

Shopping Buddy

Software Requirements Specification Document

Shopping Buddy

Software Requirements Specification

Version 1.1

11/10/2019

Vennela Cheekoti

Sathwika Gone

Susritha Gade

Rakesh Avirineni

Yaswanth Yarram

Sandeep Devineni

Mehar Choudhary Meenavilli

Submitted in partial fulfillment

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

Shopping Buddy

Revision History

Date	Description	Author	Comments
06/11/2019	Version 1.0	Vennela Cheekoti Sathwika Gone Susritha Gade Rakesh Avirineni Yaswanth Yarram Sandeep Devineni Mehar Choudhary Meenavilli	First Revision
11/10/2019	Version 1.1	Susritha Gade Yaswanth Yarram	Second version
12/1/2019	Version 1.2	Susritha Gade Yaswanth Yarram	Third Version

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Shopping Buddy

Signature	Printed Name	Title	Date
	Dr. Zhengrui Qin	Project Requirements	06/12/2019
	Dr. Zhengrui Qin	ER Diagram	06/21/2019
	Dr. Zhengrui Qin	Data Flow Diagram Sequence Diagram	12/01/2019
	Dr. Zhengrui Qin	UI designs	12/01/2019

Table of Contents	Page Number
1. Introduction	6
1.1. Purpose	6
1.2. Scope	6
1.3. Definitions, Acronyms, and Abbreviations	6
1.4. References	7
1.5. Overview	7
2. General Description	8
2.1. Product Perspective	8
2.2. Product Functions	8
2.3. User Characteristics	9
2.4. General Constraints	10
2.5. Assumptions and Dependencies	10
3. Specific Requirements	11
3.1. External Interface Requirements	11
3.1.1. User Interfaces	11
3.1.2. Hardware Interfaces	11
3.1.3. Software Interfaces	11
3.1.4. Communications Interface	12
3.2. Functional Requirements	12
3.3. Use Cases	16

Shopping Buddy

3.4. Non-Functional Requirements	16
3.4.1. Performance	16
3.4.2. Reliability	17
3.4.3. Availability	17
3.4.4. Security	17
3.4.5. Portability	17
4. Design	18
4.1. ER diagram	18
4.2. GUI	19
5. Analysis Models	27
5.1. Data Flow Diagram	27
5.2. Sequence Diagram	28
6. Technical Manual (User)	29
7. Technical Manual (Admin)	62
8. User Manual (Admin and User)	89
9. Testing	97

1. Introduction

1.1. Purpose

The purpose of this project is to create a user-friendly e-commerce application which allows the user to shop online and place the order and receive those orders.

1.2. Scope

This is an e-commerce application with a login page, if the user is new to the application then he will be allowed to register. User can browse through the different categories (Clothing, Household, Furniture, etc.) and place their order after that user will be redirected to the payment, once payment is done user order is placed successfully and the user will know about when their order will reach. This app also has admin functionality which is to add products to the application so that user can add go through and buy the products, edit /delete products, add/edit/delete promo codes.

1.3. Definitions, Acronyms, and Abbreviations

Definition	Abbreviation
UI	User Interface
ERD	Entity Relationship Diagram
URD	User Requirement Document
SRD	Software Requirement Document

1.4. References

Developers, G. (n.d.). *Firebase Documentation*. Retrieved from
<https://firebase.google.com/docs/android/setup>

Developers, G. (n.d.). *Android Studio Documentation*. Retrieved from
<https://developer.android.com/docs>

1.5. Overview

This document is divided into five chapters, Introduction, General Description, Specific requirements, Design and Analysis Models.

In the Introduction it describe the purpose and scope of the project.

The General Description gives a clear and concise description of product prospective. It consists of a short view on the product, a short summary of the general capabilities of the product and more in-depth view of the main parts of the product.

The Specific requirements chapter will describe all the external interface requirements, functional requirements, use cases, class/object definitions and non-functional requirements. Moreover, here it will describe the user constraints, logical database requirements, and prototype and use case diagrams.

The Design chapter will describe the ER diagram and the UI designs.

The last chapter will describe the Data Flow diagram and Sequence design.

2. General Description

2.1 Product Perspective

Shopping Buddy is an application used for online shopping developed using open source technologies. This application provides a simple mechanism for users to buy new products and order them online. Some of the main features included in this application are

- User Account: Our application allows the user to create their accounts in the system and provide features of updating viewing their profiles.
- Features & Specifications: The user can get to know about every detail of a particular product.
- Number of users being supported by the application: Though the number is not mentioned but our application supports a good number of users at a time.
- Review: The user is able to review the particular product before the user orders.

Since this is a data-centric product it will need a database to store the data. We add and modify the data whenever needed. All the database communication will go over the internet.

2.2 Product Functions

The product functions are divided into two tasks: Admin and User

Admin Tasks:

1. Login
2. Add product
3. Edit Product
4. Add promo
5. Edit promo

Shopping Buddy

6. Check Added products, deleted products, added promo, and deleted promo
7. Inventory (All products)
8. Chat with user.

User/Customer tasks:

1. Login
2. Look for products from different categories
3. Send/reply messages
4. Change his/her password
5. Use deals (saves their money)
6. Share Products
7. View Orders
8. Make Payment
9. Enter Shipping Address

2.3 User Characteristics

- Two types of user characteristics: Admin and User/Customer
- Admin has access to add, edit and delete products whereas user can only view the products and buy products.
- Admin can add, edit and delete promo codes whereas user can only view those promo codes and use them while purchasing the product.
- User can change his/her password and admin can also change password.
- Admin and user can directly message using messaging characteristics.

2.4 General Constraints

The internet connection is a constraint for this application. Since the application fetches the data from the database over the internet, it is important to make sure that there is an internet connection for the application to function.

2.5 Assumptions and Dependencies

We assume that the application will always work on mobile phones that have enough performance. If the mobile does not have enough hardware resources available for the application, like if the users might have allocated them with other applications, there may be situations where the application does not work.

