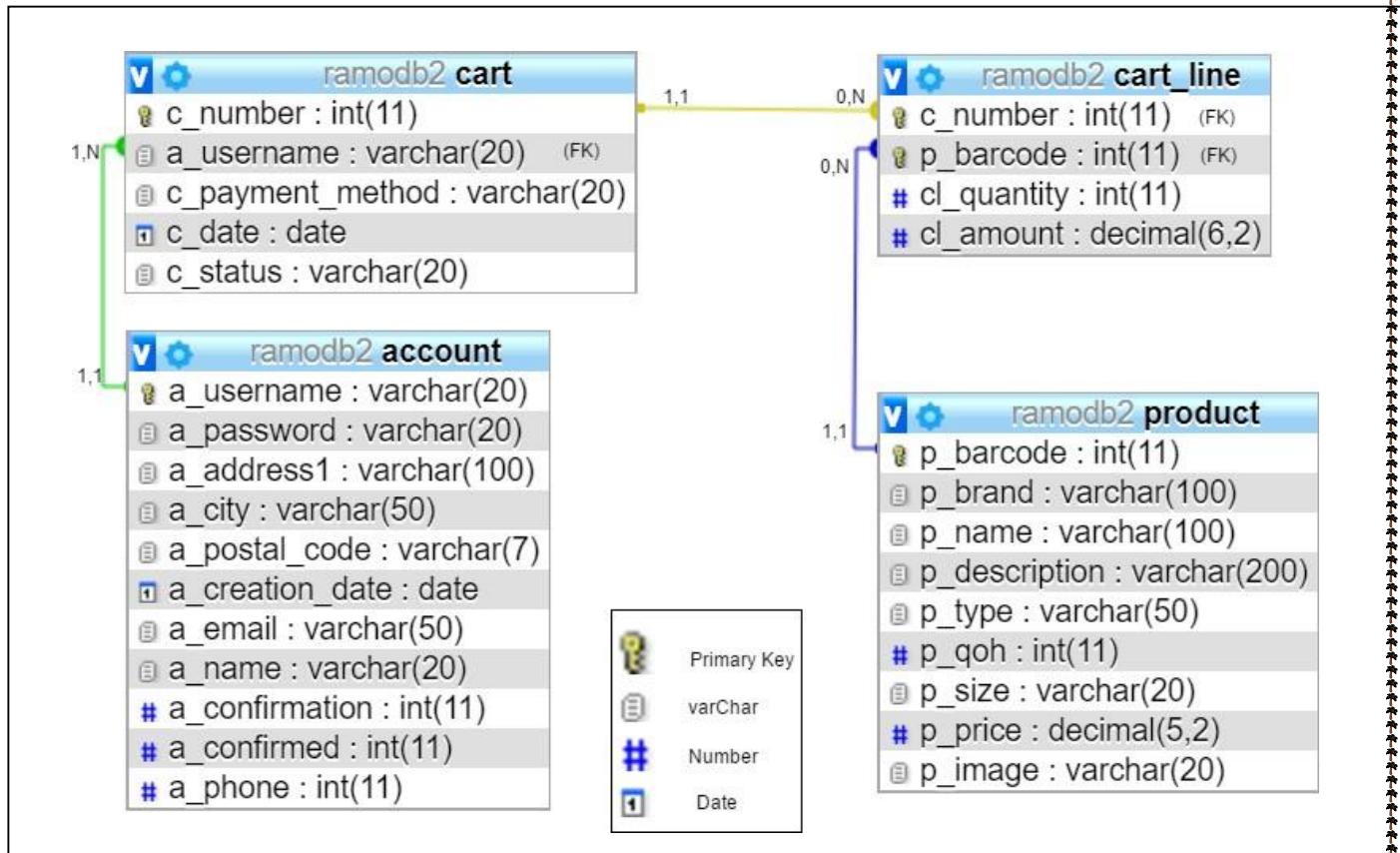
8.2 Interaction Overview Diagram

“Interaction overview diagrams provide overview of the flow of control where nodes of the flow are interactions or interaction uses. Interaction overview diagrams do look like activity diagrams that can only have inline interactions or



8.3 Data Model

Entity Relationship Diagram (ERD):



After applying the tools we learned in the Database Systems class, each table has its primary key and there is no transition relationship or a dependency relationship so in this case we can conclude that these tables are all in 3NF and that makes our implementation easier.

- **Cart** is an important entity involved in our business. This entity is linked to account entity and cart line entity. One cart can be used in one account and one account can use only one card. Moreover, Cart can have many cart lines while a cart line can have only one cart as a resource.
- **Cart line** is another important entity involved in our business. This entity is linked to entity cart which is described previously and Product entity. In fact, a cart line can have one and only one product while products can be in many cart lines.
- **Account** represents the accounts of the customers. It also holds a relationship with the table Cart, which is a unique feature of the Android

application that allows the customer to buy multiple different products online with different quantities.

- **Product:** This is one of the central and heaviest tables of our database. It contains a list of all the products (available or not) that have been, are, or will be provided by the store. It has a relationship with cart_line.

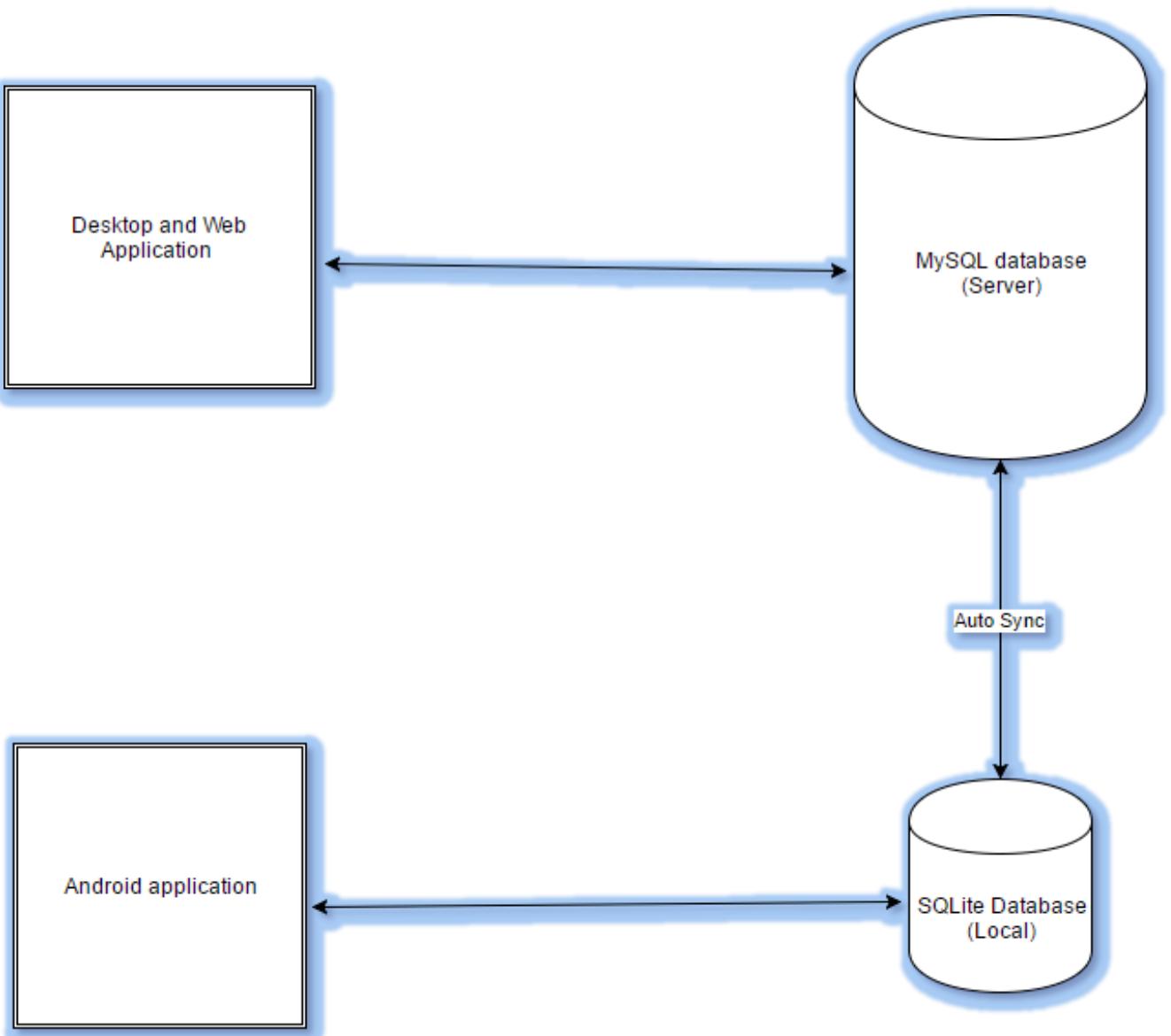
Table	Column	Data Type	Column Description	Table Description
Account	a_username	String	The username is the primary key of the table and is chosen by the user.	The account table will store the information relevant to the user of the Android application (and the web application). It stores the user's authentication info as well as his personal information. The account/user will be able to create a cart and add cart lines to it. The a_username column will be a foreign key of the cart table.
	a_password	String	The password is chosen by the user and is used for authentication. It is encrypted using MD5 hash algorithm.	
	a_name	String	The full name of the user.	
	a_phone	String	The phone number of the user.	
	a_email	String	The email of the user, which will be confirmed using PHPMailer.	
	a_address	String	The address of the user.	
	a_city	String	The city of the user.	
	a_postal_code	String	The postal code of the user.	
	a_confirmation	Integer	This column indicates whether the confirmation email is sent.	
	a_confirmed	Integer	This column indicates if the email is confirmed.	
	a_creation_date	Date	This column stores the creation date of the account and is automatically generated	
Product	p_barcode	Integer	The barcode of the product, it is the primary key of the table and it is auto-incremented.	The product table stores the product information, its barcode and a link to an image of the product.
	p_brand	String	The brand of the product.	
	p_name	String	The name of the product.	
	p_description	String	The description of the product.	
	p_type	String	The type/category of the product.	
	p_qoh	Integer	The quantity on hand (in the stock) of that product.	
	p_size	String	The size of the product.	
	p_price	Double	The price of the product.	
	p_image	String	The link of the image of the product.	

Cart	c_number	Integer	The cart number, it is the primary key of the table and it is auto-incremented.	The cart table stores general information about a cart. We can consider this table as an "order" table, because the admin of the system will receive the carts as orders.
	a_username	String	The username of the user who created the Cart, it is a foreign key.	
	c_date	Date	The date the cart was created.	
	c_payment_method	String	The payment method chosen by the user who created the cart or made the order.	
	c_status	String	The status of the order/cart (active, complete or canceled)	
Cart Line	c_number	Integer	The cart number is at the same time a foreign key and a primary key together with the product barcode. It references to the cart it belongs to.	The table Cart Line will store the different products in a cart/order. Each product in a cart will be stored in a different cart line and will reference to the same cart.
	p_barcode	Integer	The product barcode is at the same time a foreign key and a primary key together with the cart number.	
	cl_quantity	Integer	The quantity of a product added to the cart.	
	cl_amount	Double	The amount of money of a single cart line.	

