

VEGETABLE VENDOR ANDROID APPLICATION

By

LINET M SHAJI (P191314)

MAIN PROJECT REPORT

MASTER OF SCIENCE

IN

COMPUTER SCIENCE



CENTRAL UNIVERSITY OF TAMIL NADU

THIRUVARUR – 610 005

MAY 2021

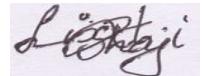
DECLARATION

I hereby declare that the thesis entitled “**VEGETABLE VENDOR APPLICATION**” submitted by me to the Department of Computer Science, Central University of Tamil Nadu in partial fulfillment of the requirements for the award of the degree of Master of Computer Science of Central University of Tamil Nadu is a bonafide work carried out by me under the supervision of guide_name, Department of Computer Science, Central University of Tamil Nadu.

I further declare that the work reported in this thesis has not been submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.

Date:

Place: Thiruvarur



LINET M SHAJI

BONAFIDE CERTIFICATE

This is to certify that the project report entitled “**VEGETABLE VENDOR APPLICATION**” submitted by **LINET M SHAJI (Reg. No:P191314)** of II year / IV semester during the academic year 20XX-XX in partial fulfillment of the requirement for the award of the degree of Master of Computer Science of Central University of Tamil Nadu is a record of bonafide work done carried out by him / her under my guidance.

This work has not been submitted elsewhere for the award of any other degree to the best of our knowledge.

GUIDE

HEAD OF DEPARTMENT

Submitted for the University Viva-Voce Examination held on _____

INTERNAL EXAMINER(S)

ACKNOWLEDGEMENT

Sometimes our light goes out, but is blown into flame by another human being. We owe our deepest thanks to those who have this light”.

I hereby dedicated this project to the god almighty who's abundant and grace and mercy enabled its successful completion. I would like to extend my gratitude to all the people who have inspired and motivated me to make this project success.

I express my gratitude to **Dr.A.MARTIN**, Department of Computer Science my project guide for their source encouragement and expert guidance throughout the project period.

I am thankful to all faculty members of the department of Computer Science, Central University of Tamilnadu, for the inspiration and encouragement shown to me throughout the semester.

I would like to extend my sincere thanks to all my colleagues, without whose support and encouragement this project would not have attained its perfection in its total. I convey my thanks to all who helped me directly and indirect to carry out the project successfully.

LINET M SHAJI

ABSTRACT

The proposed system Vegetable Vendor Application is an android application developed using the Android Studio Java and Firebase. The main objective of this project to order and sell vegetables and fruits online. It contains users such as Seller and Customer. In this application multiple sellers from a place can register as seller and can sell vegetables and fruits. So the customer can buy fruits and vegetables from other sellers also if the required fruits and vegetables are not there in stock with a seller.

The Sellers user can sell vegetables by registering to this application. Once the Seller has registered with this application, the seller can login, the seller can add products, update the product details such as product name, image, price and stock status (such as In stock or Out of Stock).

The Customers user can order can order vegetables online by registering to this application. Once the Customer has registered with these application, the customer can login, view products which are In stock, filter by categories(such as fruits, leafy, root and other vegetables), can search by vegetables or fruits name, can add vegetables and fruits to cart and the customer can order it together if the seller name of all the vegetables and fruits are same, if it is different we can order one at a time and logout.

CONTENTS

DECLARATION

BONAFIDE

ACKNOWLEDGEMENT

ABSTRACT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. INTRODUCTION

1.1 ABOUT THE PROJECT

2. PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

2.1 INTRODUCTION

2.2 PROBLEM DEFINITION

2.2.1 EXISTING SYSTEM

2.2.2 DRAWBACKS OF EXISTING SYSTEM

2.2.3 PROPOSED SYSTEM

2.2.4 FEATURES OF PROPOSED SYSTEM

2.3 FEASIBILITY ANALYSIS

2.3.1 OPERATING FEASIBILITY

2.3.2 TECHNICAL FEASIBILITY

2.3.3 ECONOMICAL FEASIBILITY

3. SOFTWARE REQUIREMENT AND SPECIFICATION

3.1 INTRODUCTION

3.2 SCOPE

3.3 GENERAL DESCRIPTIONS

3.3.1 PRODUCT FUNCTION

3.3.2 USER CHARACTERISTICS

3.3.3 GENERAL CONSTRAINTS

3.3.4 FUNCTIONAL REQUIREMENTS

3.3.4.1 TECHNICAL ISSUES

3.3.5 INTERFACE REQUIREMENTS

3.3.5.1 HARDWARE REQUIREMENTS

3.3.5.2 SOFTWARE REQUIREMENTS

3.3.6 OTHER FUNCTIONAL REQUIREMENTS

3.3.6.1 SECURITY

3.3.6.2 RELIABILITY

3.3.6.3 MAINTAINABILITY

3.3.6.4 USABILITY

4. SYSTEM ANALYSIS

4.1 BEHAVIORAL DIAGRAMS

4.1.1 USE CASE DIAGRAM

4.1.3 ACTIVITY DIAGRAM

5. SYSTEM DESIGN

5.1 ARCHITECTURAL DESIGN

6. IMPLEMENTATION

7. TESTING

7.1 TESTING PLAN

7.2 UNIT TESTING

7.2 INTEGRATION TESTING

7.3 VALIDATION TESTING

8. CONCLUSIONS

BIBLIOGRAPHY

APPENDIX A-SCREEN SHOTS

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

The proposed system Vegetable Vendor Application is an android application developed using the Android Studio Java and Firebase. Which is a very useful application in this pandemic situation. During this pandemic situation it is very important to maintain social distance and avoid unwanted interactions with peoples. The proposed system helps customer to buy vegetables and fruits online, by using ordering vegetables and fruits online the customer can stay home, avoid unwanted interaction with other by going out.

The main objective of this project to order and sell vegetables and fruits online. It contains users such as Sellers and Customers.

The Sellers user can sell vegetables by registering to this application. Once the Seller has registered with this application, the seller can login, the seller can add products, update the product details such as product name, image, price and stock status (such as In stock or Out of Stock).

The Customers user can order can order vegetables online by registering to this application. Once the Customer has registered with these application, the customer can login, view products which are In stock, filter by categories(such as fruits, leafy, root and other vegetables), can search by vegetables or fruits name, can add vegetables and fruits to cart and the customer can order it together if the seller name of all the vegetables and fruits are same, if it is different we can order one at a time and logout.

CHAPTER 2

PROBLEM DEFINITION AND FEASIBILITY ANALYSIS

2.1 INTRODUCTION

The vegetable vendor application is a very useful application which helps people to save time and avoid unwanted interaction by staying home itself to order vegetables and fruits online.

2.2 PROBLEM DEFINITION

2.2.1 EXISTING SYSTEM

In the existing we need to go outside to buy vegetables and fruits everyday.

2.2.2 DRAWBACK OF EXISTING SYSTEM

In the existing every time we need to go outside to buy fruits and vegetables from market, so it is a time consuming process and also travelling everyday to buy vegetables and fruits will make lot unwanted interactions with peoples when we are travelling and go to market. In this pandemic situation maintaining social distancing with others and avoid unwanted interactions are very important to not avoid chances of getting corona. So it is not a approach in this pandemic to go out every time to get vegetables and fruits daily.

2.2.3 PROPOSED SYSTEM

The proposed system is a vegetable vendor android application developed using Android Studio Java and Firebase database. It contains two users such as sellers and customers.

2.2.4 FEATURES OF THE PROPOSED SYSTEM

The main feature of proposed system is we can order fruits and vegetables online by using our phone and we will get delivery at home. It contains users such as Sellers and Customers. Also in this application multiple sellers from a place can register as seller and can sell vegetables and fruits. So the customer can buy fruits and vegetables from other sellers also if the required fruits and vegetables are not there in stock with a seller.

Functionalities of Sellers user

- **Register:** Seller user can register with this application using name, phone number and by entering password.
- **Login :** Seller can login to the system by using registered phone number and password.
- **Update profile details :** Seller can update profile by adding profile picture, address, new phone number and name.
- **Add Vegetables and Fruits :** Seller can add vegetables and fruits to application based on category such as root, fruits, leafy and other vegetables by login to the system by entering name, price and image. Once the product is added to the system by default the stock stack will be “In stock”.
- **Update Price and Stock status :** Seller can update daily price and stock status by entering product price and stock status as “In stock” or “Out of stock”.