Dependencies required for this application are Java language, Android Studio, Firebase, GitHub version control system.

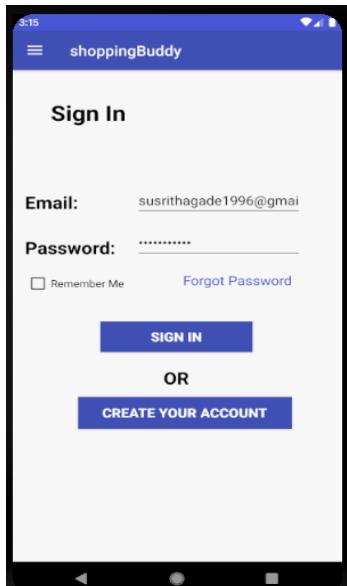
3. Specific Requirements

3.1. External Interface Requirements

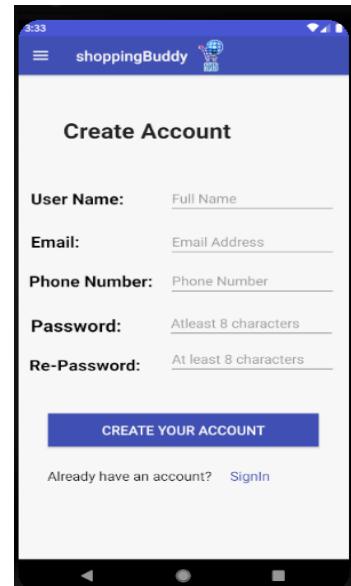
3.1.1. User Interfaces:

A good user interface provides a "user-friendly" experience, allowing the user to interact with the software or hardware in a natural and intuitive way. The GUI of our software program includes many controls like the menu bar, Input selector, keyboard, icons, images, and buttons.

User Login Screen



User Registration Screen



3.1.2. Hardware Interfaces:

This application works on Android device (Samsung, pixel, etc...). No other hardware is required.

3.1.3. Software Interfaces:

Android apps can be written using Kotlin, Java, and C++ languages. The Android SDK tools compile your code along with any data and resource files into an APK, an *Android package*, which is an archive file with an .apk suffix.

Shopping Buddy

- The Android operating system is a multi-user Linux system in which each app is a different user.
- By default, the system assigns each app a unique Linux user ID (the ID is used only by the system and is unknown to the app). The system sets permissions for all the files in an app so that only the user ID assigned to that app can access them.
- Each process has its own virtual machine (VM), so an app's code runs in isolation from other apps.
- By default, every app runs in its own Linux process. The Android system starts the process when any of the app's components need to be executed, and then shuts down the process when it's no longer needed or when the system must recover memory for other apps.

3.1.4. Communications Interface:

Google Docs, sheets and slides:

Google Docs, sheets and slides lends itself to collaborative projects in which multiple authors work together in real time from geographically diverse locations. All participants can see who made specific document changes and when those alterations were done. Because documents are stored online and can also be stored on users' computers, there is no risk of total data loss as a result of a localized catastrophe. However, the Internet-based nature of Google Docs, sheets and slides has given rise to concerns among some authors that their work may not be private or secure.

We use this for creating project documents, sheets and slides for presentation. It is reliable for team-work.

Outlook:

It is Microsoft email service. We use outlook to send and receive mails. Used to contact team members.

3.2 Functional Requirements

Admin Requirements

1. Admin login

Administrator should be able to log in to the system with admin privileges using his admin username and password

Admin username	shoppingbuddyseven@gmail.com
Password	Minimum of 6 characters At least one number or symbol (!, #, %, & *) (admin7)

2. Add new products:

Administrator should be able to add new items to the list. Most recently added item should be displayed on the top of the inventory list. New item should have following fields.

Product ID	Product Name	Description	Category	Cost	Image

3. Edit/Remove Products:

Administrator should be able to edit or remove any existing items on the list. When editing, he can edit all the fields except ID.

4. Add new promo Codes:

Shopping Buddy

Administrator should be able to add new promo codes to the list. Most recently added item should be displayed on the top of the inventory list. New item should have following fields.

Promo ID	Product Code	Description	Amount to Deduct

5. Edit/Remove Products:

Administrator should be able to edit or remove any existing promo code on the list. When editing, he can edit all the fields except ID.

6. Check Added products, deleted products, added promo, and deleted promo:

Administrator should be able to view the products that he/she added and deleted and also promo codes that are added and deleted

7. Inventory (All products):

Administrator should be able to see all the products that added are to the application in purpose for the users to buy.

8. Chat with User:

Administrator should be able to chat with user if in case user messages.

User Requirements:

1. Register:

When a potential user wants to buy anything from Shopping Buddy, he/she should fill the sign up form. When a user registers for the system, he/she should provide the following details.

Shopping Buddy

User Name	Email	Phone Number	Password	Re- Enter Password
-----------	-------	--------------	----------	--------------------

2. Log in to Shopping Buddy - Android

Once user is done with registration then user can log in to the Shopping Buddy– Android using his email address and password.

3. Recover lost or forgotten password

If user lost or forgot the password, he can recover the password by entering his/her email then the link to reset password will be sent to their email.

4. View the Products and details

Once the user log in to the Shopping Buddy– Android, he/she can view the list of products of different categories. In the initial list it shows the product name and price.

Product Name	Product Price
--------------	---------------

If user wants to see more details about a specific product, he can tap on the product and view other information about the products.

5. Share Products:

User can share the products he/she likes with others, so that even others can give their opinion about the product and even buy if they need the same product.

6. Add to cart:

User can add the products to the cart and can buy the products.

7. Shipping Address and Payment:

User can enter his/her shipping address and then he/she can do payment in order to buy the product. If the payment is done then the user is done with purchasing the product.

8. Orders:

User can view all the orders that he/she has done.

9. Settings:

User can change his/her password, and can also contact admin by using the email or he/she can message admin.

10. Logout

3.3. Use Cases

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. A use case can be thought of as a collection of possible scenarios related to a particular goal, indeed, the use case and goal are sometimes considered to be synonymous.