9 IMPLEMENTATION

As mentioned before, the implementation of the web and desktop applications is not part of my project scope. They were implemented previously during the Database Design class. However, I had to do many changes in desktop application in order to improve it and to adapt it to the Android application. I also had to change some things in the web application. These changes took me several days. Afterwards, I had to redesign the database in order to adapt it to the requirements of my project.

Altogether, I spent almost two weeks improving the previous parts of the project. Afterwards, I started implementing the Android application. To be honest, it wasn't as easy as expected, I had to re-discover Android applications' development. I did a lot of research and learned a lot of new things before and during the implementation of the application. First, I took some time to design an attractive GUI. I used Photoshop for that. Then I started by implementing the Login and Register activities working with a SQLite local database. After the authentication, I started working on the display of the home screen and then the products, which are stored in the local database. This SQLite database is synchronized with the MySQL database connected with the desktop and web applications (hosted online), using JSON as follows.



The Login activity:

The following is the code and GUI of the Login activity:

```
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.google.firebase.iid.FirebaseInstanceId;
import com.tecmanic.deshitouch.Config.BaseURL;
import com.tecmanic.deshitouch.ModelClass.ForgotEmailModel;
import com.tecmanic.deshitouch.ModelClass.NotifyModelUser;
import com.tecmanic.deshitouch.R;
import com.tecmanic.deshitouch.network.ApiInterface;
import com.tecmanic.deshitouch.util.LocaleHelper;
import com.tecmanic.deshitouch.util.Session_management;

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

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

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.moshi.MoshiConverterFactory;
```

```

import static com.tecmanic.deshitouch.Config.BaseURL.Login;

public class LoginActivity extends AppCompatActivity {

    Button SignIn;
    TextView forgotPAss, btnignUp;
    EditText etMob, etPass;
    String androidID, token;
    ProgressDialog progressDialog;
    LinearLayout skip;
    private Session_management sessionManagement;

    protected void attachBaseContext(Context newBase) {

        newBase = LocaleHelper.onAttach(newBase);
        super.attachBaseContext(newBase);
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        // token = FirebaseInstanceId.getInstance().getToken();
        FirebaseInstanceId.getInstance().getInstanceId()
            .addOnCompleteListener(task -> {
                if (!task.isSuccessful()) {
                    token = "";
                    Log.i("Login", "getInstanceId failed", task.getException());
                    return;
                }

                // Get new Instance ID token
                token = task.getResult().getToken();
            });
        sessionManagement = new Session_management(LoginActivity.this);
        new Thread(this::checkUserNotify).start();
        init();
    }

    private void init() {

        progressDialog = new ProgressDialog(LoginActivity.this);
        progressDialog.setMessage("Loading...");
        progressDialog.setCancelable(false);

        etMob = findViewById(R.id.etMob);
        etPass = findViewById(R.id.etPass);
        SignIn = findViewById(R.id.btn_Login);
        forgotPAss = findViewById(R.id.btn_ForgotPass);
    }
}

```

```

btnSignUp = findViewById(R.id.btn_Signup);
skip = findViewById(R.id.skip);
checkOtpStatus();
skip.setOnClickListener(v -> {
    sessionManagement.createLoginSession("", "", "", "", "", "", true);
    Intent intent = new Intent(LoginActivity.this, MainActivity.class);
    startActivity(intent);
    finish();
});

forgotPAss.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(LoginActivity.this, ForgotPassOtp.class);
        startActivity(intent);

    }
});

btnSignUp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), SignUpActivity.class);
        startActivity(intent);
        finish();
    }
});

SignIn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (etMob.getText().toString().trim().equalsIgnoreCase("")) {
            Toast.makeText(getApplicationContext(), "Android Number required!", Toast.LENGTH_SHORT).show();
        } else if (etMob.getText().toString().trim().length() < 9) {
            Toast.makeText(getApplicationContext(), "Valid Android Number required!", Toast.LENGTH_SHORT).show();
        } else if (etPass.getText().toString().trim().equalsIgnoreCase("")) {
            Toast.makeText(getApplicationContext(), "Password required!", Toast.LENGTH_SHORT).show();
        } else if (!isOnline()) {
            Toast.makeText(getApplicationContext(), "Please check your Internet Connection!", Toast.LENGTH_SHORT).show();
        } else {
            progressDialog.show();
            loginUrl();

        }
    }
});

```

```

private void loginUrl() {

    if (token != null && !token.equalsIgnoreCase("")) {
        StringRequest stringRequest = new StringRequest(Request.Method.POST, Login, response -> {
            Log.d("Login", response);

            try {
                JSONObject jsonObject = new JSONObject(response);
                String status = jsonObject.getString("status");
                String msg = jsonObject.getString("message");
                if (status.equals("1")) {
                    JSONArray jsonArray = jsonObject.getJSONArray("data");
                    for (int i = 0; i < jsonArray.length(); i++) {

                        JSONObject obj = jsonArray.getJSONObject(i);
                        String user_id = obj.getString("user_id");
                        String user_fullname = obj.getString("user_name");
                        String user_email = obj.getString("user_email");
                        String user_phone = obj.getString("user_phone");
                        String password = obj.getString("user_password");
                        String userimage=obj.getString("user_image");
                        String block = obj.getString("block");

                        progressDialog.dismiss();
                        SharedPreferences.Editor editor = getSharedPreferences(BaseURL.MyPrefreance,
                        MODE_PRIVATE).edit();
                        editor.putString(BaseURL.KEY_ANDROID, user_phone);
                        editor.putString(BaseURL.KEY_PASSWORD, password);
                        editor.apply();
                        sessionManagement.createLoginSession(user_id, user_email, user_fullname,
                        user_phone, password,userimage);
                        sessionManagement.setUserBlockStatus(block);
                        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                        startActivity(intent);
                        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
                        finish();
                    }
                } else {
                    progressDialog.dismiss();
                    Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            progressDialog.dismiss();
        }, error -> progressDialog.dismiss());
    }
}

```

```

@Override
protected Map<String, String> getParams() throws AuthFailureError {
    HashMap<String, String> param = new HashMap<>();
    param.put("user_phone", etMob.getText().toString());
    param.put("user_password", etPass.getText().toString());
    param.put("device_id", token);
    Log.d("ff", token);
    return param;
}

RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.getCache().clear();
requestQueue.add(stringRequest);
} else {
    FirebaseInstanceId.getInstance().getInstanceId()
        .addOnCompleteListener(task -> {
            if (!task.isSuccessful()) {
                token = "";
                Log.i("Login", "getInstanceId failed", task.getException());
                return;
            }
            // Get new Instance ID token
            token = task.getResult().getToken();
            loginUrl();
        });
}
}

private void checkOtpStatus() {
    progressDialog.show();
    Retrofit emailOtp = new Retrofit.Builder()
        .baseUrl(BaseURL.BASE_URL)
        .addConverterFactory(MoshiConverterFactory.create())
        .build();

    ApiInterface apiInterface = emailOtp.create(ApiInterface.class);

    Call<ForgotEmailModel> checkOtpStatus = apiInterface.getOtpOnOffStatus();
    checkOtpStatus.enqueue(new Callback<ForgotEmailModel>() {
        @Override
        public void onResponse(@NonNull Call<ForgotEmailModel> call, @NonNull
retrofit2.Response<ForgotEmailModel> response) {
            if (response.isSuccessful()) {
                ForgotEmailModel model = response.body();
                if (model != null) {

```

```

        if (model.getStatus().equalsIgnoreCase("0")) {
            sessionManagement.setOtpStatus("0");
        } else {
            sessionManagement.setOtpStatus("1");
        }
    }

    progressDialog.dismiss();
}

@Override
public void onFailure(@NonNull Call<ForgotEmailModel> call, @NonNull Throwable t) {
    progressDialog.dismiss();
}
});

}

private boolean isOnline() {
    ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);

    return cm.getActiveNetworkInfo() != null && cm.getActiveNetworkInfo().isConnected();
}

private void checkUserNotify() {
    Retrofit emailOtp = new Retrofit.Builder()
        .baseUrl(BaseURL.BASE_URL)
        .addConverterFactory(MoshiConverterFactory.create())
        .build();

    ApiInterface apiInterface = emailOtp.create(ApiInterface.class);

    Call<NotifyModelUser> checkOtpStatus =
apiInterface.getNotifyUser(sessionManagement.userId());

    checkOtpStatus.enqueue(new Callback<NotifyModelUser>() {
        @Override
        public void onResponse(@NonNull Call<NotifyModelUser> call, @NonNull Response<NotifyModelUser> response) {

            if (response.isSuccessful()){
                if (response.body()!=null){
                    NotifyModelUser modelUser = response.body();
                    if (modelUser.getStatus().equalsIgnoreCase("1")){
                        sessionManagement.setEmailServer(modelUser.getData().getEmail());
                        sessionManagement.setUserSMSService(modelUser.getData().getSms());
                        sessionManagement.setUserInAppService(modelUser.getData().getApp());
                    }
                }
            }
        }
    });
}

```

```
        }
    }

}

@Override
public void onFailure(@NonNull Call<NotifyModelUser> call, @NonNull Throwable t) {
    }

});
```

This activity responsible for the authentication of the user. It is the first activity of the application. The user can choose to Login or Sign up if he is not registered yet.

Home and Navigation Drawer Activity:

After logging in, the Home screen is as follows:

This is the landing activity after the login. And this is the navigation drawer menu.