- **View orders** : Seller can view orders and update delivery status as delivered.
- **View cart products** : Seller can also view the products which are in users cart
- **Logout**: Seller can logout from the system once the all activity has finished

Functionalities of Customer user

- **Register:** Seller user can register with this application using name, phone number and by entering password.
- **Login :** Seller can login to the system by using registered phone number and password.
- **Update profile details** : Seller can update profile by adding profile picture, address, new phone number and name.
- **View Vegetables and Fruits :** **Customer** can view vegetables and fruits where in stock, it will display name, price, category, stock status and contact details of seller such as seller name and phone number.
- **Filter by category :** Customer can view vegetables and fruits category based such as fruits, leafy, root and other vegetables.
- **Search vegetables and fruits :** Customer can search vegetables and fruits based entering vegetables or fruits name.
- **Add to cart :** Customer can add vegetables and fruits to cart.
- **Place order:** Customer can order more than one item at a time if vegetables and fruits if are from same seller or else can place one order at a time.
- **Logout:** Customer can logout from the system once the all activity has finished

2.3 FEASIBILITY ANALYSIS

A feasibility study is conducted to select the best system that meets performance requirement. This entails an identification description, and evaluation of candidate system and the selection of best system for the job. The system required performance is defined by a statement of constraints, the identification of specific system objective and a description of outputs.

The key consideration in feasibility study is:

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility

2.3.1 OPERATIONAL FEASIBILITY

The system will be used if it is developed well then be resistance for users that undetermined. No major training and new skills are required as it is based on DBMS model. It will help in the time saving and fast processing and dispersal of user request and applications. New product will provide all the benefits of present system with better performance.Improved information, better management and collection of reports.

User support.

User involvement in the building of present system is sought to keep in mind the user specific requirement and need.User will have control over their information.Faster and systematic processing of user request approval.

2.2.2 TECHNICAL FEASIBILITY

It is a measure of the practicality of a specific technical solution and the availability of technical resources and expertise. The proposed system uses XML as Front-end, Firebase and Android Studio Java as Back-end tool. Firebase is a popular tool used to cloud hosting and design and develop database objects such as table view, indexes. The above tools are readily available, easy to work with and widely used for developing commercial application.

Hardware used in developing this project are Intel CORE i3 CPU @ 2.00GHz, 4.00 GB RAM, 64-bit operating system, x64-based processor, 1 TB Hard disk. This hardware's were already available on the existing computer system. The software Windows 10 is installed on the existing computer system. So no additional hardware and software were required to purchase and it is feasible. As the users increase we have to buy a new powerful server. IN initial stage is not required

2.3.3 ECONOMIC FEASIBILITY

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and estimate time for the recovery of cost incurred. It is also determines whether it is worthwhile to invest the money in the proposed project. Economic feasibility determines by the means of cost benefit analysis. The proposed system is economically feasible because we don't have to give the salary to the admin panel. As per the nature of this application we don't need to invest much more money in this system. Because it is not specified for the outside marketing world. It is only for the former customers. So investing money is not needed. And one more thing, we are not applying any kind of fees like procedure to the users. So it is completely free for the usable. That is it is economically feasible for users, they don't have to spend money for being a user. This is also applicable in the case of the developing side. The less time involved also helped in its economic feasibility. Because of initially we are planning to release online hotel booking in a particular hotel. So no need for a high performing server. The backend required for storing other details is also the same database that is Firebase. The computers in the organization are highly sophisticated and don't needs extra components to load the software. Hence the organization can implement the new system without any additional expenditure. Hence, it is economically feasible.

Chapter 3

SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1 INTRODUCTION

- **Operation System** : Microsoft Windows 7/8/10 (32-bit or 64- bit) □
- **IDE Software** : Android studio□
- **Front End** : XML □
- **Language** : JAVA □
- **Back End** : Firebase Realtime Database
- **Testing**: Android OS(mobile OS)

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

XML is used to design visual structure for a user interface, such as the UI for an Android application. **Java** is used as a controller for controlled UI (Layout file). It gets the data from the Layout file and after processing that data output will be shown in the UI layout. It works on the backend of an Android application.

Firebase Realtime Database is a cloud-hosted database. Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

To test the application developed we can use either android emulator or our android phone. Here I have used to android phone for testing. There is no need to reinstall the application every time made changes to applications. We just need to run the application from android studio to made the changes the application that we have already installed in our phone..

3.2 SCOPE

In the future, we can develop a more advanced form of this Android Application. We can add features such as native language implementation, and we can also integrate it with other languages also. Now the application is developed in English, we can integrate it with Tamil, Telugu, Malayalam etc. Now the application provides only cash on delivery, in future we can implement payment gateway to provide net banking, Google pay or other UPI payments.

3.3. GENERAL DESCRIPTION

3.3.1. PRODUCT FUNCTION

Functionalities of Sellers user

- **Register:** Seller user can register with this application using name, phone number and by entering password.

- **Login :** Seller can login to the system by using registered phone number and password.
- **Update profile details :** Seller can update profile by adding profile picture, address, new phone number and name.
- **Add Vegetables and Fruits :** Seller can add vegetables and fruits to application based on category such as root, fruits, leafy and other vegetables by login to the system by entering name, price and image. Once the product is added to the system by default the stock stack will be “In stock”.
- **Update Price and Stock status :** Seller can update daily price and stock status by entering product price and stock status as “In stock” or “Out of stock”.
- **View orders :** Seller can view orders and update delivery status as delivered.
- **View cart products :** Seller can also view the products which are in users cart
- **Logout:** Seller can logout from the system once the all activity has finished

Functionalities of Customer user

- **Register:** Seller user can register with this application using name, phone number and by entering password.
- **Login :** Seller can login to the system by using registered phone number and password.
- **Update profile details :** Seller can update profile by adding profile picture, address, new phone number and name.
- **View Vegetables and Fruits :** **Customer** can view vegetables and fruits where in stock, it will display name, price, category, stock status and contact details of seller such as seller name and phone number.
- **Filter by category :** Customer can view vegetables and fruits category based such as fruits, leafy, root and other vegetables.
- **Search vegetables and fruits :** Customer can search vegetables and fruits based entering vegetables or fruits name.
- **Add to cart :** Customer can add vegetables and fruits to cart.
- **Place order:** Customer can order more than one item at a time if vegetables and fruits if are from same seller or else can place one order at a time.
- **Logout:** Customer can logout from the system once the all activity has finished

3.3.2. USER CHARACTERISTICS:

Users of the website are members, administrators who maintain this website. Users are assumed to have basic knowledge of computers and Internet browsing. Friendly user interface, online help and user guide must be sufficient to educate the users on how to use this product without any problems or difficulties.

3.3.3. GENERAL CONSTRAINTS:

The information of the all the users details of vegetables and fruits must be stored in the database. Users must register with application to use it. And login should based on the registration details such the password and phone number given during registration.

3.3.4. FUNCTIONAL REQUIREMENTS:

Various functional modules that can be implemented by the system will be,

1. Register, Login and Logout
2. Manage Vegetables and Fruits
3. Manage Order
4. Manage Cart

3.2.4.1 TECHNICAL ISSUES

As android studios supports in the device with 2GB RAM it is not sufficient to work with because the emulator uses more working space (i.e) RAM space so android studios will work slow the developer will not be able to get the expected output. The android studios will need the device which consist of minimum of 4GB RAM to work with and it will be more comfortable with 8GB RAM and more than 8GB RAM.

3.2.5 INTERFACE REQUIREMENTS

For developing an android app android studios is required. The output is acquired by the use of ANDROID EMULATOR. If at all we can get the output by connecting to an android mobile. The mobile can be connected using an USB cable. To use the EMULATOR we can select our own android version. We require ANDROID SDK(Software Development Kit).Where we can run and get the output. If we are using android mobile for retrieving the output there are some steps to be followed, they are

1. Connect your device to your development machine with a USB cable. If you're developing on Windows, you might need to install the appropriate USB driver for your device.
2. Enable USB debugging in the Developer options as follows.
3. First, we must enable the developer options:
4. Open the Settings app.
5. (Only on Android 8.0 or higher) Select System.
6. Scroll to the bottom and select About phone.
7. Scroll to the bottom and tap Build number 7 times.
8. Return to the previous screen to find Developer options near the bottom.
9. Open Developer options, and then scroll down to find and enable USB debugging.

3.2.5.1 HARDWARE REQUIREMENTS

Android development tools run on all flavours of computers of relatively recent vintage: Linux, Mac OS, or Microsoft Windows. An Android device such as a Smartphone or tablet is useful (and of course the ultimate target for development), but is in fact not essential to getting started since the software contains virtual device emulators that allow you to develop and test.

3.2.5.2 SOFTWARE REQUIREMENTS □

- **Operation System :** Microsoft Windows 7/8/10 (32-bit or 64- bit) □
- **IDE Software :** Android studio□
- **Front End :** XML □
- **Language :** JAVA □
- **Back End :** Firebase Realtime Database
- **Testing:** Android OS(mobile OS)

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

XML is used to design visual structure for a user interface, such as the UI for an Android application. **Java** is used as a controller for controlled UI (Layout file). It gets the data from the Layout file and after processing that data output will be shown in the UI layout. It works on the backend of an Android application.