A use case (or set of use cases) has these characteristics:

- Organizes functional requirements
- Models the goals of system/actor (user) interactions
- Records paths (called *scenarios*) from trigger events to goals
- Describes one main flow of events (also called a basic course of action), and possibly other ones, called *exceptional* flows of events (also called alternate courses of action)
- Is multi-level, so that one use case can use the functionality of another one.

3.4. Non-Functional Requirements

3.4.1. Performance:

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.

3.4.2. Reliability:

Requirements about how often the software fails. The measurement is often expressed in MTBF (mean time between failures). The definition of a failure must be clear. Specify the consequences of software failure, how to protect from failure, a strategy for error detection, and a strategy for correction.

3.4.3. Availability:

Availability is gauged by the period of time that the system's functionality and services are available for use with all operations.

3.4.4. Security:

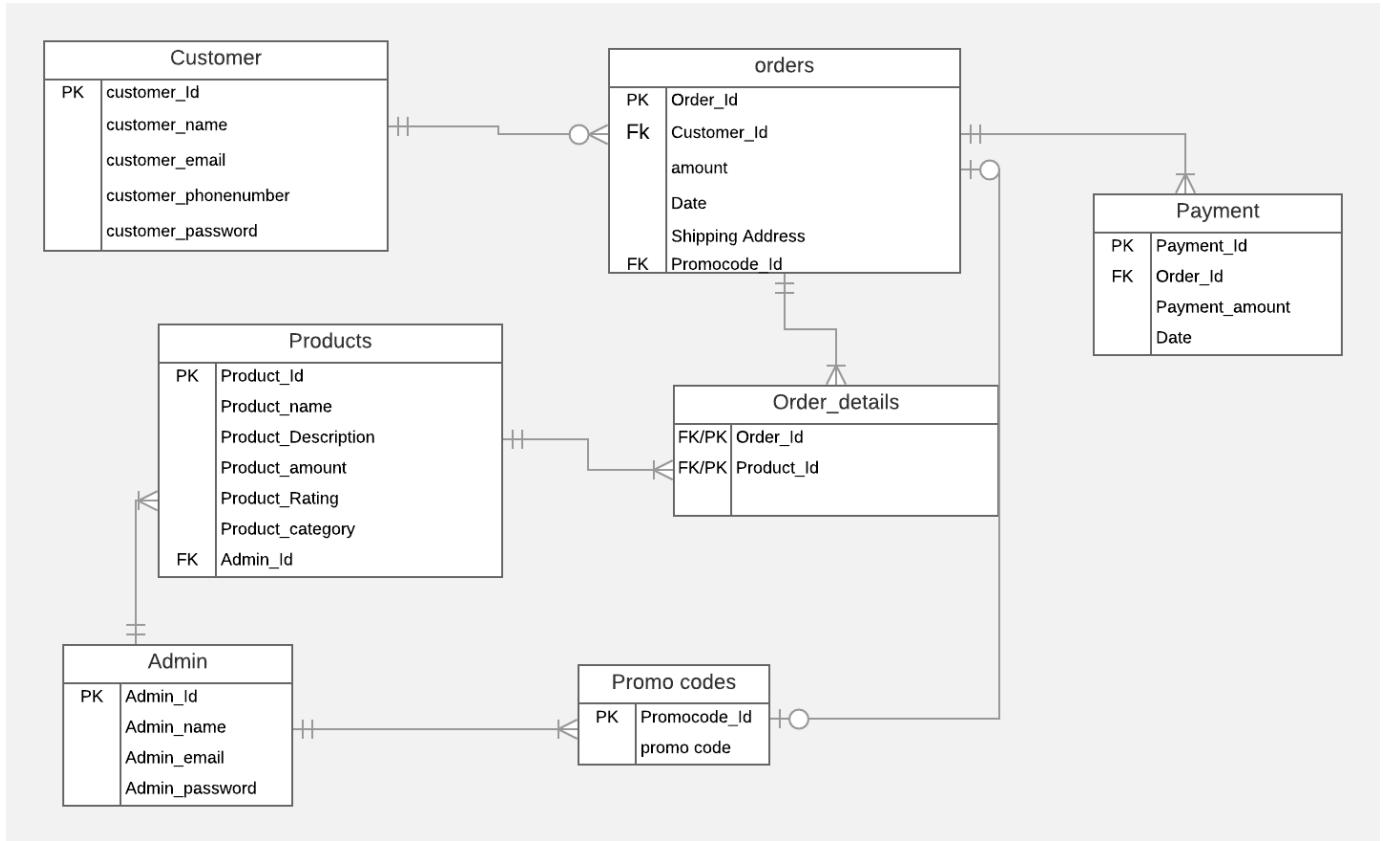
One or more requirements about the protection of your system and its data. The measurement can be expressed in a variety of ways (effort, skill level, and time) to break into the system.

3.4.5. Portability:

The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

4. Design

4.1 ER Diagram:



In the above Entity Relationship diagram, it has 7 entities namely Customer, Products, Admin, Orders, Order_details, Promo codes and payment. The customer can be considered as user and admin can be considered as seller. From the above diagram we can also see that we have only one associative entity that is Order_details which is a weak entity. Customer_id, Product_id, Admin_id, Order_id, Promocode_id, Payment_id are primary and foreign keys.