The code used to implement the navigation drawer and the Home activity is as follows:

```
import android.Manifest;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.graphics.PorterDuff;
import android.graphics.drawable.Drawable;
import android.graphics.drawable.Icon;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.util.Base64;
import android.util.Log;
import android.view.Gravity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.SubMenu;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
```

```
import androidx.core.app.ActivityCompat;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentManager;
import androidx.fragment.app.FragmentTransaction;
import androidx.navigation.ui.AppBarConfiguration;
```



```
import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.google.android.material.navigation.NavigationView;
import com.tecmanic.deshitouch.Config.BaseURL;
import com.tecmanic.deshitouch.Fragments.CartFragment;
import com.tecmanic.deshitouch.Fragments.CategoryFragment;
import com.tecmanic.deshitouch.Fragments.Contact_Us_fragment;
import com.tecmanic.deshitouch.Fragments.Edit_profile_fragment;
import com.tecmanic.deshitouch.Fragments.HomeeeFragment;
import com.tecmanic.deshitouch.Fragments.NotificationFragment;
import com.tecmanic.deshitouch.Fragments.Reward_fragment;
import com.tecmanic.deshitouch.Fragments.SearchFragment;
import com.tecmanic.deshitouch.Fragments.Terms_and_Condition_fragment;
import com.tecmanic.deshitouch.Fragments.Wallet_fragment;
import com.tecmanic.deshitouch.ModelClass.ForgotEmailModel;
import com.tecmanic.deshitouch.ModelClass.NewPendingDataModel;
import com.tecmanic.deshitouch.ModelClass.NotifyModelUser;
import com.tecmanic.deshitouch.R;
import com.tecmanic.deshitouch.network.ApiInterface;
import com.tecmanic.deshitouch.util.DatabaseHandler;
import com.tecmanic.deshitouch.util.FetchAddressTask;
import com.tecmanic.deshitouch.util.GPSTracker;
import com.tecmanic.deshitouch.util.Session_management;
```



```
import org.json.JSONException;
import org.json.JSONObject;
```

```

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

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Retrofit;
import retrofit2.converter.moshi.MoshiConverterFactory;

import static com.tecmanic.deshitouch.Config.BaseURL.ADDRESS;
import static com.tecmanic.deshitouch.Config.BaseURL.CITY;
import static com.tecmanic.deshitouch.Config.BaseURL.COUNTRY;
import static com.tecmanic.deshitouch.Config.BaseURL.KEY_IMAGE;
import static com.tecmanic.deshitouch.Config.BaseURL.KEY_PINCODE;
import static com.tecmanic.deshitouch.Config.BaseURL.LAT;
import static com.tecmanic.deshitouch.Config.BaseURL.LONG;
import static com.tecmanic.deshitouch.Config.BaseURL.MyPrefreance;
import static com.tecmanic.deshitouch.Config.BaseURL.STATE;
import static com.tecmanic.deshitouch.Config.BaseURL.SupportUrl;
import static com.tecmanic.deshitouch.Config.BaseURL.TermsUrl;
import static com.tecmanic.deshitouch.Config.BaseURL.USERBLOCKAPI;
import static com.tecmanic.deshitouch.Config.BaseURL.currencyApi;

public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener,
        GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener,
        LocationListener, FetchAddressTask.OnTaskCompleted,
        SharedPreferences.OnSharedPreferenceChangeListener {
    private static final String TAG = MainActivity.class.getName();
    private static final int REQUEST_LOCATION_PERMISSION = 100;
    public static BottomNavigationView navigation;
    int padding = 0;
    LinearLayout My_Order, My_Reward, My_Wallet, My_Cart, layout_nav_Drawer;
    RelativeLayout loginSignUp, aboutUs, TermsPolicy, ContactUs, share, logout;
    NavigationView navigationView;
    LinearLayout viewpa;
    TextView mTitle, username;
    Button login, signup;
    TextView totalBudgetCount;
    Toolbar toolbar;
    ImageView sliderr;
    ImageView bell;
    Uri path;
    List<Address> addresses = new ArrayList<>();
    String latitude, longitude, address, city, state, country, postalCode;
    LocationManager locationManager;
}

```

```

SharedPreferences sharedpreferences;
LocationCallback mLocationCallback = new LocationCallback() {
    @Override
    public void onLocationResult(LocationResult locationResult) {
        Log.e(TAG, "onLocationResult: called");
    }
};

private DatabaseHandler dbcart;
private Session_management sessionManagement;
private AppBarConfiguration mAppBarConfiguration;
private ImageView profile;
private Menu nav_menu;
private ImageView iv_profile;

private FusedLocationProviderClient mFusedLocationClient;
private SharedPreferences pref;
private BottomNavigationView mbottomNavigationMenuView;
private DrawerLayout drawer;
Bitmap bitmap;
String getimage;
private Object icon;

@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.M) {
        getWindow().setStatusBarColor(Color.BLACK);
    }

    // Bitmap icon = BitmapFactory.decodeResource(getApplicationContext().getResources(),
    // R.drawable.splashicon);

    // Toast.makeText(this, ""+icon, Toast.LENGTH_SHORT).show();

    sessionManagement = new Session_management(MainActivity.this);
    getimage = sessionManagement.getUserDetails().get(KEY_IMAGE);

    bitmap=convertBase64ToBitmap(getimage);

    // //
}

```

```

dbcart = new DatabaseHandler(this);
pref = getSharedPreferences("GOGrocer", Context.MODE_PRIVATE);
pref.registerOnSharedPreferenceChangeListener(this);
navigation = findViewById(R.id.nav_view12);

bell = findViewById(R.id.bell);

profile = findViewById(R.id.profile);

profile.setImageBitmap(bitmap);

int badgeCount = pref.getInt("cardqnty", 0);
if (badgeCount > 0) {
    navigation.getOrCreateBadge(R.id.navigation_notifications123).setNumber(badgeCount);

} else {
    navigation.removeBadge(R.id.navigation_notifications123);
    // totalBudgetCount.setVisibility(View.GONE);
}

profile.setOnClickListener(v -> {
    if (sessionManagement.isLoggedIn()) {
        Edit_profile_fragment fm = new Edit_profile_fragment();
        FragmentManager fragmentManager = getSupportFragmentManager();
        fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
            .addToBackStack(null).commit();
    } else {
        Intent i = new Intent(MainActivity.this, LoginActivity.class);
        startActivity(i);
        overridePendingTransition(0, 0);
    }
});

drawer = findViewById(R.id.drawer_layout);
ImageView menuSlider = findViewById(R.id.sliderr);

menuSlider.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        drawer.openDrawer(Gravity.LEFT);
    }
});
final ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
    this, drawer, toolbar, R.string.navigation_drawer_open,

```

```

R.string.navigation_drawer_close);
    drawer.setDrawerListener(toggle);
    toggle.syncState();
    navigationView = findViewById(R.id.nav_view);
    Menu m = navigationView.getMenu();
    for (int i = 0; i < m.size(); i++) {
        MenuItem mi = m.getItem(i);

        //for applying a font to subMenu ...
        SubMenu subMenu = mi.getSubMenu();
        if (subMenu != null && subMenu.size() > 0) {
            for (int j = 0; j < subMenu.size(); j++) {
                MenuItem subMenuItem = subMenu.getItem(j);
                applyFontToMenuItem(subMenuItem);
            }
        }
    }

    //the method we have create in activity
    //    applyFontToMenuItem(mi);
}

View headerView = navigationView.getHeaderView(0);
navigationView.getBackground().setColorFilter(0x80000000, PorterDuff.Mode.MULTIPLY);
navigationView.setNavigationItemSelectedListener(this);
nav_menu = navigationView.getMenu();
View header = ((NavigationView) findViewById(R.id.nav_view)).getHeaderView(0);
viewpa = header.findViewById(R.id.viewpa);
if (sessionManagement.isLoggedIn()) {
    viewpa.setVisibility(View.VISIBLE);
}

My_Order = header.findViewById(R.id.my_orders);
My_Reward = header.findViewById(R.id.my_reward);
My_Wallet = header.findViewById(R.id.my_wallet);
My_Cart = header.findViewById(R.id.my_cart);

iv_profile = header.findViewById(R.id.iv_header_img);

iv_profile.setImageBitmap(bitmap);

username = header.findViewById(R.id.tv_header_name);
layout_nav_Drawer=header.findViewById(R.id.layout_nav_Drawer);
layout_nav_Drawer.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

```

```

if (sessionManagement.isLoggedIn()) {
    Edit_profile_fragment fm = new Edit_profile_fragment();
    FragmentManager fragmentManager = getSupportFragmentManager();
    fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
        .addToBackStack(null).commit();
} else {
    Intent i = new Intent(MainActivity.this, LoginActivity.class);
    startActivity(i);
    overridePendingTransition(0, 0);
}
});

/* login = (Button) findViewById(R.id.login);
signup = (Button) findViewById(R.id.sign);
username=findViewById(R.id.tv_header_name);*/



My_Order.setOnClickListener(v -> {
    drawer.closeDrawer(GravityCompat.START);
    if (sessionManagement.isLoggedIn()) {
        Intent intent = new Intent(MainActivity.this, My_Order_activity.class);
        startActivityForResult(intent, 4);
    } else {
        Intent intent = new Intent(MainActivity.this, LoginActivity.class);
        startActivityForResult(intent);
    }
});
My_Reward.setOnClickListener(v -> {
    if (sessionManagement.isLoggedIn()) {

        drawer.closeDrawer(GravityCompat.START);

        Reward_fragment fm = new Reward_fragment();
        FragmentManager manager = getSupportFragmentManager();
        FragmentTransaction transaction = manager.beginTransaction();
        transaction.replace(R.id.contentPanel, fm);
        transaction.commit();

        // android.app.FragmentManager fragmentManager = getFragmentManager();
        // fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
        //     .addToBackStack(null).commit();
    } else {
        Intent intent = new Intent(MainActivity.this, LoginActivity.class);
        startActivityForResult(intent);
    }
});
My_Wallet.setOnClickListener(new View.OnClickListener() {
    @Override

```

```

public void onClick(View v) {
    if (sessionManagement.isLoggedIn()) {
        drawer.closeDrawer(GravityCompat.START);
        if (sessionManagement.userBlockStatus().equalsIgnoreCase("2")) {
            Wallet_fragment fm = new Wallet_fragment();
            FragmentManager manager = getSupportFragmentManager();
            FragmentTransaction transaction = manager.beginTransaction();
            transaction.replace(R.id.contentPanel, fm);
            transaction.commit();
        } else {
            showBloackDialog();
        }
    }

    // Wallet_fragment fm = new Wallet_fragment();
    // android.app.FragmentManager fragmentManager = getSupportFragmentManager();
    // fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
    //     .addToBackStack(null).commit();
} else {
    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
    startActivity(intent);
}
}

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

        if (dbcart.getCartCount() > 0) {
            CartFragment favourite_fragment = new CartFragment();
            FragmentManager manager1 = getSupportFragmentManager();
            FragmentTransaction transaction1 = manager1.beginTransaction();
            transaction1.replace(R.id.contentPanel, favourite_fragment);
            transaction1.addToBackStack(null);
            transaction1.commit();
        } else {
            Toast.makeText(MainActivity.this, "No Item in Cart", Toast.LENGTH_SHORT).show();
        }
    }
});

iv_profile.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (sessionManagement.isLoggedIn()) {
            Edit_profile_fragment fm = new Edit_profile_fragment();
            FragmentManager fragmentManager = getSupportFragmentManager();
            fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
                .addToBackStack(null).commit();
        } else {
    }
}