Firebase Realtime Database is a cloud-hosted database. Realtime Database uses data synchronization—every time data changes, any connected device receives that update within milliseconds. Firebase apps remain responsive even when offline because the Firebase Realtime Database SDK persists your data to disk. Once connectivity is reestablished, the client device receives any changes it missed, synchronizing it with the current server state.

To test the application developed we can use either android emulator or our android phone. Here I have used to android phone for testing. There is no need to reinstall the application every time made changes to applications, We just need to run the application from android studio to made the changes the application that we have already installed in our phone..

3.2.6 OTHER FUNCTIONAL ATTRIBUTES

3.2.6.1 SECURITY

This app is a completely secured app where, this app uses local database. So the data within this app will not be known by others.

3.2.6.2 RELIABILITY

The mobile app provides a reliable environment to both customers and owner. All are running and executing without any errors.

3.2.6.3 MAINTAINABILITY

There is no special care is needed to maintain this app. In this app we need not want to install the application for each and every updates instead we just update it whenever we want we can update and just install it and this updating will take less time and quickly runs.

3.2.6.4 USABILITY

This app is a user friendly app. This app takes less memory space so there no fear of memory consumption. This has a look and feel appearance of fresh to meat application. Which has a colorful appearance so the user will get interest to use this app.

CHAPTER 4

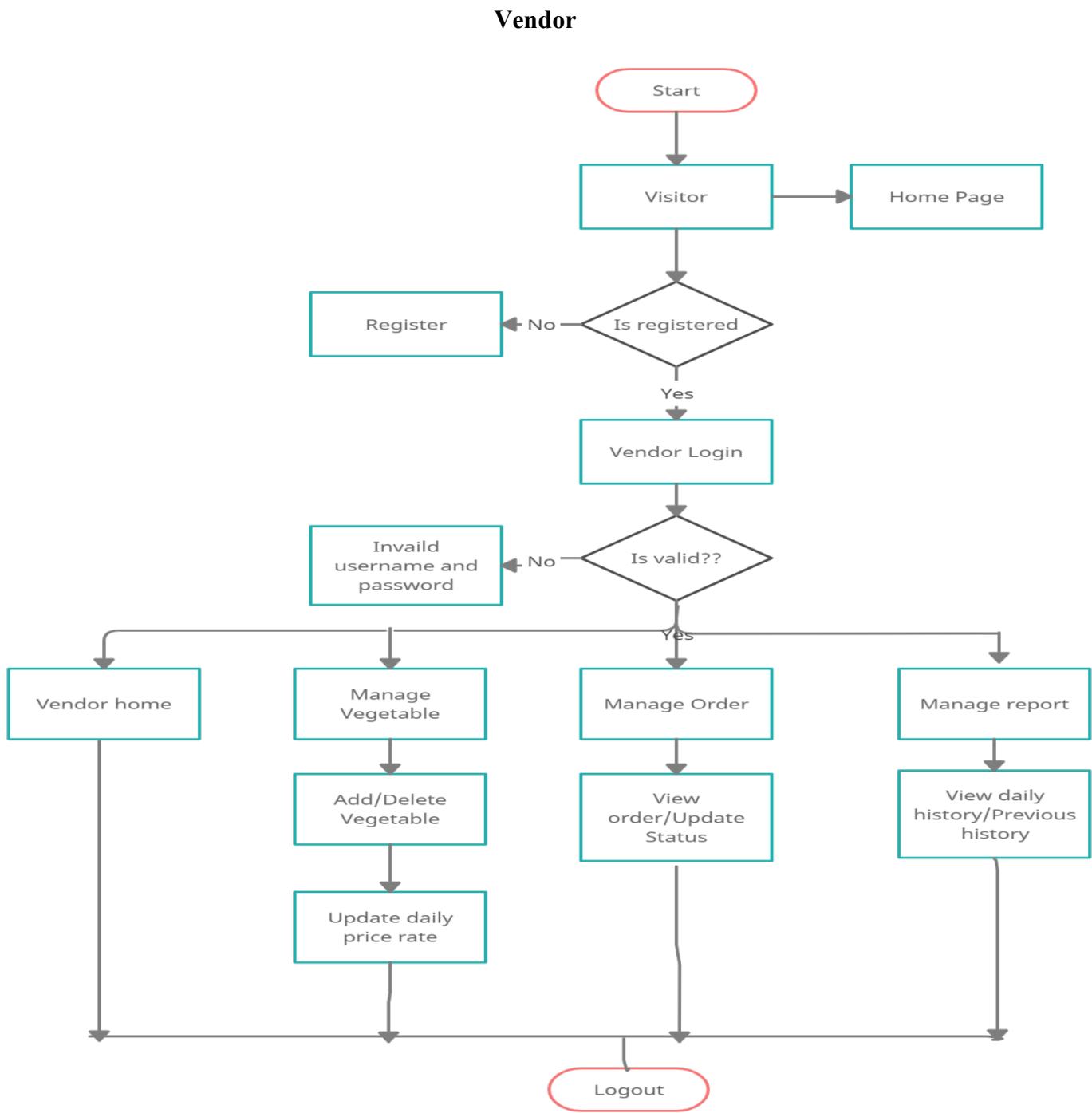
SYSTEM ANALYSIS

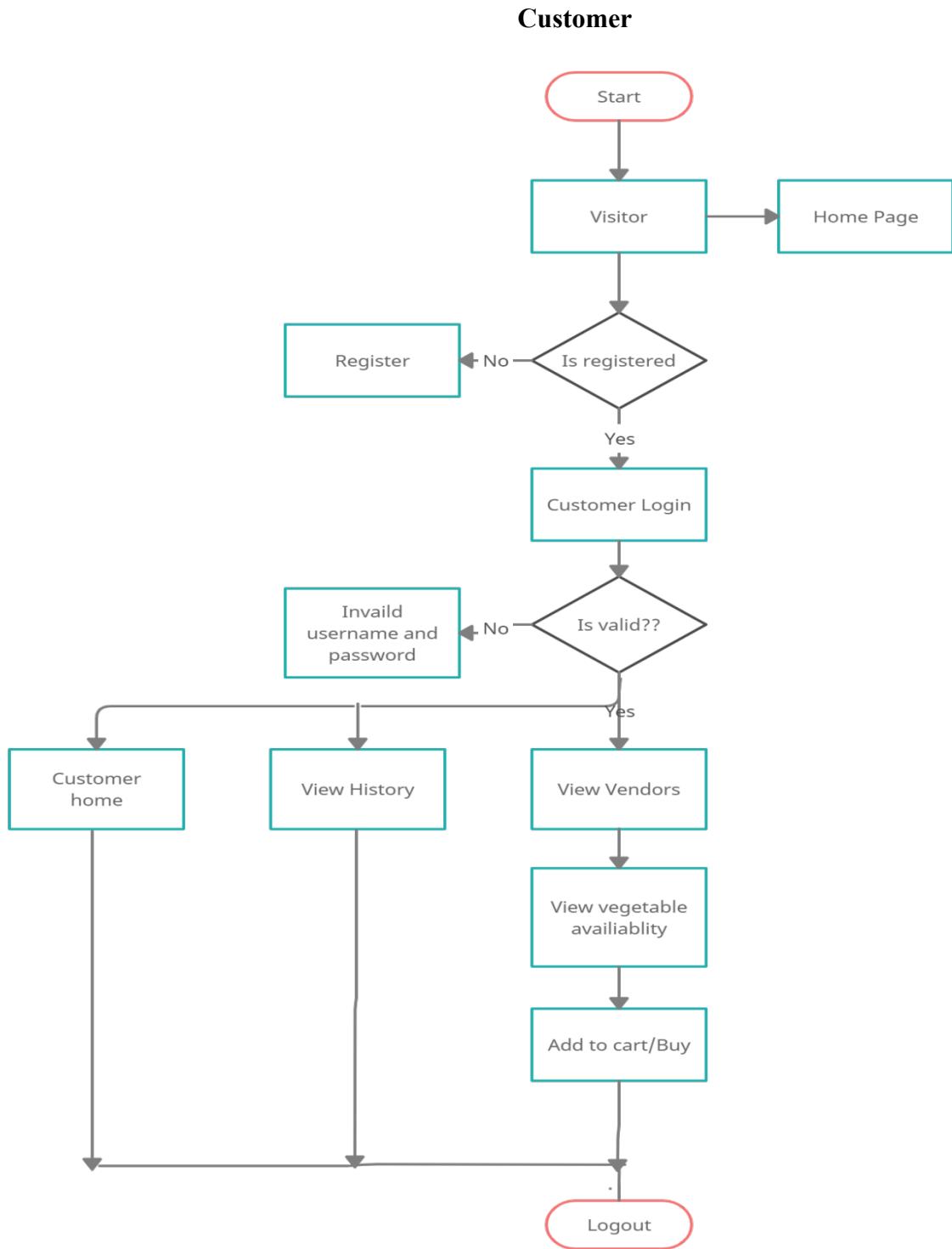
4.1 BEHAVIORAL DIAGRAMS

4.1.1 USE CASE DIAGRAM



4.1.3 ACTIVITY DIAGRAM





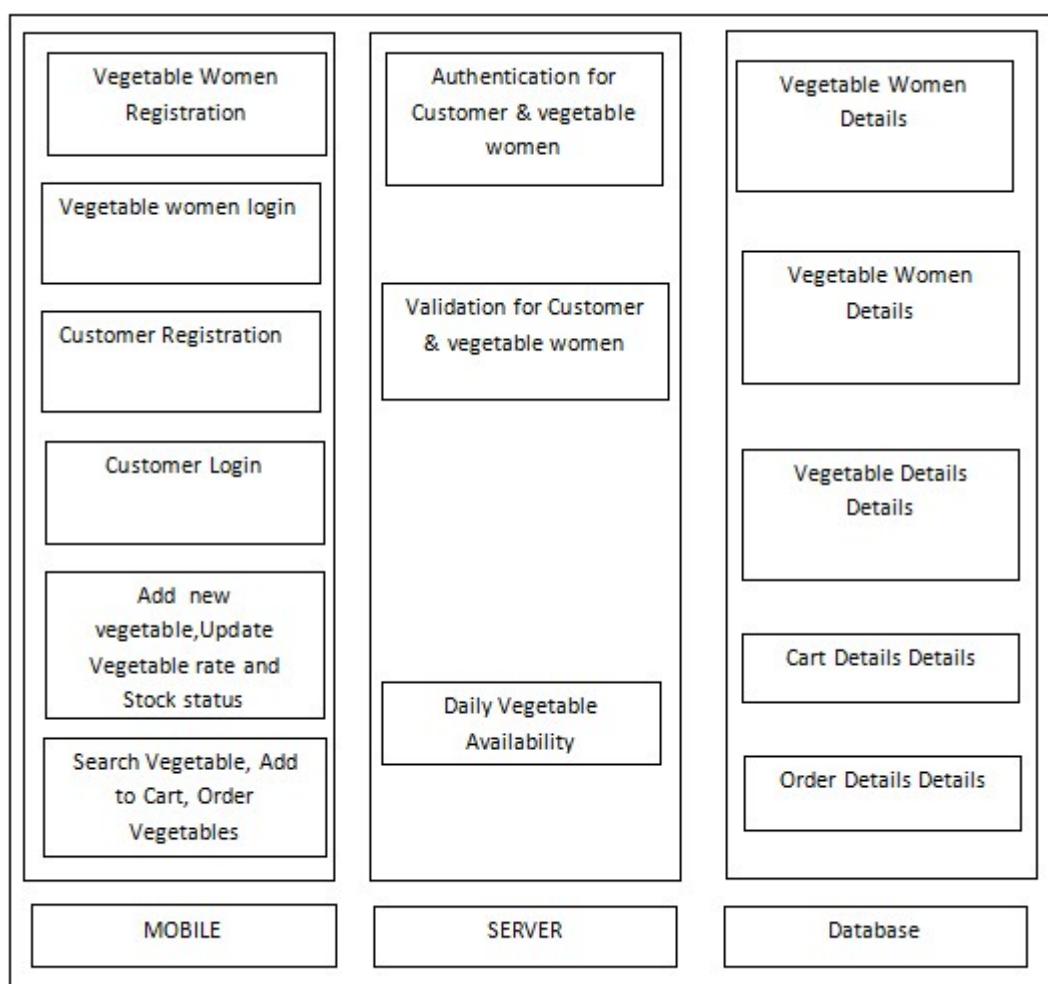
CHAPTER 5

SYSTEM DESIGN

5.1 ARCHITECTURAL DESIGN



Architectural Design for Vegetable Vendor System



CHAPTER 6

IMPLEMENTATION

HomeActivity.java

```
package com.app.greenroot;

import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.ActionBarDrawerToggle;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;
import androidx.appcompat.widget.Toolbar;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.app.greenroot.Model.Products;
import com.app.greenroot.Prevalent.Prevalent;
import com.app.greenroot.ViewHolder.ProductViewHolder;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.navigation.NavigationView;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;

import de.hdodenhof.circleimageview.CircleImageView;
import io.paperdb.Paper;

public class HomeActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener {
private DatabaseReference ProductsRef;
DrawerLayout drawerLayout;
NavigationView navigationView;
Toolbar toolbar;
TextView textView;
private RecyclerView recyclerView;
RecyclerView.LayoutManager layoutManager;

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

```
AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_YES);
ProductsRef = FirebaseDatabase.getInstance().getReference().child("Products");

drawerLayout=findViewById(R.id.drawer_layout);
navigationView=findViewById(R.id.nav_view);
toolbar=findViewById(R.id.toolbar);
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
toolbar.setTitle("Home");
setSupportActionBar(toolbar);

navigationView.bringToFront();
ActionBarDrawerToggle toggle=new
ActionBarDrawerToggle(this,drawerLayout,toolbar,R.string.navigation_drawer_open,R.string.navigation_
drawer_close);
drawerLayout.addDrawerListener(toggle);
toggle.syncState();
navigationView.setNavigationItemSelectedListener(this);