Relationships:

Customer: Orders - 1: M

Orders: Payment - 1: M

Orders: Order_details - 1: M

Shopping Buddy

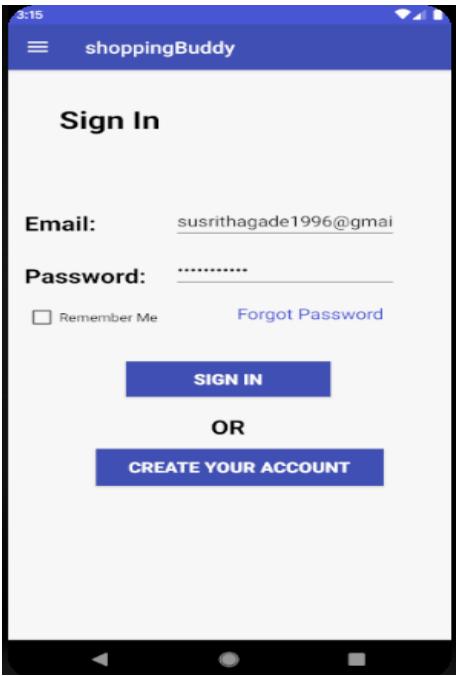
Product: Order_details - 1: M

Orders: Promo codes - 1:1

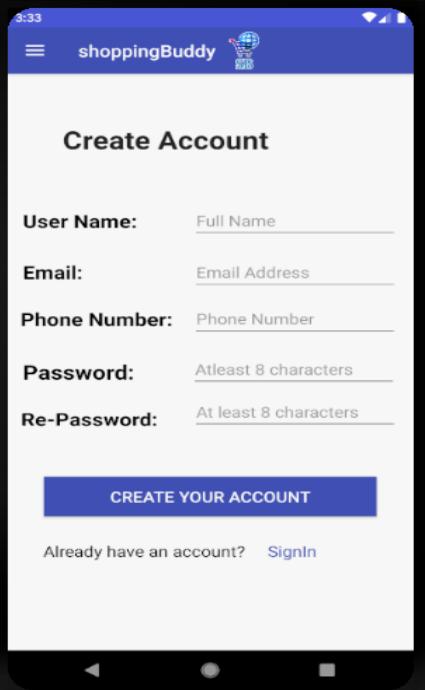
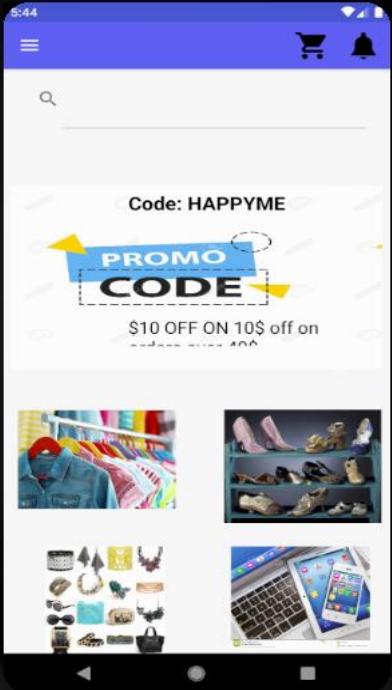
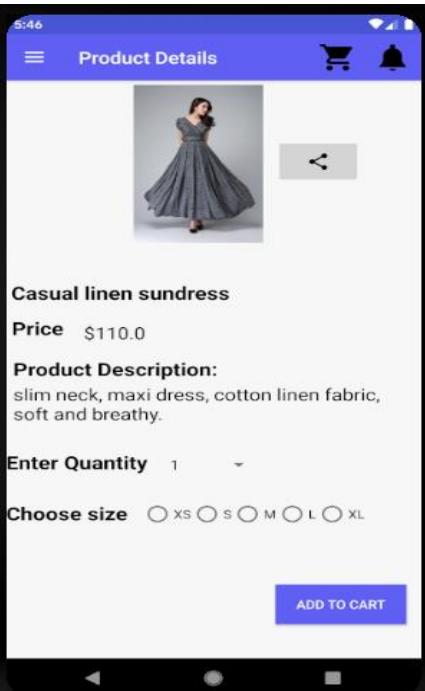
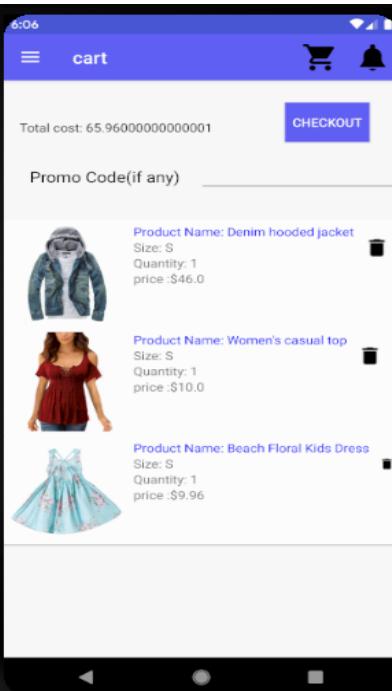
Product: Admin - 1: M

Admin: Promo codes - 1:M

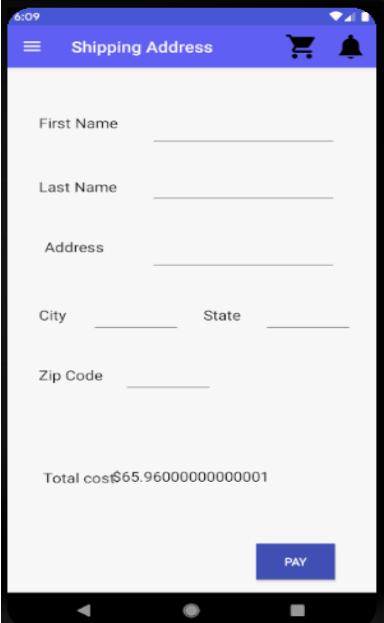
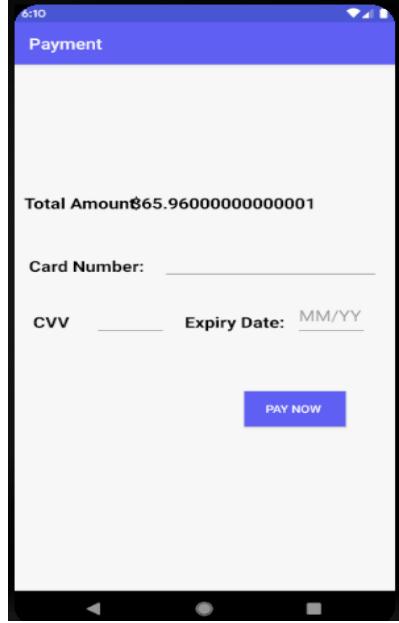
4.2 GUI