```

```

Intent i = new Intent(MainActivity.this, LoginActivity.class);
startActivity(i);
overridePendingTransition(0, 0);
}
});
});

sideMenu();

new Thread(this::getCurrency).start();
new Thread(this::checkOtpStatus).start();
new Thread(this::checkUserNotify).start();

// checkConnection();

if (savedInstanceState == null) {
    HomeeeFragment fm = new HomeeeFragment();
    FragmentManager fragmentManager = getSupportFragmentManager();
    fragmentManager.beginTransaction()
        .replace(R.id.contentPanel, fm, "Home_fragment")
        .setTransition(FragmentTransaction.TRANSIT_FRAGMENT_OPEN)
        .commit();
}
getSupportFragmentManager().addOnBackStackChangedListener(new
FragmentManager.OnBackStackChangedListener() {
    @Override
    public void onBackStackChanged() {
        try {
            InputMethodManager inputMethodManager = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
            inputMethodManager.hideSoftInputFromWindow(getApplicationContext().getWindowToken(),
0);
            Fragment fr = getSupportFragmentManager().findFragmentById(R.id.contentPanel);

            final String fm_name = fr.getClass().getSimpleName();
            Log.e("backstack: ", ":" + fm_name);
            if (fm_name.contentEquals("Home_fragment")) {
                drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED);
                toggle.setDrawerIndicatorEnabled(true);
                if (getSupportActionBar() != null) {
                    getSupportActionBar().setDisplayHomeAsUpEnabled(false);
                }
                toggle.syncState();
            } else if (fm_name.contentEquals("My_order_fragment") ||
fm_name.contentEquals("Thanks_fragment")) {
                drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
                toggle.setDrawerIndicatorEnabled(false);
                if (getSupportActionBar() != null) {

```

```

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
    toggle.syncState();

    toggle.setToolbarNavigationClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            HomeeeFragment fm = new HomeeeFragment();
            FragmentManager fragmentManager = getSupportFragmentManager();
            fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
                .addToBackStack(null).commit();
        }
    });
} else {

    drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);

    toggle.setDrawerIndicatorEnabled(false);
    if (getSupportActionBar() != null) {
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
    toggle.syncState();

    toggle.setToolbarNavigationClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            onBackPressed();
        }
    });
}

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

GPSTracker mGPS = new GPSTracker(getApplicationContext());
if (mGPS.canGetLocation()) {
    mGPS.getLocation();

    latitude = String.valueOf(mGPS.getLatitude());
    longitude = String.valueOf(mGPS.getLongitude());

    Log.d("lat", latitude);
    Log.d("long", longitude);

    Geocoder geocoder = new Geocoder(getApplicationContext(), Locale.getDefault());
    try {
        addresses = geocoder.getFromLocation(mGPS.getLatitude(), mGPS.getLongitude(), 1); //1
    }
}

```

```

num of possible location returned
    } catch (IOException e) {
        e.printStackTrace();
    }
    if (address != null) {
        address = addresses.get(0).getAddressLine(0); //0 to obtain first possible address
        city = addresses.get(0).getLocality();
        state = addresses.get(0).getAdminArea();
        country = addresses.get(0).getCountryName();
        postalCode = addresses.get(0).getPostalCode();           // commented on 27 sept 19 bcoz of
screen crash
        // create your custom title
        String title = address + "-" + city + "-" + state;
        Log.d("addresss", title + country + "-" + postalCode);

        SharedPreferences.Editor editor = getSharedPreferences(MyPrefreance,
MODE_PRIVATE).edit();
        editor.putString(ADDRESS, address);
        editor.putString(CITY, city);
        editor.putString(STATE, state);
        editor.putString(KEY_PINCODE, postalCode);
        editor.putString(COUNTRY, country);
        editor.putString(LAT, latitude);
        editor.putString(LONG, longitude);
        editor.apply();
        editor.commit();
    }
}

initComponent();
loadFragment(new HomeeeFragment());
// updateHeader();
//sideMenu();

getLocation();

mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

mFusedLocationClient.getLastLocation().addOnSuccessListener(new
OnSuccessListener<Location>() {
    @Override
    public void onSuccess(Location location) {
        Log.e(TAG, "location create"+location.getLatitude() + ", "+location.getLongitude());
        if (location != null) {
            new FetchAddressTask(MainActivity.this, MainActivity.this).execute(location);
        }
    }
});
```

```

}

private void showBloackDialog() {
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.this);
    alertDialog.setCancelable(true);
    alertDialog.setMessage("You are blocked from backend.\n Please Contact with customer care!");
    alertDialog.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.dismiss();
        }
    });
    alertDialog.show();
}

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

private void fetchBlockStatus() {

    if (!sessionManagement.userId().equalsIgnoreCase("")) {
        StringRequest stringRequest = new StringRequest(Request.Method.POST, USERBLOCKAPI, new
Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            Log.d("adresssHoww", response);
            try {
                JSONObject jsonObject = new JSONObject(response);
                String status = jsonObject.getString("status");
                String msg = jsonObject.getString("message");
                if (status.equals("2")) {
                    JSONArray jsonArray = jsonObject.getJSONObject("data");
                    String userStatusValue = jsonArray.getString("block");
                    sessionManagement.setUserBlockStatus("2");
                } else {
                    JSONArray jsonArray = jsonObject.getJSONObject("data");
                    String userStatusValue = jsonArray.getString("block");
                    sessionManagement.setUserBlockStatus("1");
                    Toast.makeText(getApplicationContext(),msg,Toast.LENGTH_SHORT).show();
                }
                Toast.makeText(MainActivity.this,""+msg,Toast.LENGTH_SHORT).show();
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
}

```

```

        }
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
    }
});
@Override
protected Map<String, String> getParams() throws AuthFailureError {
    HashMap<String, String> param = new HashMap<>();
    param.put("user_id", sessionManagement.userId());
    return param;
}
);
RequestQueue requestQueue = Volley.newRequestQueue(MainActivity.this);
requestQueue.getCache().clear();
requestQueue.add(stringRequest);
}

}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    if (requestCode == REQUEST_LOCATION_PERMISSION) {// If the permission is granted, get the location,
        // otherwise, show a Toast
        if (grantResults.length > 0
            && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        //     getLocation();
        Log.e(TAG, "Granted");
        if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
            return;
        }
        mFusedLocationClient.getLastLocation().addOnSuccessListener(new
OnSuccessListener<Location>() {
            @Override
            public void onSuccess(Location location) {
                if (location != null) {
                    Log.e(TAG, "location create" + location.getLatitude() + ", " +
location.getLongitude());
                    new FetchAddressTask(MainActivity.this, MainActivity.this)
                        .execute(location);
                }
            }
        });
    }
}

```

```

        } else {
//            Log.e(TAG, "permission denied");
            Toast.makeText(MainActivity.this, "Location permission is necessary",
Toast.LENGTH_SHORT).show();
            finish();
        }
    }

private void getLocation() {
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]
            {Manifest.permission.ACCESS_FINE_LOCATION},
        REQUEST_LOCATION_PERMISSION);
    } else {
        Log.d(TAG, "getLocation: permissions granted");
    }
}

private void checkUserNotify() {
    Retrofit emailOtp = new Retrofit.Builder()
        .baseUrl(BaseURL.BASE_URL)
        .addConverterFactory(MoshiConverterFactory.create())
        .build();

    ApiInterface apiInterface = emailOtp.create(ApiInterface.class);

    Call<NotifyModelUser> checkOtpStatus =
apiInterface.getNotifyUser(sessionManagement.userId());

    checkOtpStatus.enqueue(new Callback<NotifyModelUser>() {
        @Override
        public void onResponse(@NotNull Call<NotifyModelUser> call, @NotNull
retrofit2.Response<NotifyModelUser> response) {

            if (response.isSuccessful()){
                if (response.body()!=null){
                    NotifyModelUser modelUser = response.body();
                    if (modelUser.getStatus().equalsIgnoreCase("1")){
                        sessionManagement.setEmailServer(modelUser.getData().getEmail());
                        sessionManagement.setUserSMSService(modelUser.getData().getSms());
                        sessionManagement.setUserInAppService(modelUser.getData().getApp());
                    }else {
                        sessionManagement.setEmailServer("0");
                        sessionManagement.setUserSMSService("0");
                        sessionManagement.setUserInAppService("0");
                    }
                }
            }
        }
    });
}

```

```

        }
    }

}

@Override
public void onFailure(@NonNull Call<NotifyModelUser> call, @NonNull Throwable t) {
    }

});

}

@Override
protected void onStop() {
    super.onStop();
    //stop
//    mFusedLocationClient.removeLocationUpdates(mLocationCallback);
}

@Override
public boolean onSupportNavigateUp() {
    return false;
}

private void loadFragment(Fragment fragment) {
    this.getSupportFragmentManager().beginTransaction()
        .replace(R.id.contentPanel, fragment)
        .commitAllowingStateLoss();
}

private void initComponent() {
    navigation.setOnNavigationItemSelected(item -> {
        switch (item.getItemId()) {
            case R.id.navigation_home:
                loadFragment(new HomeeeFragment());

                return true;
            case R.id.navigation_dashboard:
                loadFragment(new CategoryFragment());

            case R.id.navigation_notifications1:
                loadFragment(new SearchFragment());

                return true;
        }
    });
}

```

```

        case R.id.navigation_notifications12:
            loadFragment(new NotificationFragment());
            return true;

        case R.id.navigation_notifications123:
            loadFragment(new CartFragment());
            return true;
    }
    return false;
});
}

@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
    int id = item.getItemId();
    Fragment fm = null;
    Bundle args = new Bundle();
    if (id == R.id.sign) {
        Intent i = new Intent(getApplicationContext(), LoginActivity.class);
        startActivity(i);
    } else if (id == R.id.sign) {
        Intent i = new Intent(getApplicationContext(), LoginActivity.class);
        startActivity(i);
    }

    /* else if (id == R.id.nav_shop_now) {
        fm = new Shop_Now_fragment();
    }*/
    else if (id == R.id.nav_my_profile) {
        fm = new Edit_profile_fragment();

    } else if (id == R.id.nav_aboutus) {
        toolbar.setTitle("About");
        startActivity(new Intent(getApplicationContext(), About_us.class));
    } else if (id == R.id.nav_policy) {
        fm = new Terms_and_Condition_fragment();
        args.putString("url", TermsUrl);
        args.putString("title", getResources().getString(R.string.nav_terms));
        fm.setArguments(args);
    }
    // else if (id == R.id.nav_review) {
    //     //reviewOnApp();
    // }

    else if (id == R.id.nav_contact) {
        fm = new Contact_Us_fragment();
        args.putString("url", SupportUrl);
    }
}