View headerView = navigationView.getHeaderView(0);
TextView userNameTextView = headerView.findViewById(R.id.user_profile_name);
CircleImageView profileImageView = headerView.findViewById(R.id.user_profile_image);
userNameTextView.setText(Prevalent.currentOnlineUser.getName());
Picasso.get().load(Prevalent.currentOnlineUser.getImage()).placeholder(R.drawable.profile).into(profileIm
ageView);
recyclerView = findViewById(R.id.recycler_menu);
recyclerView.setHasFixedSize(true);
layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent = new Intent(HomeActivity.this,CartActivity.class);
startActivity(intent);
}
});

}

@Override
protected void onStart() {
super.onStart();
FirebaseRecyclerOptions<Products> options =
new FirebaseRecyclerOptions.Builder<Products>()
.setQuery(ProductsRef.orderByChild("description").equalTo("In Stock"), Products.class)
.build();

FirebaseRecyclerAdapter<Products, ProductViewHolder> adapter =
new FirebaseRecyclerAdapter<Products, ProductViewHolder>(options) {
@Override
protected void onBindViewHolder(@NonNull ProductViewHolder holder, int position, @NonNull final
Products model)
{
holder.txtProductName.setText(model.getPname());
holder.txtProductDescription.setText(model.getDescription());
holder.txtProductPrice.setText("Rs." + model.getPrice());
holder.txtProductCount.setText(model.getCount());
}
};

recyclerView.setAdapter(adapter);
}

});
```

```

holder.txtProductCategory.setText(model.getCategory());
holder.txtSellerName.setText("Seller name: " + model.getSeller_name());
holder.txtSellerPhone.setText("Seller PhoneNumber: " + model.getSeller_phone());
holder.txtProductDescription.setText(model.getDescription());
holder.txtProductPrice.setText("Price = " + model.getPrice() + "Rs/Kg");
Picasso.get().load(model.getImage()).into(holder.imageView);
holder.itemView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(HomeActivity.this, ProductDetailsActivity.class);
        intent.putExtra("pid", model.getPid());
        intent.putExtra("seller_name", model.getSeller_name());

        startActivity(intent);
    }
});

@Override
public ProductViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.product_items_layout, parent, false);
    ProductViewHolder holder = new ProductViewHolder(view);
    return holder;
}
recyclerView.setAdapter(adapter);
adapter.startListening();

}

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

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

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    return super.onOptionsItemSelected(item);
}

```

```

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

    if (id == R.id.nav_cart) {
        Intent intent = new Intent(HomeActivity.this, CartActivity.class);
        startActivity(intent);
    } else if (id == R.id.nav_search) {
        Intent intent = new Intent(HomeActivity.this, SearchProductsActivity.class);
        startActivity(intent);
    } else if (id == R.id.nav_categories) {
        Intent intent = new Intent(HomeActivity.this, CustomerViewCategory.class);
        startActivity(intent);
    } else if (id == R.id.nav_settings) {
        Intent intent = new Intent(HomeActivity.this, SettinsActivity.class);
        startActivity(intent);
    } else if (id == R.id.nav_logout) {
        Paper.book().destroy();
        Intent intent = new Intent(HomeActivity.this, MainActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
        startActivity(intent);
        finish();
    }

    drawerLayout.closeDrawer(GravityCompat.START);
    return true;
}

```

VendorRegistrationActivity.java

```

package com.app.greenroot;

import android.app.ProgressDialog;
import android.content.Intent;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;

```