Login Page	Forgot Password
	
Register	Home Screen

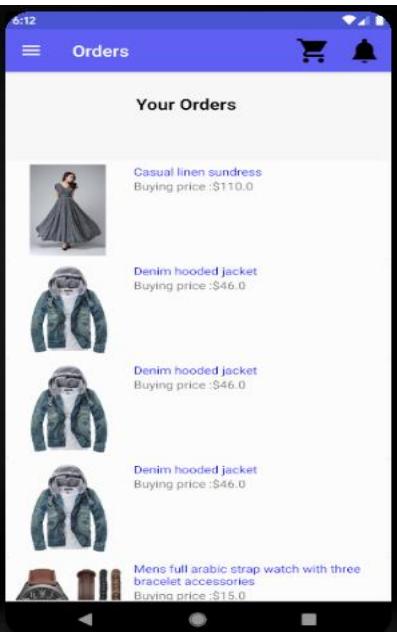
Shopping Buddy

 The 'Create Account' screen shows fields for User Name (Full Name), Email, Phone Number, Password (Atleast 8 characters), and Re-Password (At least 8 characters). A 'CREATE YOUR ACCOUNT' button is at the bottom, and a 'Signin' link is below it.	 The home screen displays a promotional banner for 'HAPPYME' with a 'PROMO CODE' section offering '\$10 OFF ON 10\$ off on'. It also shows four categories of products: clothing, shoes, accessories, and electronics.
Item Details	Cart
 The 'Product Details' screen shows an image of a woman in a grey dress, product name 'Casual linen sundress', price '\$110.0', and a detailed description: 'slim neck, maxi dress, cotton linen fabric, soft and breathy.' It includes a dropdown for 'Enter Quantity' (set to 1), size selection ('Choose size' with options xs, s, m, l, xl), and an 'ADD TO CART' button.	 The 'cart' screen shows a total cost of '65.96000000000001' and a 'CHECKOUT' button. It lists three items: 'Denim hooded jacket' (size S, quantity 1, price \$46.0), 'Women's casual top' (size S, quantity 1, price \$10.0), and 'Beach Floral Kids Dress' (size S, quantity 1, price \$9.96). Each item has a delete icon next to its details.

Shopping Buddy

Shipping Address	Payment
	
Orders	Settings(Change Password)