```

```

        args.putString("title", getResources().getString(R.string.nav_terms));
        fm.setArguments(args);

    }

    // else if (id == R.id.nav_review) {
    //     reviewOnApp();
    // }
    else if (id == R.id.nav_share) {
        shareApp();
    } else if (id == R.id.nav_logout) {
        sessionManagement.logoutSession();
        login.setVisibility(View.VISIBLE);
        finish();
    }

    else if (id == R.id.nav_powerd) {
        // stripUnderlines(textView);
        String url = "https://eyewebsolution.com/";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
        finish();
    }

    if (fm != null) {

        FragmentManager fragmentManager = getSupportFragmentManager();
        fragmentManager.beginTransaction().replace(R.id.contentPanel, fm)
            .addToBackStack(null).commit();
    }

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

public void shareApp() {
    Intent sendIntent = new Intent();
    sendIntent.setAction(Intent.ACTION_SEND);

    sendIntent.putExtra(Intent.EXTRA_TEXT, "Hi friends i am using ." + " https://deshitouch.in/" +
        " APP"); //getPackageName()
    sendIntent.setType("text/plain");
    startActivity(sendIntent);

}

public void sideMenu() {

```

```

if (sessionManagement.isLoggedIn()) {
    // tv_number.setVisibility(View.VISIBLE);
    nav_menu.findItem(R.id.nav_logout).setVisible(true);
    nav_menu.findItem(R.id.nav_my_profile).setVisible(true);
    // nav_menu.findItem(R.id.login).setVisible(true);
    nav_menu.findItem(R.id.sign).setVisible(false);
    nav_menu.findItem(R.id.nav_powerd).setVisible(true);

    username.setText("Welcome! " +
        "" + sessionManagement.getUserDetails().get(BaseURL.KEY_NAME));

} else {

    nav_menu.findItem(R.id.login).setVisible(false);
    nav_menu.findItem(R.id.nav_my_profile).setVisible(false);
    nav_menu.findItem(R.id.nav_logout).setVisible(false);
    nav_menu.findItem(R.id.sign).setVisible(true);

    // nav_menu.findItem(R.id.nav_user).setVisible(false);
}
}

private LocationRequest getLocationRequest() {
    LocationRequest locationRequest = new LocationRequest();
    locationRequest.setInterval(20000);
    locationRequest.setFastestInterval(20000);
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
    return locationRequest;
}

@Override
public void onConnected(@Nullable Bundle bundle) {
    Log.e(TAG, "onConnected: ");
}

@Override
public void onConnectionSuspended(int i) {
    Log.e(TAG, "onConnectionSuspended: ");
}

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    Log.e(TAG, "onConnectionFailed: ");
}

@Override

```

```

public void onLocationChanged(Location location) {
    Log.e(TAG, "onLocationChanged: ");
}

@Override
public void onTaskCompleted(String result) {
    Log.e(TAG, "onTaskCompleted: " + result);

    ((TextView) findViewById(R.id.address)).setText(result);
}

@Override
public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key) {
    if (key.equalsIgnoreCase("cardqnty")) {
        // totalBudgetCount.setText(pref.getInt("cardqnty",0));
        int badgeCount = pref.getInt("cardqnty", 0);
        if (badgeCount > 0) {
            // totalBudgetCount.setVisibility(View.VISIBLE);
            // totalBudgetCount.setText(""+badgeCount);
            navigation.getOrCreateBadge(R.id.navigation_notifications123).setNumber(badgeCount);
        } else {
            // totalBudgetCount.setVisibility(View.GONE);
            navigation.removeBadge(R.id.navigation_notifications123);
        }
    }
}

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

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

    if (requestCode == 4) {
        if (data != null && data.getExtras() != null) {

            ArrayList<NewPendingDataModel> orderSubModels = (ArrayList<NewPendingDataModel>)
data.getSerializableExtra("datalist");
            if (orderSubModels != null) {
                dbcart.clearCart();
                for (int i = 0; i < orderSubModels.size(); i++) {
                    NewPendingDataModel odModel = orderSubModels.get(i);
                    if (odModel.getDescription() != null && !odModel.getDescription().equalsIgnoreCase(""))
                        double price =

```

```

Double.parseDouble(odModel.getPrice())/Double.parseDouble(odModel.getQty());
    HashMap<String, String> map = new HashMap<>();
    map.put("varient_id", odModel.getVarient_id());
    map.put("product_name", odModel.getProduct_name());
    map.put("category_id", odModel.getVarient_id());
    map.put("title", odModel.getProduct_name());
    map.put("price", String.valueOf(price));
    map.put("mrp", odModel.getTotal_mrp());
    map.put("product_image", odModel.getVarient_image());
    map.put("status", "1");
    map.put("in_stock", "");
    map.put("unit_value", odModel.getQuantity()+" "+odModel.getUnit());
    map.put("unit", "");
    map.put("increment", "0");
    map.put("rewards", "0");
    map.put("stock", "0");
    map.put("product_description", odModel.getDescription());

    if (!odModel.getQty().equalsIgnoreCase("0")) {
        dbcart.setCart(map, Integer.parseInt(odModel.getQty()));
    } else {
        dbcart.removeItemFromCart(map.get("varient_id"));
    }

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        pref.edit().putInt("cardqty", dbcart.getCartCount()).apply();
    }
}
}

loadFragment(new CartFragment());
}

}

super.onActivityResult(requestCode, resultCode, data);
}

private void getCurrency(){
    StringRequest stringRequest = new StringRequest(Request.Method.GET, currencyApi, response ->
{
    Log.d("currency api", response);

    try {

        JSONObject currencyObject = new JSONObject(response);
        if (currencyObject.getString("status").equalsIgnoreCase("1") &&

```

```

currencyObject.getString("message").equalsIgnoreCase("currency")){
    JSONObject dataObject = currencyObject.getJSONObject("data");
    sessionManagement.setCurrency(dataObject.getString("currency_name"),dataObject.getString("currency_sign"));
}
} catch (JSONException e) {
    e.printStackTrace();
}
}, error -> {
}) {
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        return super.getParams();
    }
};

RequestQueue requestQueue = Volley.newRequestQueue(this);
requestQueue.getCache().clear();
requestQueue.add(stringRequest);
}

private void checkOtpStatus() {
    Retrofit emailOtp = new Retrofit.Builder()
        .baseUrl(BaseURL.BASE_URL)
        .addConverterFactory(MoshiConverterFactory.create())
        .build();

    ApiInterface apiInterface = emailOtp.create(ApiInterface.class);

    Call<ForgotEmailModel> checkOtpStatus = apiInterface.getOtpOnOffStatus();
    checkOtpStatus.enqueue(new Callback<ForgotEmailModel>() {
        @Override
        public void onResponse(@NonNull Call<ForgotEmailModel> call, @NonNull retrofit2.Response<ForgotEmailModel> response) {
            if (response.isSuccessful()) {
                ForgotEmailModel model = response.body();
                if (model != null) {
                    if (model.getStatus().equalsIgnoreCase("0")) {
                        sessionManagement.setOtpStatus("0");
                    } else {
                        sessionManagement.setOtpStatus("1");
                    }
                }
            }
        }
    });
}

```

```

        }

    }

    @Override
    public void onFailure(@NonNull Call<ForgotEmailModel> call, @NonNull Throwable t) {
        }

    });

}

private Bitmap convertBase64ToBitmap(String b64) {
    byte[] imageAsBytes = Base64.decode(b64.getBytes(), Base64.DEFAULT);
    return BitmapFactory.decodeByteArray(imageAsBytes, 0, imageAsBytes.length);
}

}

```

Category Code:-

Then, the user will be able to view the categories of products and browse the products in each category:

This is some of the code responsible for this activity:

After choosing a category, the application displays the list of products under that category in a gridView. This is done using a function call in the DatabaseHelper.

```

package com.tecmanic.deshitouch.Fragments;

import android.app.AlertDialog;
import android.content.Context;
import android.net.ConnectivityManager;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.fragment.app.Fragment;

```

```

import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.Volley;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.tecmanic.deshitouch.Adapters.HomeCategoryAdapter;
import com.tecmanic.deshitouch.ModelClass.HomeCate;
import com.tecmanic.deshitouch.R;
import com.tecmanic.deshitouch.util.CustomVolleyJsonRequest;

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

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

import static com.tecmanic.deshitouch.Config.BaseURL.Categories;

public class CategoryFragment extends Fragment {

    RecyclerView recyclerView;
    RecyclerView recyclerSubCate;
    HomeCategoryAdapter cateAdapter, subcateAdapter;
    ProgressDialog progressDialog;
    String catId;
    Gson gson;
    private List<HomeCate> cateList = new ArrayList<>();
    private List<HomeCate> subcateList = new ArrayList<>();
    private boolean isSubcat = false;

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

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

        View view = inflater.inflate(R.layout.fragment_categories, container, false);
        recyclerView = view.findViewById(R.id.recyclerCATE);
        recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
    }
}

```

```

getActivity().setTitle(getResources().getString(R.string.Category));
progressDialog = new ProgressDialog(getContext());

progressDialog.setMessage("Loading...");
progressDialog.setCancelable(false);

GsonBuilder gsonBuilder = new GsonBuilder();
gsonBuilder.setDateFormat("M/d/yy hh:mm a");
gson = gsonBuilder.create();
if (isOnline()) {
    categoryUrl();
}

return view;
}

private void categoryUrl() {
cateList.clear();
// Tag used to cancel the request
String tag_json_obj = "json_get_address_req";

Map<String, String> params = new HashMap<String, String>();
params.put("parent", "");

CustomVolleyJsonRequest jsonObjReq = new CustomVolleyJsonRequest(Request.Method.GET,
        Categories, params, new Response.Listener<JSONObject>() {

    @Override
    public void onResponse(JSONObject response) {
        Log.d("categdrytyguoj", response.toString());

        try {
            if (response != null && response.length() > 0) {
                String status = response.getString("status");
                if (status.equals("1")) {
                    JSONArray array = response.getJSONArray("data");

                    for (int i = 0; i < array.length(); i++) {

                        JSONObject object = array.getJSONObject(i);
                        HomeCate model = new HomeCate();

                        model.setDetail(object.getString("description"));
                        model.setId(object.getString("cat_id"));
                        model.setImages(object.getString("image"));
                        model.setName(object.getString("title"));
                    }
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

model.setSub_array(object.getJSONArray("subcategory"));
cateList.add(model);

cateAdapter = new HomeCategoryAdapter(cateList, getContext());
recyclerView.setAdapter(cateAdapter);
cateAdapter.notifyDataSetChanged();
}

}

} else {
// Toast.makeText(getActivity(),msg,Toast.LENGTH_SHORT).show();
}
progressDialog.dismiss();
} catch (JSONException e) {
e.printStackTrace();
}
progressDialog.dismiss();
}

}, new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
}
}) {
@Override
protected Map<String, String> getParams() throws AuthFailureError {
HashMap<String, String> param = new HashMap<>();
return param;
};
RequestQueue requestQueue = Volley.newRequestQueue(getContext());
requestQueue.getCache().clear();
requestQueue.add(jsonObjReq);
}

}

private boolean isOnline() {
ConnectivityManager cm = (ConnectivityManager)
getActivity().getSystemService(Context.CONNECTIVITY_SERVICE);

return cm.getActiveNetworkInfo() != null && cm.getActiveNetworkInfo().isConnected();
}
}