```

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import java.util.HashMap;

public class VendorRegisterActivity extends AppCompatActivity {
    private Button CreateAccountButton;
    private EditText InputName, InputPhoneNumber, InputPassword;
    private ProgressDialog loadingBar;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vendor_register);
        CreateAccountButton = (Button) findViewById(R.id.register_vendor_btn);
        InputName = (EditText) findViewById(R.id.register_vendor_username_input);
        InputPassword = (EditText) findViewById(R.id.register_vendor_password_input);
        InputPhoneNumber = (EditText) findViewById(R.id.register_vendor_phone_number_input);
        loadingBar = new ProgressDialog(this);
        CreateAccountButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                CreateAccount();
            }
        });
    }

    private void CreateAccount(){
        String name = InputName.getText().toString();
        String phone = InputPhoneNumber.getText().toString();
        String password = InputPassword.getText().toString();
        if(TextUtils.isEmpty(name))
        {
            Toast.makeText(this, "Please write your name...", Toast.LENGTH_SHORT).show();
        }
        else if(TextUtils.isEmpty(phone))
        {
            Toast.makeText(this, "Please write your phone number...", Toast.LENGTH_SHORT).show();
        }
        else if(TextUtils.isEmpty(password))
        {
            Toast.makeText(this, "Please write your password...", Toast.LENGTH_SHORT).show();
        }
        else
        {
            loadingBar.setTitle("Create Account");
            loadingBar.setMessage("Please wait, while we are checking the credentials.");
            loadingBar.setCanceledOnTouchOutside(false);
            loadingBar.show();
        }
    }
}

```

```

ValidatephoneNumber(name, phone, password);
}

}

private void ValidatephoneNumber(final String name, final String phone,final String password) {
final DatabaseReference RootRef;
RootRef = FirebaseDatabase.getInstance().getReference();
RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if (!(dataSnapshot.child("Vendor").child(phone).exists())){
HashMap<String, Object> vendordataMap = new HashMap<>();
vendordataMap.put("phone", phone);
vendordataMap.put("password", password);
vendordataMap.put("name", name);
RootRef.child("Vendor").child(phone).updateChildren(vendordataMap).addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {

if(task.isSuccessful())
{
Toast.makeText(VendorRegisterActivity.this, "Congratulations, your vendor account has been created.", Toast.LENGTH_SHORT).show();
loadingBar.dismiss();
Intent intent = new Intent(VendorRegisterActivity.this, com.app.greenroot.vendorLoginActivity.class);
startActivity(intent);
}
else
{
loadingBar.dismiss();
Toast.makeText(VendorRegisterActivity.this, "Network Error: Please try again after some time...", Toast.LENGTH_SHORT).show();
}
}
});

}

else {
Toast.makeText(VendorRegisterActivity.this, "This " + phone + " already exists.", Toast.LENGTH_SHORT).show();
loadingBar.dismiss();
Toast.makeText(VendorRegisterActivity.this, "Please try again using another phone number.", Toast.LENGTH_SHORT).show();
Intent intent = new Intent(VendorRegisterActivity.this, com.app.greenroot.MainActivity.class);
startActivity(intent);
}
}

@Override
public void onCancelled(DatabaseError databaseError) {

```

```
});  
}  
}  
}
```

VendorLoginActivity.java

```
package com.app.greenroot;  
  
import android.app.AlertDialog;  
import android.content.Intent;  
  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.app.greenroot.Model.Vendor;  
import com.app.greenroot.Prevalent.VendorPrevalent;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
import com.rey.material.widget.CheckBox;  
import io.paperdb.Paper;  
  
public class vendorLoginActivity extends AppCompatActivity {  
    private EditText InputPhoneNumber, InputPassword;  
    private Button LoginButton;  
    private ProgressDialog loadingBar;  
    private String parentDbName = "Vendor";  
    private CheckBox chkBoxRememberMe;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_vendor_login);  
        LoginButton = (Button) findViewById(R.id.login_btn);  
        InputPassword = (EditText) findViewById(R.id.login_password_input);  
        InputPhoneNumber = (EditText) findViewById(R.id.login_phone_number_input);  
        loadingBar = new ProgressDialog(this);  
        chkBoxRememberMe = (CheckBox) findViewById(R.id.remember_me_chkb);  
        Paper.init(this);  
        LoginButton.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                LoginUser();  
            }  
        });  
    }  
}
```

```

}

private void LoginUser()
{
String phone = InputPhoneNumber.getText().toString();
String password = InputPassword.getText().toString();

if(TextUtils.isEmpty(phone))
{
    Toast.makeText(this, "Please write your phone number...", Toast.LENGTH_SHORT).show();
}
else if(TextUtils.isEmpty(password))
{
    Toast.makeText(this, "Please write your password...", Toast.LENGTH_SHORT).show();
}
else
{
    loadingBar.setTitle("Login Account");
    loadingBar.setMessage("Please wait, while we are checking the credentials.");
    loadingBar.setCanceledOnTouchOutside(false);
    loadingBar.show();

    AllowAccessToAccount(phone, password);
}
}

private void AllowAccessToAccount(final String phone, final String password)
{
    if(chkBoxRememberMe.isChecked())
    {
        Paper.book().write(VendorPrevalent.UserPhoneKey, phone);
        Paper.book().write(VendorPrevalent.UserPasswordKey, password);
    }
    final DatabaseReference RootRef;
    RootRef = FirebaseDatabase.getInstance().getReference();
    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if(dataSnapshot.child(parentDbName).child(phone).exists()){

                Vendor userData = dataSnapshot.child(parentDbName).child(phone).getValue(Vendor.class);
                if(userData.getPhone().equals(phone))
                {
                    if(userData.getPassword().equals(password))
                    {
                        if(parentDbName.equals("Admins"))
                        {
                            Toast.makeText(vendorLoginActivity.this, "Welcome Vendor, you are logged in Successfully...", Toast.LENGTH_SHORT).show();
                            loadingBar.dismiss();
                        }
                    }
                }
            }
        }
    });
}

Intent intent = new Intent(vendorLoginActivity.this, VendorCategoryActivity.class);
startActivity(intent);
}
else if(parentDbName.equals("Vendor")){
}

```

```
Toast.makeText.vendorLoginActivity.this, "logged in Successfully...", Toast.LENGTH_SHORT).show();
loadingBar.dismiss();

Intent intent = new Intent.vendorLoginActivity.this, com.app.greenroot.VendorHomeActivity.class);
VendorPrevalent.currentOnlineUser = usersData;
startActivity(intent);
}

}
else {
loadingBar.dismiss();
Toast.makeText.vendorLoginActivity.this, "Password is incorrect", Toast.LENGTH_SHORT).show();
}
}
}
else {
Toast.makeText.vendorLoginActivity.this, "Account with this " + phone + " number do not exists.",
Toast.LENGTH_SHORT).show();
loadingBar.dismiss();
}
}

@Override
public void onCancelled(DatabaseError databaseError) {

}
});
```

AddNewProductActivity.java

```
package com.app.greenroot;

import android.app.AlertDialog;
import android.content.Intent;
import android.net.Uri;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

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

import com.app.greenroot.Prevalent.VendorPrevalent;
import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
```

```

import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import com.google.firebaseio.storage.FirebaseStorage;
import com.google.firebaseio.storage.StorageReference;
import com.google.firebaseio.storage.UploadTask;

import org.jetbrains.annotations.NotNull;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;
public class VendorAddNewProduct extends AppCompatActivity {
private String CategoryName, Description, Price, Pname, saveCurrentDate, saveCurrentTime,
sName, sPhone;
private Button AddNewProductButton;
private ImageView InputProductImage;
private EditText InputProductName, InputProductDescription, InputProductPrice;
private static final int GalleryPick = 1;
private Uri ImageUri;
private String productRandomKey, downloadImageUrl;
private StorageReference ProductImagesRef;
private DatabaseReference ProductsRef, sellersRef;
private ProgressDialog loadingBar;

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

CategoryName = getIntent().getExtras().get("category").toString();
ProductImagesRef = FirebaseStorage.getInstance().getReference().child("Product Images");
ProductsRef = FirebaseDatabase.getInstance().getReference().child("Products");
sellersRef = FirebaseDatabase.getInstance().getReference().child("Vendor");

AddNewProductButton = (Button) findViewById(R.id.add_new_product);
InputProductImage = (ImageView) findViewById(R.id.select_product_image);
InputProductName = (EditText) findViewById(R.id.product_name);
//InputProductDescription = (EditText) findViewById(R.id.product_description);
InputProductPrice = (EditText) findViewById(R.id.product_price);
loadingBar = new ProgressDialog(this);

sellersRef.child(VendorPrevalent.currentOnlineUser.getPhone())
.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(@NotNull @NotNull DataSnapshot dataSnapshot) {
if (dataSnapshot.exists()) {

```