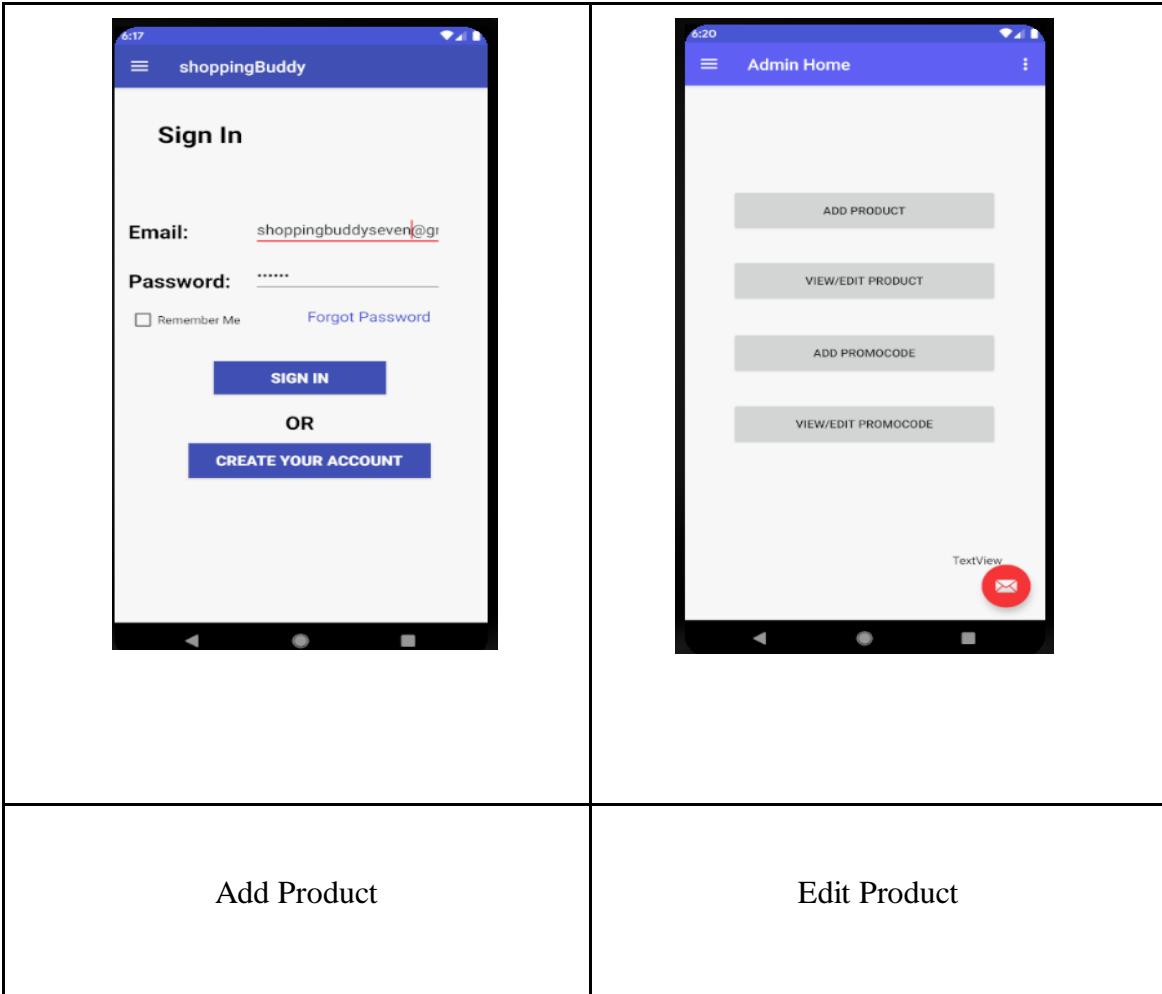
Shopping Buddy



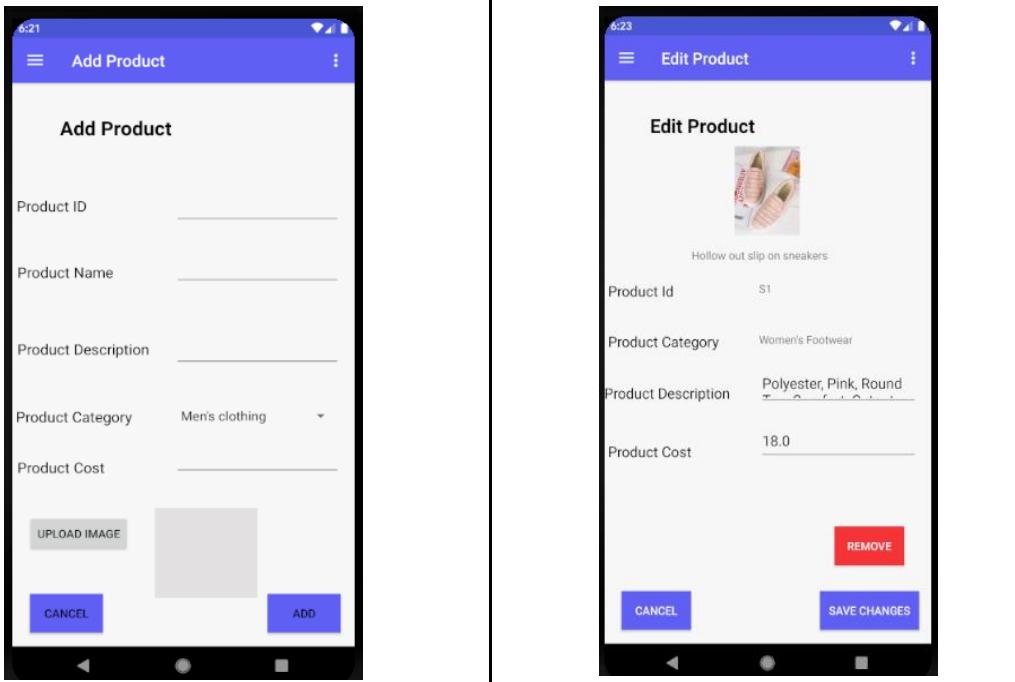
Admin Login

Admin Home

Shopping Buddy



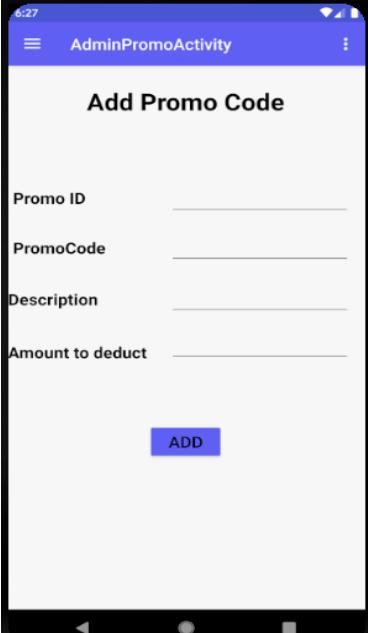
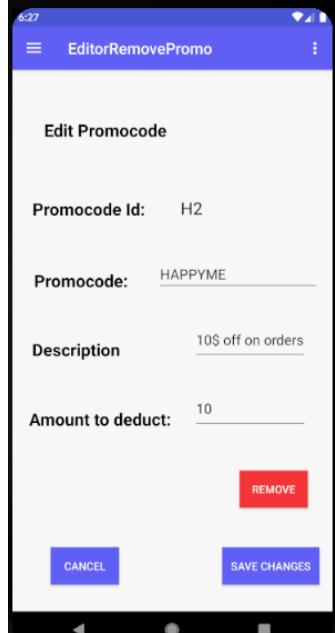
Shopping Buddy