```

Checkout:

Afterwards, the user can perform the checkout and send his order. Here is some of the code responsible for that.

```
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.RadioGroup;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.RecyclerView;

import com.android.volley.DefaultRetryPolicy;
import com.android.volley.NoConnectionError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.RetryPolicy;
import com.android.volley.TimeoutError;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.razorpay.Checkout;
import com.razorpay.PaymentResultListener;
import com.tecmanic.deshitouch.Config.BaseURL;
import com.tecmanic.deshitouch.Config.SharedPreferences;
import com.tecmanic.deshitouch.ModelClass.CoupenModel;
import com.tecmanic.deshitouch.R;
import com.tecmanic.deshitouch.util.AppController;
import com.tecmanic.deshitouch.util.CustomVolleyJsonRequest;
import com.tecmanic.deshitouch.util.DatabaseHandler;
import com.tecmanic.deshitouch.util.Session_management;
```

```

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

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

import static com.android.volley.VolleyLog.TAG;

public class PaymentDetails extends AppCompatActivity implements
PaymentResultListener {

    LinearLayout llwallet, llpromocode, llcards, llcod;

    String Prefrence_TotalAmmount;
    String lat, lng;
    String getwallet;
    LinearLayout Promo_code_layout, Coupon_and_wallet;
    RelativeLayout Apply_Coupon_Code, Relative_used_wallet, Relative_used_coupon;
    List<CoupunModel> coupunModelList = new ArrayList<>();
    RecyclerView coupen_recycler;
    SharedPreferences sharedPreferences12;
    SharedPreferences.Editor editor12;
    String code, cart_id;
    TextView twallet, tcod, tcards, tpromocode;
    LinearLayout backcart;
    int status = 0;
    String payment_method;
    Button confirm;
    String wallet_amount = "00";
    TextView reset_text;
    String wallet_status = "no";
    String total_amt;
    TextView payable_ammount, my_wallet_ammount, used_wallet_ammount,
used_coupon_ammount, order_ammount;
    CheckBox rb_Store, rb_Cod, rb_card;
    CheckBox checkBox_Wallet;
    CheckBox checkBox_coupon;
    EditText et_Coupon;
    String getvalue;
    String text;
    String cp;
    String Used_Wallet_amount;
    String total_amount;
    String order_total_amount;
}

```

```

TextView linea, lineb;
RadioGroup radioGroup;
private TextView coupontxt;
private DatabaseHandler db_cart;
private Session_management sessionManagement;
private String getlocation_id = "";
private String getstore_id = "";
private String gettime = "";
private String getdate = "";
private String getuser_id = "";
private Double rewards;
private String remaingprice = "";
private boolean coupounApplied = false;
private String payable_amt = "";
private double walletbalance = 0;
private ProgressDialog progressDialog;
private String coupon_amount = "";

private TextView coupon_apply_t;

@Override
public void onBackPressed() {
    finish();
    super.onBackPressed();
}

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

    sessionManagement = new Session_management(this);
    getwallet = SharedPref.getString(getApplicationContext(),
    baseURL.KEY_WALLET_Ammount);
    db_cart = new DatabaseHandler(this);
    getuser_id = sessionManagement.userId();
    progressDialog = new ProgressDialog(PaymentDetails.this);
    progressDialog.setMessage("Payment is in processing...");
    progressDialog.setCancelable(false);
    llwallet = findViewById(R.id.llwallet);
    coupon_apply_t = findViewById(R.id.coupon_apply_t);
    llpromocode = findViewById(R.id.llpromocode);
    llcod = findViewById(R.id.llcod);
    llcards = findViewById(R.id.llcards);
    backcart = findViewById(R.id.backcart);
    confirm = findViewById(R.id.confirm_order);
    // cart_id = OrderSummary.cart_id;
    cart_id = sessionManagement.getCartId();
}

```

```

linea = findViewById(R.id.line1);
lineb = findViewById(R.id.line2);
reset_text = findViewById(R.id.reset_text);
twallet = findViewById(R.id.wallettext);
tcod = findViewById(R.id.txtcod);
tcards = findViewById(R.id.txtcards);
tpromocode = findViewById(R.id.txtpromo);
checkBox_Wallet = findViewById(R.id.use_wallet);
rb_Store = findViewById(R.id.use_store_pickup);
rb_Cod = findViewById(R.id.use_COD);
rb_card = findViewById(R.id.use_card);
checkBox_coupon = findViewById(R.id.use_coupon);
et_Coupon = findViewById(R.id.et_coupon_code);
Promo_code_layout = findViewById(R.id.prommocode_layout);
Apply_Coupon_Code = findViewById(R.id.apply_coupoun_code);
radioGroup = findViewById(R.id.radio_group);
my_wallet_amount = findViewById(R.id.user_wallet);
order_amount = findViewById(R.id.order_amount);
used_wallet_amount = findViewById(R.id.used_wallet_amount);
used_coupon_amount = findViewById(R.id.used_coupon_ammount);
coupuntxt = findViewById(R.id.coupuntxt);
coupuntxt.setText("Apply");
checkBox_Wallet.setClickable(false);
rb_card.setClickable(false);
rb_Cod.setClickable(false);
checkBox_coupon.setClickable(false);

reset_text.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(getApplicationContext(), PaymentDetails.class);
        intent.putExtra("order_amt", total_amt);
        startActivity(intent);
        checkBox_Wallet.setClickable(false);
        rb_card.setClickable(false);
        rb_Cod.setClickable(false);
        checkBox_coupon.setClickable(false);
        total_amount = getIntent().getStringExtra("order_amt");
        payable_amt = getIntent().getStringExtra("order_amt");
        code = "";
        payment_method = "";
        wallet_status = "no";
        getwallet = SharedPref.getString(getApplicationContext(),
        BaseURL.KEY_WALLET_Ammount);
        wallet_amount = getwallet;
        walletbalnce = 0;
        my_wallet_amount.setText(sessionManagement.getCurrency() + "" +
        wallet_amount);
    }
})

```

```

        order_amount.setText(total_amount + " " + sessionManagement.getCurrency());
        my_wallet_amount.setTextColor(getResources().getColor(R.color.black));
        checkBox_Wallet.setChecked(false);
        llwallet.setBackgroundResource(R.drawable.border_rounded1);
        twallet.setTextColor(getResources().getColor(R.color.black));
        checkBox_coupon.setChecked(false);
        llpromocode.setBackgroundResource(R.drawable.border_rounded1);
        tpromocode.setTextColor(getResources().getColor(R.color.black));
        rb_Cod.setChecked(false);
        llcod.setBackgroundResource(R.drawable.border_rounded1);
        tcod.setTextColor(getResources().getColor(R.color.black));
        rb_card.setChecked(false);
        llcards.setBackgroundResource(R.drawable.border_rounded1);
        tcards.setTextColor(getResources().getColor(R.color.black));
        et_Coupon.setText("");
        Promo_code_layout.setVisibility(View.GONE);
        coupon_apply_t.setVisibility(View.VISIBLE);
        Promo_code_layout.setClickable(true);
    }
});

total_amount = getIntent().getStringExtra("order_amt");
payable_amt = getIntent().getStringExtra("order_amt");

order_amount.setText(total_amount + " " + sessionManagement.getCurrency());

backcart.setOnClickListener(v -> {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        finishAndRemoveTask();
    } else {
        finish();
    }
});

// Prefrence_TotalAmmount = SharedPref.getString(getApplicationContext(),
BaseURL.TOTAL_AMOUNT);

sharedPreferences12 = getSharedPreferences("loction", MODE_PRIVATE);
editor12 = sharedPreferences12.edit();
lat = sharedPreferences12.getString("lat", "77.1111");
lng = sharedPreferences12.getString("lng", "22.02002");

// code = getIntent().getStringExtra("code");
// Log.d("dff", code);
// checkBox_coupon.setTypeface(font);

Apply_Coupon_Code.setOnClickListener(v -> {

```

```

        if (!coupuntxt.getText().toString().trim().equalsIgnoreCase("Applied")) {
            // startActivity(new Intent(PaymentDetails.this, Coupen.class));
            code = et_Coupon.getText().toString().trim();
            apply();
        }
    }
});
```

getRefresrh();
rewardliness();

```

final String email = sessionManagement.getUserDetails().get(BaseURL.KEY_EMAIL);
final String Android =
sessionManagement.getUserDetails().get(BaseURL.KEY_ANDROID);
final String name = sessionManagement.getUserDetails().get(BaseURL.KEY_NAME);
confirm.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        progressDialog.show();
        try {
            if (rb_Cod.isChecked() && checkBox_Wallet.isChecked()) {
                wallet_status = "yes";
                payment_method = "COD";
                makeAddOrderRequest(getuser_id, cart_id, payment_method, wallet_status,
"success");
            } else if (rb_card.isChecked() && checkBox_Wallet.isChecked()) {
                wallet_status = "yes";
                payment_method = "card";
                startPayment(name, total_amount, email, Android);
            } else if (rb_card.isChecked()) {
                wallet_status = "no";
                payment_method = "card";
                startPayment(name, total_amount, email, Android);
            } else if (rb_Cod.isChecked()) {
                wallet_status = "no";
                payment_method = "COD";
                makeAddOrderRequest(getuser_id, cart_id, payment_method, wallet_status,
"success");
            } else if (checkBox_Wallet.isChecked()) {
                wallet_status = "yes";
                payment_method = "wallet";
                if (walletbalance == 0.0 && Double.parseDouble(total_amount) > 0.0) {
                    progressDialog.dismiss();
                    Toast.makeText(PaymentDetails.this, "Select Card Or COD",
Toast.LENGTH_SHORT).show();
                } else {
                    makeAddOrderRequest(getuser_id, cart_id, payment_method, wallet_status,
"success");
                }
            }
        }
    }
});
```

```

        }

    } else {
        if (checkBox_coupon.isChecked()) {
            if (Double.parseDouble(total_amount) > 0.0) {
                progressDialog.dismiss();
                Toast.makeText(PaymentDetails.this, "Select One plz",
Toast.LENGTH_SHORT).show();
            } else {
                wallet_status = "no";
                payment_method = "promocode";
                makeAddOrderRequest(getuser_id, cart_id, payment_method,
wallet_status, "success");
            }
        }
    }

} catch (Exception ex)
{
    Toast.makeText(PaymentDetails.this, "confirm"+ex.getMessage(),
Toast.LENGTH_SHORT).show();
}
}

});  

//  

llcards.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (total_amount.equalsIgnoreCase("0")||total_amount.equalsIgnoreCase("0.0")||total_amou
nt.equalsIgnoreCase("")){

    }else {
        if (!rb_card.isChecked()) {
            rb_card.setChecked(true);
            rb_Cod.setChecked(false);
            llcards.setBackgroundResource(R.drawable.gradientbg);
            tcards.setTextColor(getResources().getColor(R.color.white));
            llcod.setBackgroundResource(R.drawable.border_rounded1);
            tcod.setTextColor(getResources().getColor(R.color.black));
            if (checkBox_Wallet.isChecked()) {
                wallet_status = "yes";
            } else {
                wallet_status = "no";
            }
            payment_method = "cards";
        }
    }
}