```

//database reference to get vendor details
sName = dataSnapshot.child("name").getValue().toString();
sPhone = dataSnapshot.child("phone").getValue().toString();
}
}

@Override
public void onCancelled(@NonNull @NotNull DatabaseError error) {
}

InputProductImage.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
OpenGallery();
}
});

AddNewProductButton.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
ValidateProductData();
}
});
}

private void OpenGallery() {
Intent galleryIntent = new Intent();
galleryIntent.setAction(Intent.ACTION_GET_CONTENT);
galleryIntent.setType("image/*");
startActivityForResult(galleryIntent, GalleryPick);
}

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

if(requestCode == GalleryPick && resultCode == RESULT_OK && data != null) {
ImageUri = data.getData();
InputProductImage.setImageURI(ImageUri);
}
}

private void ValidateProductData() {
Price = InputProductPrice.getText().toString();
Pname = InputProductName.getText().toString();
if(ImageUri == null) {
Toast.makeText(this, "Product image is mandatory...", Toast.LENGTH_SHORT).show();
} else if(TextUtils.isEmpty(Price)) {
Toast.makeText(this, "Please write product Price...", Toast.LENGTH_SHORT).show();
} else if(TextUtils.isEmpty(Pname)) {
}
}

```

```

Toast.makeText(this, "Please write product name...", Toast.LENGTH_SHORT).show();
} else {
StoreProductInformation();
}

}

private void StoreProductInformation() {
loadingBar.setTitle("Add New Product");
loadingBar.setMessage("Dear Seller" + sName + ", please wait while we are adding the new product.");
loadingBar.setCanceledOnTouchOutside(false);
loadingBar.show();

Calendar calendar = Calendar.getInstance();

SimpleDateFormat currentDate = new SimpleDateFormat("MMM dd, yyyy");
saveCurrentDate = currentDate.format(calendar.getTime());

SimpleDateFormat currentTime = new SimpleDateFormat("HH:mm:ss a");
saveCurrentTime = currentTime.format(calendar.getTime());

productRandomKey = saveCurrentDate + saveCurrentTime;

final StorageReference filePath = ProductImagesRef.child(Uri.getLastPathSegment() +
productRandomKey + ".jpg");

final UploadTask uploadTask = filePath.putFile(Uri);

uploadTask.addOnFailureListener(new OnFailureListener() {
@Override
public void onFailure(@NonNull Exception e) {
String message = e.toString();
Toast.makeText(VendorAddNewProduct.this, "Error: " + message, Toast.LENGTH_SHORT).show();
loadingBar.dismiss();
}).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
@Override
public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
Toast.makeText(VendorAddNewProduct.this, "Product Image uploaded Successfully...", Toast.LENGTH_SHORT).show();
Task<Uri> urlTask = uploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
@Override
public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws Exception {
if (!task.isSuccessful()) {
throw task.getException();
}
}

downloadImageUrl = filePath.getDownloadUrl().toString();
return filePath.getDownloadUrl();
}).addOnCompleteListener(new OnCompleteListener<Uri>() {
@Override

```

```
public void onComplete(@NonNull Task<Uri> task) {
if(task.isSuccessful()) {
downloadImageUrl = task.getResult().toString();

Toast.makeText(VendorAddNewProduct.this, "got the Product image Url Successfully...", Toast.LENGTH_SHORT).show();

SaveProductInfoToDatabase();
}
}
});
}
});

}

private void SaveProductInfoToDatabase() {
HashMap<String, Object> productMap = new HashMap<>();
productMap.put("pid", productRandomKey);
productMap.put("date", saveCurrentDate);
productMap.put("time", saveCurrentTime);
productMap.put("description", "In Stock");
productMap.put("image", downloadImageUrl);
productMap.put("category", CategoryName);
productMap.put("price", Price);
productMap.put("pname", Pname);
productMap.put("seller_name", VendorPrevalent.currentOnlineUser.getName());
productMap.put("seller_phone", VendorPrevalent.currentOnlineUser.getPhone());

ProductsRef.child(productRandomKey).updateChildren(productMap).
addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if(task.isSuccessful()) {
Intent intent = new Intent(VendorAddNewProduct.this, VendorHomeActivity.class);
startActivity(intent);

loadingBar.dismiss();
Toast.makeText(VendorAddNewProduct.this, "Product is added successfully..",
Toast.LENGTH_SHORT).show();

} else {
loadingBar.dismiss();
String message = task.getException().toString();
Toast.makeText(VendorAddNewProduct.this, "Error: " + message, Toast.LENGTH_SHORT).show();
}
}
});

}

}
```

ManageProductActivity

```
package com.app.greenroot;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Toast;

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

import com.app.greenroot.Prevalent.VendorPrevalent;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;

import java.util.HashMap;

public class VendorManageProductDetailsActivity extends AppCompatActivity {

    private Button UpdateProductButton, DeleteProductButton;
    private EditText name,description,price;
    private ImageView imageview;
    private String productID="";
    private DatabaseReference productsRef;

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

        productID = getIntent().getStringExtra("pid");
        productsRef = FirebaseDatabase.getInstance().getReference().child("Products").child(productID);

        UpdateProductButton = (Button) findViewById(R.id.update_btn);
        DeleteProductButton = (Button) findViewById(R.id.delete_btn);

        imageview = findViewById(R.id.select_product_image);
        name = findViewById(R.id.product_name);
        description = findViewById(R.id.product_description);
        price = findViewById(R.id.product_price);

        displayProductInfo();
        UpdateProductButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
```



```

Intent intent = new Intent(VendorManageProductDetailsActivity.this, VendorHomeActivity.class);
startActivity(intent);
finish();
}

});

}

}

private void displayProductInfo() {
productsRef.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if (dataSnapshot.exists()){
String pName = dataSnapshot.child("pname").getValue().toString();
String pDescription = dataSnapshot.child("description").getValue().toString();
String pPrice = dataSnapshot.child("price").getValue().toString();

name.setText(pName);
price.setText(pPrice);
description.setText(pDescription);

}
}

@Override
public void onCancelled(DatabaseError databaseError) {
}

});
}
}
}

```

AddToCartActivity

```

package com.app.greenroot;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

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

```

```

import com.app.greenroot.Model.Products;
import com.app.greenroot.Prevalent.Prevalent;
import com.cepheuen.elegantnumberbutton.view.ElegantNumberButton;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import com.squareup.picasso.Picasso;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;

public class ProductDetailsActivity extends AppCompatActivity {
    private Button addToCartButton;
    private ImageView productImage;
    private ElegantNumberButton numberButton;
    private TextView productPrice,productDescription,productName;
    private String productID = "", state = "Normal";
    private String seller_name;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_product_details);
        productID = getIntent().getStringExtra("pid");
        seller_name = getIntent().getStringExtra("seller_name");
        addToCartButton = (Button) findViewById(R.id.pd_add_to_cart_button);
        numberButton = (ElegantNumberButton) findViewById(R.id.number_btn);
        productImage = (ImageView) findViewById(R.id.product_image_details);
        productName = (TextView) findViewById(R.id.product_name_details);
        productDescription = (TextView) findViewById(R.id.product_description_details);
        productPrice = (TextView) findViewById(R.id.product_price_details);
        getProductDetails(productID);
        addToCartButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                if (state.equals("Order Placed") || state.equals("Order Shipped")){
                    Toast.makeText(ProductDetailsActivity.this,"You can add Purchase more product, once your order is shipped or confirmed",Toast.LENGTH_LONG).show();
                }
                else
                {
                    addingToCartList();
                }
            }
        });
    }

    @Override

```

```

protected void onStart() {
    super.onStart();
    CheckOrderState();
}

private void addingToCartList() {
    String saveCurrentTime,saveCurrentDate;
    Calendar calForDate = Calendar.getInstance();
    SimpleDateFormat currentDate = new SimpleDateFormat("MMM dd, yyyy");
    saveCurrentDate = currentDate.format(calForDate.getTime());
    SimpleDateFormat currentTime = new SimpleDateFormat("HH:mm:ss a");
    saveCurrentTime = currentDate.format(calForDate.getTime());
    final DatabaseReference cartListRef = FirebaseDatabase.getInstance().getReference().child("Cart List");
    final HashMap<String, Object>cartMap = new HashMap<>();
    cartMap.put("pid",productID);
    cartMap.put("seller name",seller name);
    cartMap.put("pname",productName.getText().toString());
    cartMap.put("price",productPrice.getText().toString());
    cartMap.put("date",saveCurrentDate);
    cartMap.put("time",saveCurrentTime);
    cartMap.put("quantity",numberButton.getNumber());
    cartMap.put("discount", "");

    cartListRef.child("User
view").child(Prevalent.currentOnlineUser.getPhone()).child("Products").child(productID).updateChildren(c
artMap).addOnCompleteListener(new OnCompleteListener<Void>() {
    @Override
    public void onComplete(@NonNull Task<Void> task) {
        if(task.isSuccessful()){
            cartListRef.child("Admin view").child(Prevalent.currentOnlineUser.getPhone())
            .child("Products").child(productID)
            .updateChildren(cartMap)
            .addOnCompleteListener(new OnCompleteListener<Void>() {
                @Override
                public void onComplete(@NonNull Task<Void> task) {
                    if(task.isSuccessful()){
                        Toast.makeText(ProductDetailsActivity.this,"Added to cart List",Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(ProductDetailsActivity.this,CartActivity.class);
                        intent.putExtra("seller_name", seller_name);
                        startActivity(intent);
                    }
                }
            });
        }
    }
});
}
}

}

private void getProductDetails(String productID) {
    DatabaseReference productsRef = FirebaseDatabase.getInstance().getReference().child("Products");
    productsRef.child(productID).addValueEventListener(new ValueEventListener() {
        @Override

```

```
public void onDataChange(DataSnapshot dataSnapshot) {
if (dataSnapshot.exists()){
Products products=dataSnapshot.getValue(Products.class);
productName.setText(products.getPname());
productPrice.setText(products.getPrice());
productDescription.setText(products.getDescription());
Picasso.get().load(products.getImage()).into(productImage);

}
}

@Override
public void onCancelled(DatabaseError databaseError) {

}
});

}

//



private void CheckOrderState()
{
DatabaseReference ordersRef;
ordersRef =
FirebaseDatabase.getInstance().getReference().child("Orders").child(Prevalent.currentOnlineUser.getPhone());
ordersRef.addValueEventListener(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if (dataSnapshot.exists()){
String shippingState = dataSnapshot.child("state").getValue().toString();
if (shippingState.equals("Shipped")){
state ="Order Delivered";
}
else if (shippingState.equals("Not Shipped")){
state ="Order Placed";
}
}
}

@Override
public void onCancelled(DatabaseError databaseError) {

}
});
}

{
```

ConfirmOrderActivity

```

package com.app.greenroot;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.app.greenroot.Model.Cart;
import com.app.greenroot.Prevalent.CartViewHolder;
import com.app.greenroot.Prevalent.Prevalent;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebaseio.ui.database.FirebaseRecyclerOptions;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class CartActivity extends AppCompatActivity{
private RecyclerView recyclerView;
private RecyclerView.LayoutManager layoutManager;
private Button NextProcessBtn;
private TextView txtTotalAmount, txtMsg1;
private int overTotalPrice=0;
private String seller_name,product_name,product_quantity;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_cart);
recyclerView = findViewById(R.id.cart_list);
recyclerView.setHasFixedSize(true);
layoutManager = new LinearLayoutManager(this);
recyclerView.setLayoutManager(layoutManager);
NextProcessBtn = (Button)findViewById(R.id.next_btn);
txtTotalAmount = (TextView)findViewById(R.id.total_price);
txtMsg1 = (TextView)findViewById(R.id.msg1);

seller_name = getIntent().getStringExtra("seller_name");
product_name = getIntent().getStringExtra("product_name");
product_quantity = getIntent().getStringExtra("product_quantity");
}

```

```

NextProcessBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        txtTotalAmount.setText("Total Price = Rs." + String.valueOf(overTotalPrice));
        Intent intent = new Intent(CartActivity.this, ConfirmFinalOrderActivity.class);
        intent.putExtra("Total Price", String.valueOf(overTotalPrice));
        intent.putExtra("seller_name", seller_name);
        intent.putExtra("product_name", product_name);
        intent.putExtra("product_quantity", product_quantity);

        startActivity(intent);
        finish();
    }
});

@Override
protected void onStart() {
    super.onStart();
    CheckOrderState();
    final DatabaseReference cartListRef = FirebaseDatabase.getInstance().getReference().child("Cart List");
    FirebaseRecyclerOptions<Cart> options =
            new FirebaseRecyclerOptions.Builder<Cart>()
                    .setQuery(cartListRef.child("User view")
                            .child(Prevalent.currentOnlineUser.getPhone()).child("Products"), Cart.class).build();
    FirebaseRecyclerAdapter<Cart, CartViewHolder> adapter
            = new FirebaseRecyclerAdapter<Cart, CartViewHolder>(options) {
        @Override
        protected void onBindViewHolder(@NonNull CartViewHolder holder, int position, @NonNull final Cart model) {
            holder.txtProductQuantity.setText("Quantity = " + model.getQuantity());
            holder.txtProductPrice.setText("Price = " + model.getPrice() + " Rs.");
            holder.txtProductName.setText(model.getPname());
            holder.txtSellerName.setText("Seller name:" + model.getSeller_name());
            int oneTyprProductTPrice = ((Integer.valueOf(model.getPrice())) * Integer.valueOf(model.getQuantity()));
            overTotalPrice = overTotalPrice + oneTyprProductTPrice;
            int PricePerProduct = oneTyprProductTPrice;

            holder.itemView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    CharSequence options[] = new CharSequence[]
                            {
                                "Edit",
                                "Confirm",
                                "Remove"
                            };
                    AlertDialog.Builder builder = new AlertDialog.Builder(CartActivity.this);
                    builder.setTitle("Cart Options:");
                    builder.setItems(options, new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialogInterface, int i) {
                            if (i == 0) {
                                Intent intent = new Intent(CartActivity.this, ProductDetailsActivity.class);

```

```

intent.putExtra("pid", model.getPid());
startActivity(intent);
}
if(i==1){
txtTotalAmount.setText("Total Price = Rs." +String.valueOf(PricePerProduct));
Intent intent = new Intent(CartActivity.this, com.app.greenroot.ConfirmFinalOrderActivity.class);
intent.putExtra("Total Price", String.valueOf(PricePerProduct));
intent.putExtra("product_name", model.getPname());
intent.putExtra("product_quantity", model.getQuantity());
intent.putExtra("seller_name", model.getSeller_name());
startActivity(intent);

}
if (i==2){
cartListRef.child("User view")
.child(Prevalent.currentOnlineUser.getPhone())
.child("Products")
.child(model.getPid())
.removeValue()
.addOnCompleteListener(new OnCompleteListener<Void>() {
@Override
public void onComplete(@NonNull Task<Void> task) {
if (task.isSuccessful()){
Toast.makeText(CartActivity.this, "Item removed Successfully.", Toast.LENGTH_SHORT).show();
Intent intent = new Intent(CartActivity.this, com.app.greenroot.HomeActivity.class);
startActivity(intent);
}
}
});
}
});
}
});
}
builder.show();
}
});
}
}

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

recyclerView.setAdapter(adapter);
adapter.startListening();
}
private void CheckOrderState()
{
DatabaseReference ordersRef;
ordersRef =
FirebaseDatabase.getInstance().getReference().child("Orders").child(Prevalent.currentOnlineUser.getPhone());
ordersRef.addValueEventListener(new ValueEventListener() {

```

```

@Override
public void onDataChange(DataSnapshot dataSnapshot) {
if(dataSnapshot.exists()){
String shippingState = dataSnapshot.child("state").getValue().toString();
String userName = dataSnapshot.child("name").getValue().toString();
if(shippingState.equals("Delivered")){
txtTotalAmount.setText("Dear "+userName+"\n order is delivered successfully.");
recyclerView.setVisibility(View.GONE);
txtMsg1.setVisibility(View.VISIBLE);
txtMsg1.setText("Congratulations, Your Final order has been added successfully. Soon you will received your order at your door step.");
NextProcessBtn.setVisibility(View.GONE);
Toast.makeText(CartActivity.this,"You can purchase more products, Once you received your first order",Toast.LENGTH_SHORT).show();
}
else if(shippingState.equals("Not Delivered")){
txtTotalAmount.setText("Delivery State = Not Delivered");
recyclerView.setVisibility(View.GONE);
txtMsg1.setVisibility(View.VISIBLE);

NextProcessBtn.setVisibility(View.GONE);
Toast.makeText(CartActivity.this,"You can purchase more products, Once you received your first order",Toast.LENGTH_SHORT).show();
}
}
}
}

```