```

```

rb_card.setChecked(false);
llcards.setBackgroundResource(R.drawable.border_rounded1);
tcards.setTextColor(getResources().getColor(R.color.black));
if (checkBox_Wallet.isChecked()) {
    wallet_status = "yes";
    payment_method = "wallet";
} else {
    payment_method = "";
    wallet_status = "no";
}
}

checkBox_Wallet.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if
(total_amount.equalsIgnoreCase("0")||total_amount.equalsIgnoreCase("0.0")||total_amou
nt.equalsIgnoreCase("")){

    }else {
        if (!rb_Cod.isChecked()) {
            rb_Cod.setChecked(true);
            rb_card.setChecked(false);
            llcod.setBackgroundResource(R.drawable.gradientbg);
            tcod.setTextColor(getResources().getColor(R.color.white));
            llcards.setBackgroundResource(R.drawable.border_rounded1);
            tcards.setTextColor(getResources().getColor(R.color.black));
            if (checkBox_Wallet.isChecked()) {
                wallet_status = "yes";
            } else {
                wallet_status = "no";
            }
            payment_method = "COD";
        } else {
            rb_Cod.setChecked(false);
            rb_card.setChecked(false);
            llcod.setBackgroundResource(R.drawable.border_rounded1);
            tcod.setTextColor(getResources().getColor(R.color.black));
            if (checkBox_Wallet.isChecked()) {
                wallet_status = "yes";
            }
        }
    }
}
});
```

```

        payment_method = "wallet";
    } else {
        wallet_status = "no";
        payment_method = "";
    }
}

checkBox_Wallet.setClickable(false);
rb_card.setClickable(false);
rb_Cod.setClickable(false);
checkBox_coupon.setClickable(false);
}
});

// checkBox_Wallet.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
//     @Override
//     public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
//
//     }
// });
// });

llwallet.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    if (!checkBox_Wallet.isChecked()) {
        checkBox_Wallet.setChecked(true);
        llwallet.setBackgroundResource(R.drawable.gradientbg);
        twallet.setTextColor(getResources().getColor(R.color.white));
        my_wallet_amount.setTextColor(getResources().getColor(R.color.white));
        double amt = Double.parseDouble(total_amount);
        double wallet = Double.parseDouble(wallet_amount);
        if (wallet > 0) {
            if (amt <= wallet) {
                walletbalance = wallet - amt;
                total_amount = "0";
                llcards.setClickable(false);
                llcod.setClickable(false);
                llpromocode.setClickable(false);
                my_wallet_amount.setText(sessionManagement.getCurrency() + "" +
walletbalance);
                order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
            } else {
                walletbalance = 0;
                total_amount = String.valueOf((amt - wallet));
                my_wallet_amount.setText(sessionManagement.getCurrency() + "" +
walletbalance);
            }
        }
    }
}
});

```

```

        order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
//      startActivity(new Intent(getApplicationContext(), RechargeWallet.class));
    }
    wallet_status = "yes";
    payment_method = "wallet";
} else {
    wallet_status = "no";
    payment_method = "";
    checkBox_Wallet.setChecked(false);
    llwallet.setBackgroundResource(R.drawable.border_rounded1);
    twallet.setTextColor(getResources().getColor(R.color.black));
    my_wallet_amount.setTextColor(getResources().getColor(R.color.black));
    double wallett = Double.parseDouble(wallet_amount);
    my_wallet_amount.setText(sessionManagement.getCurrency() + wallett);
    total_amount = payable_amt;
    walletbalance = wallet;
    order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
    wallet_status = "no";
    payment_method = "";
    llcod.setClickable(true);
    llcards.setClickable(true);
    llpromocode.setClickable(true);
    startActivityForResult(new Intent(PaymentDetails.this, RechargeWallet.class),
5);
}

}
else {
    checkBox_Wallet.setChecked(false);
    llwallet.setBackgroundResource(R.drawable.border_rounded1);
    twallet.setTextColor(getResources().getColor(R.color.black));
    my_wallet_amount.setTextColor(getResources().getColor(R.color.black));
    double wallet = Double.parseDouble(wallet_amount);
    my_wallet_amount.setText(sessionManagement.getCurrency() + "" + wallet);

    if (checkBox_coupon.isChecked()){
        if (coupon_amount!=null && !coupon_amount.equalsIgnoreCase("")){
            double amountd = Double.parseDouble(payable_amt)-
Double.parseDouble(coupon_amount);
            total_amount = String.valueOf(amountd);
            walletbalance = wallet;
            order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
            wallet_status = "no";
            payment_method = "promocode";
        }else {
            total_amount = payable_amt;
        }
    }
}

```

```

        walletbalance = wallet;
        order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
        wallet_status = "no";
        payment_method = "";
    }

}else {
    total_amount = payable_amt;
    walletbalance = wallet;
    order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
    wallet_status = "no";
    payment_method = "";
}
llcod.setClickable(true);
llcards.setClickable(true);
llpromocode.setClickable(true);
}

checkBox_Wallet.setClickable(false);
rb_card.setClickable(false);
rb_Cod.setClickable(false);
checkBox_coupon.setClickable(false);
})
);

llpromocode.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (!checkBox_coupon.isChecked()) {
    coupon_apply_t.setVisibility(View.GONE);
    checkBox_coupon.setChecked(true);
    llpromocode.setBackgroundResource(R.drawable.gradientbg);
    tpromocode.setTextColor(getResources().getColor(R.color.white));
    Intent coupounIntent = new Intent(PaymentDetails.this, Coupen.class);
    startActivityForResult(coupounIntent, 2);
} else {
    coupon_apply_t.setVisibility(View.VISIBLE);
    checkBox_coupon.setChecked(false);
    llpromocode.setBackgroundResource(R.drawable.border_rounded1);
    tpromocode.setTextColor(getResources().getColor(R.color.black));
    et_Coupon.setText("");
    Promo_code_layout.setVisibility(View.GONE);
    Promo_code_layout.setClickable(true);
    if (code!=null && !code.equalsIgnoreCase("")){
        apply();
    }
}
}

```

```
        checkBox_Wallet.setClickable(false);
        rb_card.setClickable(false);
        rb_Cod.setClickable(false);
        checkBox_coupon.setClickable(false);
    }
});

//    checkBox_coupon.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
}

private void rewardliness() {

    String tag_json_obj = "json_order_detail_req";

    Map<String, String> params = new HashMap<String, String>();
    params.put("cart_id", cart_id);

    CustomVolleyJsonRequest jsonObjReq = new
CustomVolleyJsonRequest(Request.Method.POST,
    BaseURL.rewardlines, params, new Response.Listener<JSONObject>() {

        @Override
        public void onResponse(JSONObject response) {
            Log.d("TAG", response.toString());

            try {

                String status = response.getString("status");
                String message = response.getString("message");

                if (status.contains("1")) {

                    String line1 = response.getString("line1");
                    String line2 = response.getString("line2");
                    linea.setVisibility(View.VISIBLE);
                    lineb.setVisibility(View.VISIBLE);
                    if (line1.equalsIgnoreCase("")) {
                        linea.setVisibility(View.GONE);
                        lineb.setText(line2);
                    }
                    if (line2.equalsIgnoreCase("")) {
                        lineb.setVisibility(View.GONE);
                        linea.setText(line1);
                    }
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}
```

```

        }

        linea.setText(line1);
        lineb.setText(line2);

    } else {
        linea.setVisibility(View.GONE);
        lineb.setVisibility(View.GONE);
    }
}

} catch (JSONException e) {
    e.printStackTrace();
}
}, new Response.ErrorListener() {

    @Override
    public void onErrorResponse(VolleyError error) {
        VolleyLog.d("TAG", "Error: " + error.getMessage());
        if (error instanceof TimeoutError || error instanceof NoConnectionError) {
        }
    }
});

// Adding request to request queue
AppController.getInstance().addToRequestQueue(jsonObjReq, tag_json_obj);
jsonObjReq.setRetryPolicy(new DefaultRetryPolicy(1000,
    DefaultRetryPolicy.DEFAULT_MAX_RETRIES,
    DefaultRetryPolicy.DEFAULT_BACKOFF_MULT));
AppController.getInstance().getRequestQueue().getCache().clear();
AppController.getInstance().getRequestQueue().add(jsonObjReq);

}

private void apply() {
    progressDialog.show();
    String tag_json_obj = "json_order_detail_req";
    Map<String, String> params = new HashMap<String, String>();
    params.put("cart_id", cart_id);
    params.put("coupon_code", code);
    // Log.d("ssd", cart_id);
    // Log.d("dd", code);
}

CustomVolleyJsonRequest jsonObjReq = new
CustomVolleyJsonRequest(Request.Method.POST,

```

```

BaseURL.apply_coupon, params, new Response.Listener<JSONObject>() {

    @Override
    public void onResponse(JSONObject response) {
        Log.d("TAG", response.toString());
        try {

            String statuss = response.getString("status");
            String message = response.getString("message");
            if (statuss.contains("1")) {
                coupounApplied = true;
                JSONObject jsonObject = response.getJSONObject("data");
                remaingprice = jsonObject.getString("rem_price");
                String deliverycharge = jsonObject.getString("delivery_charge");
                coupuntxt.setText("Applied");
                coupon_amount = jsonObject.getString("coupon_discount");
                if (checkBox_Wallet.isChecked()) {
                    double remInt = Double.parseDouble(remaingprice);
                    double wallet = Double.parseDouble(wallet_amount);

                    if (wallet >= remInt) {
                        walletbalance = (int) (wallet - remInt);
                        total_amount = "0";
                        rb_card.setClickable(false);
                        rb_Cod.setClickable(false);
                        checkBox_coupon.setClickable(true);
                        my_wallet_ammount.setText(sessionManagement.getCurrency() +
                                walletbalance);
                        order_ammount.setText(total_amount + " " +
                                sessionManagement.getCurrency());
                    } else {
                        if (wallet == 0.0) {
                            total_amount = remaingprice;
                            order_ammount.setText(total_amount + " " +
                                sessionManagement.getCurrency());
                        } else {
                            walletbalance = 0;
                            total_amount = String.valueOf((remInt - wallet));
                            order_ammount.setText(total_amount + " " +
                                sessionManagement.getCurrency());
                        }
                    }
                } else {
                    total_amount = remaingprice;
                    order_ammount.setText(total_amount + " " +
                            sessionManagement.getCurrency());
                }
            }
        }
    }
}

```

```

        Toast.makeText(getApplicationContext(), "" + message,
        Toast.LENGTH_SHORT).show();
    }
    else if (statuss.contains("2")) {
        coupounApplied = false;
        code = "";
        status = 1;
        if (checkBox_Wallet.isChecked()) {
            double amt = Double.parseDouble(payable_amt);
            double wallet = Double.parseDouble(wallet_amount);
            if (wallet > 0) {
                if (amt <= wallet) {
                    walletbalance = wallet - amt;
                    total_amount = "0";
                    rb_card.setClickable(false);
                    rb_Cod.setClickable(false);
                    checkBox_coupon.setClickable(false);
                    my_wallet_amount.setText(sessionManagement.getCurrency() + " " +
walletbalance);
                    order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
                } else {
                    walletbalance = 0;
                    total_amount = String.valueOf((amt - wallet));
                    order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
                }
                startActivity(new Intent(getApplicationContext(), RechargeWallet.class));
            }
        }
    } else {
        total_amount = payable_amt;
        order_amount.setText(total_amount + " " +
sessionManagement.getCurrency());
    }
}
else {
    coupounApplied = false;
    code = "";
    status = 1;
    Promo_code_layout.setVisibility(View.GONE);
    coupon_apply_t.setVisibility(View.VISIBLE);
    checkBox_coupon.setChecked(false);
    llpromocode.setBackground(R.drawable.border_rounded1);
    tpromocode.setTextColor(getResources().getColor(R.color.black));
    Toast.makeText(getApplicationContext(), "" + message,
    Toast.LENGTH_SHORT).show();
}

```

```

        } catch (JSONException e) {
            e.printStackTrace();
        } finally {
            progressDialog.dismiss();
        }
    }
}, new Response.ErrorListener() {

    @Override
    public void onErrorResponse(VolleyError error) {
        VolleyLog.d("TAG", "Error: " + error.getMessage());
        progressDialog.dismiss();
        if (error instanceof TimeoutError || error instanceof NoConnectionError) {
        }
    }
});

// Adding request to request queue
AppController.getInstance().addToRequestQueue(jsonObjReq, tag_json_obj);
}

private void makeAddOrderRequest(String userid, String cart_id,
String payment_method, String wallet_status, String payment_status) {
    String tag_json_obj = "json_add_order_req";
    Map<String, String> params = new HashMap<String, String>();
    params.put("user_id", userid);
    params.put("payment_status", payment_status);
    params.put("cart_id", cart_id);
    params.put("payment_method", payment_method);
    params.put("wallet", wallet_status);
    //    params.put("lat",lat);
    //    params.put("lng",lng);
    //    Toast.makeText(this, "order:--"+params, Toast.LENGTH_SHORT).show();

    CustomVolleyJsonRequest jsonObjReq = new
    CustomVolleyJsonRequest(Request.Method.POST,
        BaseURL.ADD_ORDER_URL, params, new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                Log.d(TAG, response.toString());

                try {
                    String status = response.getString("status");
                    String message = response.getString("message");
                    if (status.equalsIgnoreCase("1")) {
                        sessionManagement.setCartID("");
                        JSONObject jsonObject = response.getJSONObject("data");

```

```

db_cart.clearCart();
Intent intent = new Intent(getApplicationContext(), OrderSuccessful.class);
intent.putExtra("msg", message);
startActivity(intent);
Toast.makeText(PaymentDetails.this, "" +
message, Toast.LENGTH_SHORT).show();
} else if (status.equalsIgnoreCase("2")) {
sessionManagement.setCartID("");
JSONObject jsonObject = response.getJSONObject("data");
db_cart.clearCart();
Intent intent = new Intent(getApplicationContext(), OrderSuccessful.class);
intent.putExtra("msg", message);
startActivity(intent);
Toast.makeText(PaymentDetails.this, "" + message, Toast.LENGTH_SHORT).show()
} else {
Toast.makeText(PaymentDetails.this, "" + message, Toast.LENGTH_SHORT).show()
}
progressDialog.dismiss();
} catch (JSONException e) {
e.printStackTrace();
}
},
new Response.ErrorListener() {

@Override
public void onErrorResponse(VolleyError error) {
progressDialog.dismiss();
VolleyLog.d(TAG, "Error: " + error.getMessage());
}
});

jsonObjReq.setRetryPolicy(new RetryPolicy() {
@Override
public int getCurrentTimeout() {
return 50000;
}

@Override
public int getCurrentRetryCount() {
return 5;
}

@Override
public void retry(VolleyError error) throws VolleyError {
Log.i("Volley", "" + error.getMessage());
}
});

AppController.getInstance().addToRequestQueue(jsonObjReq, tag_json_obj);

```

```

}

public void getRefresrh() {
    String user_id = sessionManagement.userId();
    RequestQueue rq = Volley.newRequestQueue(getApplicationContext())
    StringRequest strReq = new
StringRequest(Request.Method.POST, BaseURL.WALLET_REFRESH + user_id,
        new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jObj = new JSONObject(response);

                String statusss = jObj.getString("status");
                if (statusss.equalsIgnoreCase("1")) {
                    String wallet_amountss = jObj.getString("data");
                    wallet_amount = wallet_amountss;
                    my_wallet_ammount.setText(sessionManagement.getCurrency() + " " +
wallet_amount);
                    SharedPref.putString(PaymentDetails.this, BaseURL.KEY_WALLET_Ammount,
wallet_amount);
                } else {
                    // Toast.makeText(DashboardPage.this, " " + jObj.optString("msg"),
                    Toast.LENGTH_LONG).show();
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }, new Response.ErrorListener() {

        @Override
        public void onErrorResponse(VolleyError error) {
    }
}) {
};

rq.add(strReq);
}

public void startPayment(String name, String amount, String email, String phone) {
/*
You need to pass current activity in order to let Razorpay create CheckoutActivity

```

```

        */

    final Activity activity = this;
    final Checkout co = new Checkout();
    try {
        JSONObject options = new JSONObject();
        options.put("name", name);
        options.put("description", "Shopping Charges");
        //You can omit the image option to fetch the image from dashboard
        options.put("image", "https://s3.amazonaws.com/rzp-Android/images/rzp.png");
        options.put("currency", "INR");

        options.put("amount", Double.parseDouble(amount) * 100);

        JSONObject preFill = new JSONObject();
        preFill.put("email", email);
        preFill.put("contact", phone);
        options.put("prefill", preFill);

        co.open(activity, options);

    } catch (Exception e) {
        Toast.makeText(PaymentDetails.this, "Error in payment: " + e.getMessage(),
        Toast.LENGTH_SHORT).show();
        e.printStackTrace();
    }
}

@Override
public void onPaymentSuccess(String s) {
    makeAddOrderRequest(getuser_id, cart_id, payment_method, wallet_status, "success");
}

@Override
public void onPaymentError(int i, String s) {
    progressDialog.dismiss();
    Toast.makeText(this, s, Toast.LENGTH_SHORT).show();
}

```

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 2) {
        if (data != null && data.getExtras() != null) {
            code = data.getExtras().getString("code");
            Log.i(TAG, code);
            if (code == null) {
                code = "";
                status = 1;
                Promo_code_layout.setVisibility(View.GONE);
                coupon_apply_t.setVisibility(View.VISIBLE);
                checkBox_coupon.setChecked(false);
                llpromocode.setBackground(R.drawable.border_rounded1);
                tpromocode.setTextColor(getResources().getColor(R.color.black));
            } else if (code.equalsIgnoreCase("")) {
                code = "";
                status = 1;
                Promo_code_layout.setVisibility(View.GONE);
                coupon_apply_t.setVisibility(View.VISIBLE);
                checkBox_coupon.setChecked(false);
                llpromocode.setBackground(R.drawable.border_rounded1);
                tpromocode.setTextColor(getResources().getColor(R.color.black));
            } else {
                checkBox_coupon.setChecked(true);
                llpromocode.setBackground(R.drawable.gradientbg);
                tpromocode.setTextColor(getResources().getColor(R.color.white));
                tcod.setTextColor(getResources().getColor(R.color.black));
                tcards.setTextColor(getResources().getColor(R.color.black));
                twallet.setTextColor(getResources().getColor(R.color.black));
                my_wallet_ammount.setTextColor(getResources().getColor(R.color.black));
                et_Coupon.setText(code);
                Promo_code_layout.setVisibility(View.VISIBLE);
                Promo_code_layout.setClickable(false);
                apply();
            }
        }
    } else if (requestCode == 5) {
        if (data != null && data.getExtras() != null) {
            if (Objects.requireNonNull(data.getStringExtra("recharge")).equalsIgnoreCase("success")) {
                getRefresrh();
            }
        }
    }
}

```

10 TECHNOLOGY ENABLERS

Desktop Application:

For the desktop application that was developed before, the following technology enablers were used:

JAVA		JAVA was used to develop the whole the desktop application
MySQL		MySQL was used for the backend, it is linked to both the desktop and web application.

Web Application:

For the web application that was developed before as well, the following technology enablers were used:

HTML		HTML was used as a markup language for the web application
CSS		CSS was used for the design of the Web Application
PHP		PHP combined with HTML was used for the scripting.

Android Application:

For the Android application that is to be developed during the implementation phase of the project, these technology enablers are going to be used:

Android SDK		Android SDK is used to develop the Android application
SQLite		As the android SDK comes with a package that contains SQLite specific classes, SQLite is used for the local database of the Android application.

11 TESTING

Testing was done throughout the development of the project. In fact, I had an android phone connected with the computer. Whenever I add a feature or a functionality, I test it right away. This helped me find errors right away and have each and every functionality of the application tested. I used to host the MySQL database and test the synchronization of MySQL and SQLite database using JSON thanks to the free hosting service provided by the website.

This is a small example for testing the connection of the Android app with the MySQL database in order to synchronize the new registered account.

12 Conclusion & Future Perspectives:

Obviously, there is still a lot to work on in the Android application if I want to launch it to the market. There is also the possibility of developing the iOS version of the application. This Android e-shopping store is promising and will certainly boost Deshi Touch Grocery' business. In fact, even if the application doesn't bring the company many orders (through the application), it could serve as a marketing strategy. This will be done after adding the push notifications to it and sending notifications to users whenever there is a new product or a new discount. That is among the reasons why I chose to have the user register in the application, in order to have his data. This will allow us to send him promotions to his phone as well as his email address.

I find it amazing how I learned Android application development using Android throughout the semester while developing the application at the same time. I made use of Youtube.com tutorials and stackoverflow.com's solution to each and every problem I face while developing the app.

More than that, working on this capstone project was a great opportunity for me to put the knowledge, I've acquired during these four years, into practice. However, it showed me some of my weak points and helped me transform them into strong points.

Now, I know how to gather requirements, design, implement, and test a software. In addition, I learned how to implement the server side, the client side and the data side, each one separately from the other one.

I also got to learn how to analyze the social, technological, economical, ecological, political, legal, and ethical aspects of a project and get the best out of it.