```

@Override
public void onCancelled(DatabaseError databaseError) {

}
});
}
}
}

```

Full Source code Link:

<https://bitbucket.org/linetmshajii/vegetable-vendor/src/code/Vegetable%20Vendor-Multiple%20Seller%20based/app/src>

CHAPTER 7

TESTING

If your app is already installed on the device, simply click Apply Changes in the toolbar to update the app with the new layout. Or click Run to install and run the app. The button still does nothing. To start another activity when the button is tapped, we can troubleshoot it and can run it again using an android device or using android emulator. Each module can be unit tested and at last an integration testing can be done using the following methods

7.1 TEST PLAN

Each module can be unit tested after completing each activity. After opening new activity and after completing it we can execute a unit testing. And at the end by combining all the module and perform an integration testing.

7.2 UNIT TESTING

Each unit like login, register and other module can be individually tested by connecting with an android device by the following steps :

1. Connect your device to your development machine with a USB cable. If you're developing on Windows, you might need to install the appropriate USB driver for your device.
2. Enable USB debugging in the Developer options as follows. First, you must enable the developer options:
3. Open the Settings app. b. (Only on Android 8.0 or higher) Select System.
4. Scroll to the bottom and select About phone. d. Scroll to the bottom and tap Build number 7 times.
5. Return to the previous screen to find Developer options near the bottom. Open Developer options, and then scroll down to find and enable USB debugging.

Unit testing is performed using android EMULATOR by the following steps:

1. In Android Studio, click the app module in the Project window and then select Run > Run (or click Run in the toolbar).
2. In the Select Deployment Target window, click Create New Virtual Device.
3. In the Select Hardware screen, select a phone device, such as Pixel, and then click Next.
4. In the System Image screen, select the version with the highest API level. If you don't have that version installed, a Download link is shown, so click that and complete the download.
5. Click Next.
6. On the Android Virtual Device (AVD) screen, leave all the settings alone and click Finish.
7. Back in the Select Deployment Target dialog, select the device you just created and click OK.

7.3 INTEGRATION TESTING

Integration testing can be performed using an android device or an android emulator after completing more module and making a combination process and running it.

7.4 VALIDATION TESTING

The validation is done for email, password and confirm password in the registration module. These can be performed with an android device or using the android emulator by giving the inputs. If we give wrong input the error message will be popped up. Or when the field is empty then the error message will be popped up as the field is empty.

CHAPTER 8

CONCLUSION

As I to wish work as software developer this project help me to learn how to build android application and how use the cloud database called Firebase.Please thanks to DR.A MARTIN to giving this opportunity to me.

BIBLIOGRAPHY □

- <https://developer.android.com> □
- <https://firebase.google.com/docs/database>
- <https://developer.android.com/studio/intro>
- <https://www.tutorialspoint.com/android>
- <https://youtu.be/WINjBmKKpR4>
- <https://www.androidauthority.com>
- <https://abhiandroid.com/androidstudio> □

Reference book:

- Android Programming Concepts Book by Richard Cornez and Trish Cornez

11:49 AM



VeggiesFruits

Potato

25

In Stock

UPDATE

DELETE



11:47 AM



VeggiesFruits

Close

Update



Change Profile

Phone Number...

Full Name...

Address...



11:50 AM



VeggiesFruits

Potato

SEARCH



11:51 AM



VeggiesFruits

Total Price

Potato

Quantity = 2

Price = 25 Rs.

Seller name:Shaji John

CLICK HERE ONLY IF ALL
ORDERS ARE FROM SAME
SELLER



VeggiesFruits

Products

Ginger

Quantity = 4

Price = 25 Rs.

Product Seller Name

Orange

Quantity = 4

Price = 50 Rs.

Product Seller Name



11:39 AM



VeggiesFruits

Total Price

Ginger

Quantity = 1

Price = 25 Rs.

Seller name:Linet

Cart Options:

Edit

Confirm

Remove

CLICK HERE ONLY IF ALL
ORDERS ARE FROM SAME
SELLER



11:51 AM



VeggiesFruits

Delivery State = Not Delivered

Congratulations, Your Final order has
been placed successfully. Soon it will
be verified

You can purchase more products,
Once you received your first order



VeggiesFruits

**Welcome, Click icons to Add
New Products**



11:39 AM



VeggiesFruits

Total Price

Ginger

Quantity = 1

Price = 25 Rs.

Seller name:Linet

CLICK HERE ONLY IF ALL
ORDERS ARE FROM SAME
SELLER



VeggiesFruits

New Orders

Product: Potato

Quantity: 2

Customer Name: Linet M Shaji

Phone: 7510840607

Total Amount = Rs.50

**Shipping Address: Mangalath House, Thavinal P. O,
Muthireri, Mananthavady**

Order at: May 19. 2021 May 19. 2021

**SHOW USER ORDER
PRODUCT**



11:48 AM



VeggiesFruits

Close

Update



Change Profile

9947788654

Shaji John

Mangalath House, Thavinhal P. O,
Mananthavady



11:42 AM



VeggiesFruits

Orange

SEARCH

Orange



VeggiesFruits

Welcome, Click icons to Add New Products



Back

Logout



VeggiesFruits

New Orders

Product: Potato

Quantity: 2

Customer Name: Linet M Shaji

Phone: 7510840607

Total Amount = Rs.50

Shipping Address: Mangalath House, Thavinhal P. O,

Have you delivered this order
products?

Muthireri, Mananthavady
Order at: May 19. 2021 May 19. 2021

Yes

**SHOW USER ORDER
PRODUCT**

No



11:40 AM



VeggiesFruits

Delivery State = Not Delivered

Congratulations, Your Final order has
been placed successfully. Soon it will
be verified





Potato



Price = 25Rs.

In Stock

Root

Product is added successfully..



VeggiesFruits

Total Price

Ginger

Quantity = 4

Price = 25 Rs.

Seller name:Linet

Orange

Quantity = 4

Price = 50 Rs.

Seller name:Linet

Spinach

Quantity = 4

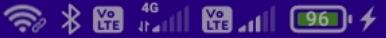
Price = 10 Rs.

Seller name:Sonat

CLICK HERE ONLY IF ALL
ORDERS ARE FROM SAME
SELLER



11:35 AM



VeggiesFruits

Welcome to Vegroot, Login as Customer



Phone Number

Password

Login



Ginger



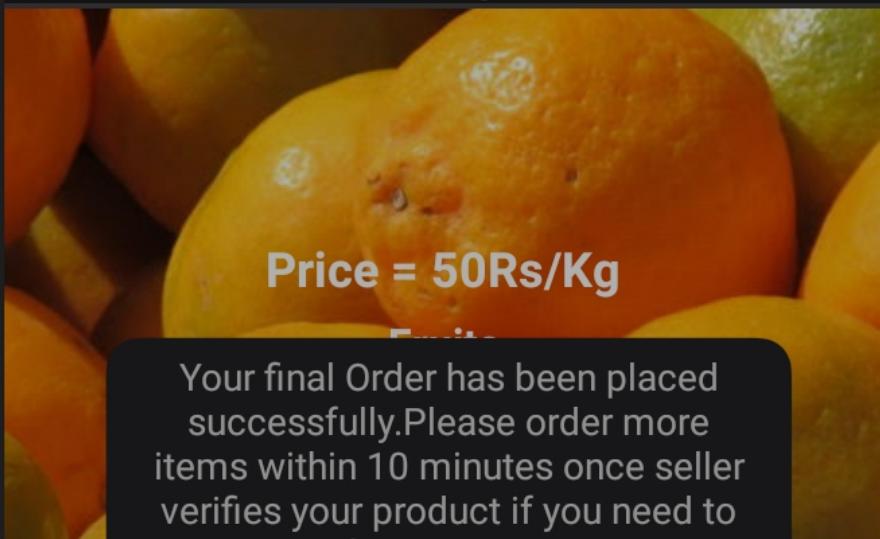
Price = 25Rs/Kg

**Root
In Stock**

Seller name: Linet

Seller Phone Number: 7510840607

Orange



Price = 50Rs/Kg

Your final Order has been placed successfully. Please order more items within 10 minutes once seller verifies your product if you need to buy from same seller

Spinach





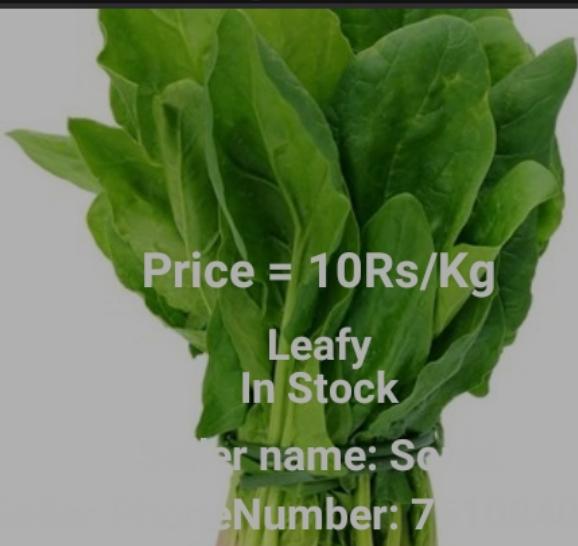
Price = 50Rs/Kg

Fruits
In Stock

Seller name: Linet

Seller Phone Number: 7510840607

Spinach



Price = 10Rs/Kg

Leafy
In Stock

Seller name: Se

Seller Phone Number: 7

Potato



Price = 25Rs/Kg

Root
In Stock

Seller name: Shaji John

Seller Phone Number: 8086237422