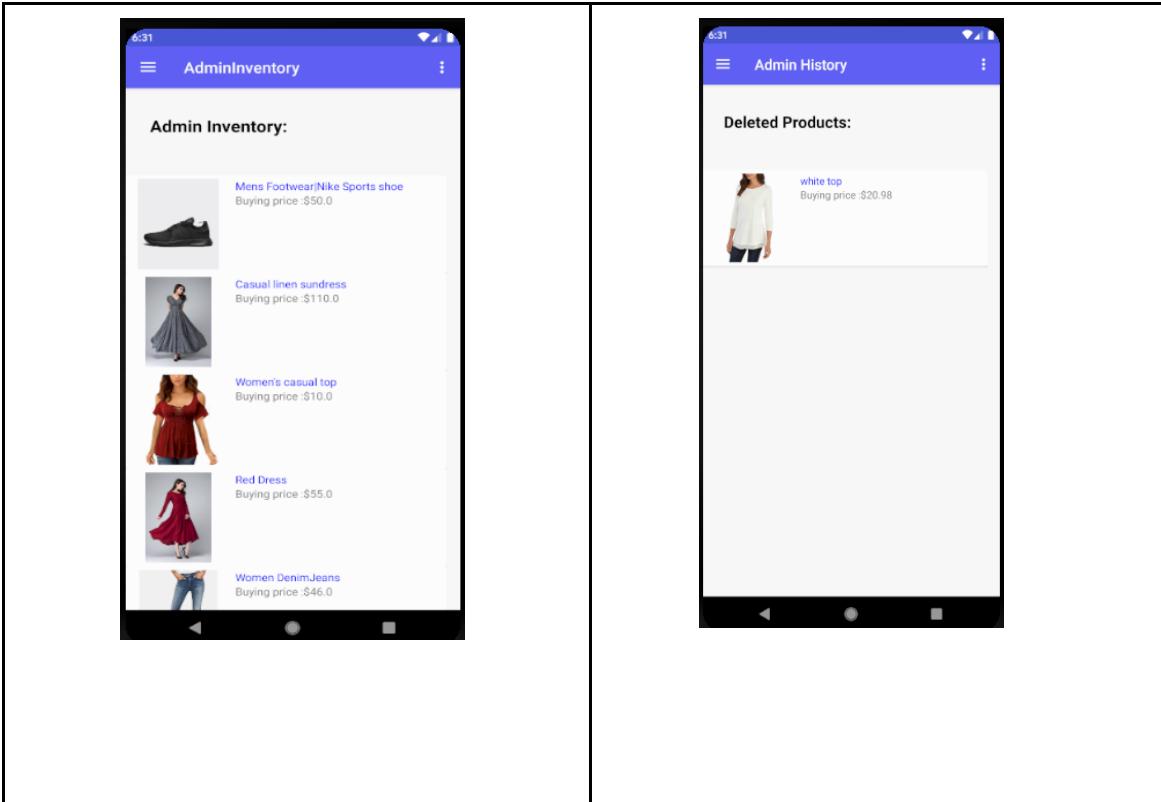
Add Promo Code

Edit/Remove Promo Code

Shopping Buddy

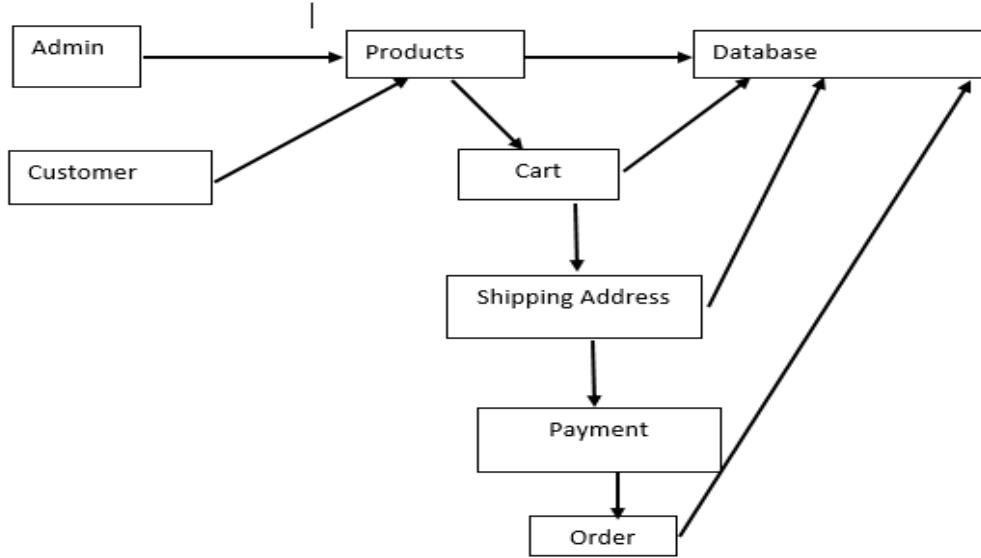
	
Admin Inventory	Admin

Shopping Buddy

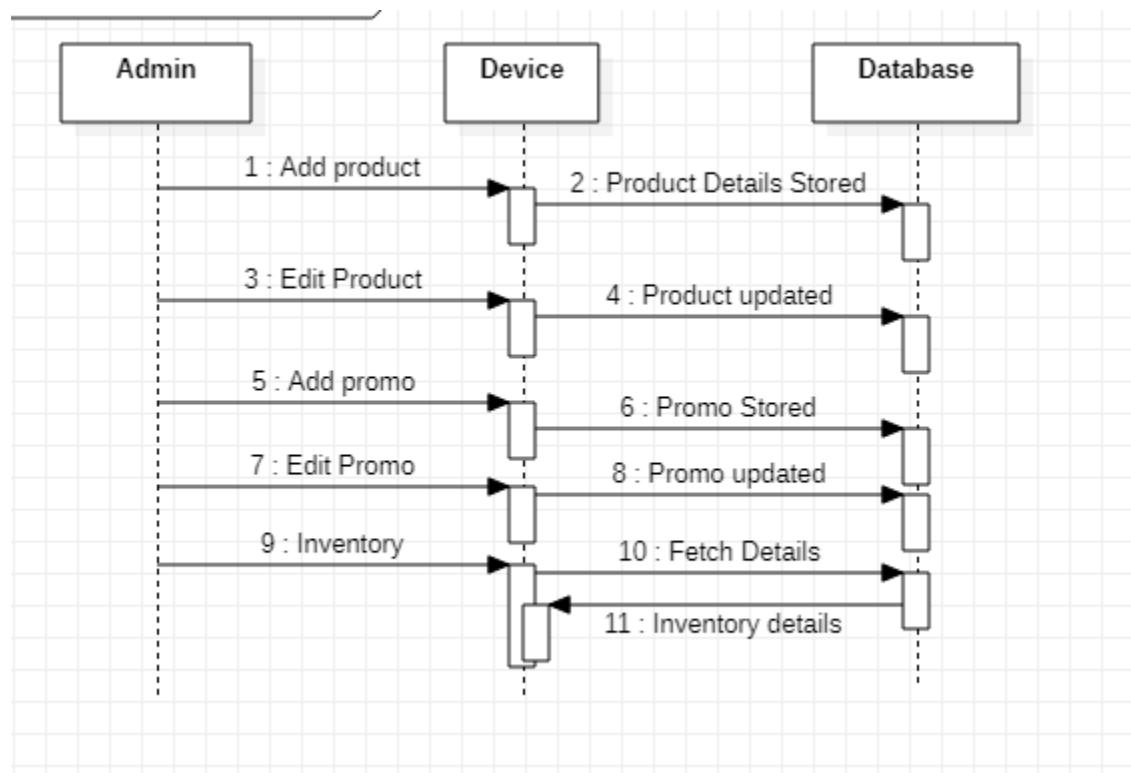


5. Analysis Models

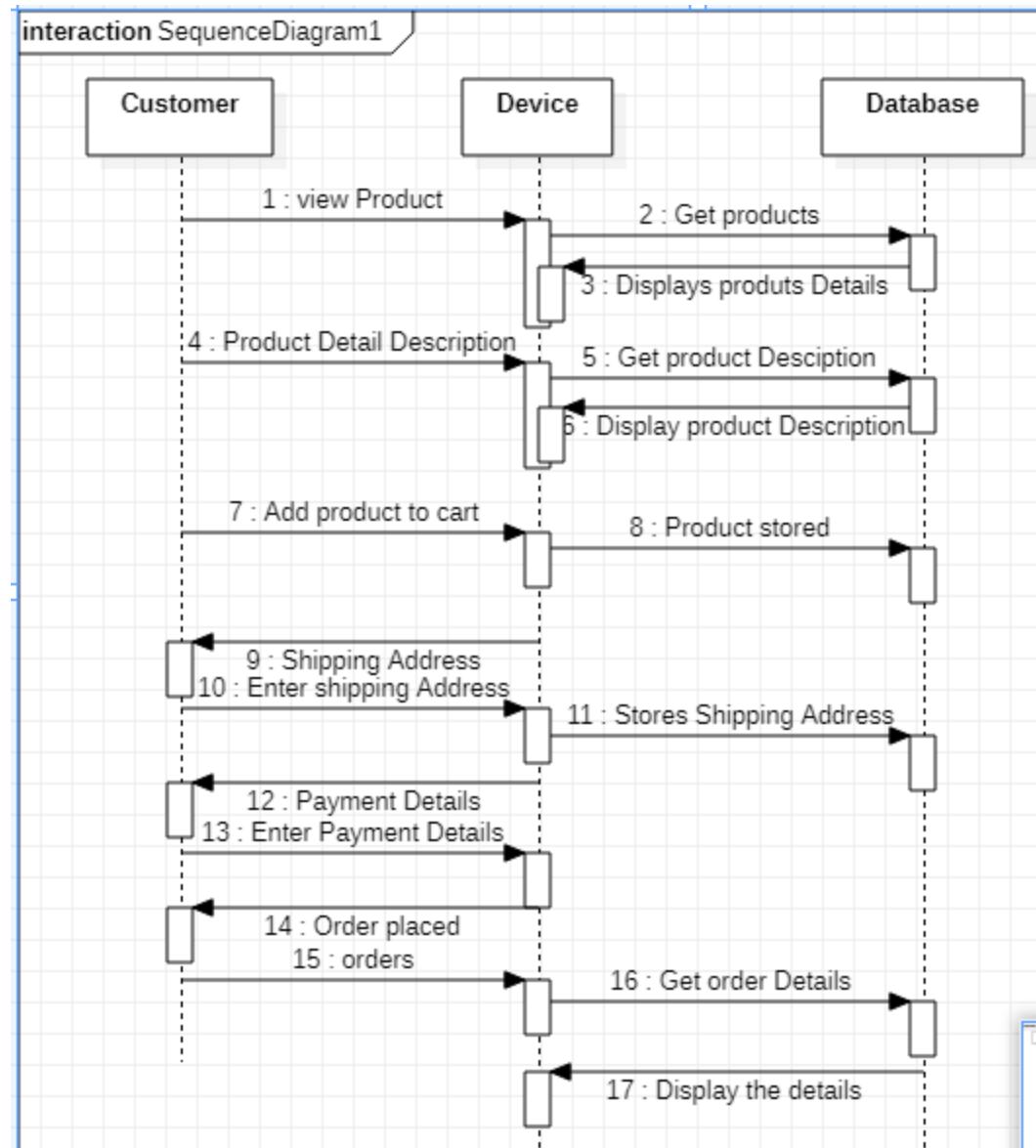
5.1 Data Flow Diagram:



5.2 Sequence Diagram:



Shopping Buddy



Shopping Buddy - Android App

Version 2.0.0

Technical Manual

Team : Android Team

Course: 44692-04

Class : GDP 02 Spring 2019

TEAM MEMBERS

Sathwika Gone

Vennela Cheekoti

Rakesh Avirineni

Yaswanth Yarram

Mehar Choudhry Meenavilli

Sandeep Devineni

Susritha Gade

1 Introduction

1.1. Document Identification

This document describe the technical aspect of design and implementation of Shopping Buddy App for android based mobile phone. This document is prepared by Android team of GDP 01 (44692-04) Fall 2019.

1.2. System overview

Shopping Buddy -Android is developed for android base mobile devices to make it possible for user to buy the products from different categories which are added to application by admin. This app consists of two parts those are admin and user. Users can download the application through internet and use it to buy the products. Admin will use the applications to add the products, reply to user messages/chats and other administrative tasks.

1.3. Tools and technologies

1.3.1. Android Studio 3.2.1

Android Studio 3.2.1 is used to develop the User Interfaces (UIs) -Activities, and frontend and backend functionalities.

1.3.2 Firebase

Google Firebase is used for backend data handling, storing and manipulation. Also Authentication, Email verification and password recovery functionalities are implemented using firebase functionalities.

Firebase Account Credentials

User Name :shoppingbuddyseven@gmail.com

Password : sevenbuddies

1.3.3 GitHub

GitHub repository is used for version control and collaborating purpose of the development process. All the implementation and documentation work were uploaded to the GitHub by each member of the team as they work.

GitHub link : <https://github.com/SathwikaGone/Shopping-Buddy>

1.4 Document overview

This document will describe the technical aspect of each activity, functionality and associations separately. The list and order of the items will be as follows.

1. Sign Up for user
2. Login for user
3. Password recovery for user
4. User logout

Documentation will be continued as the functionalities are implemented....

2. User Signup

2.1. Requirement

Once the user download the app, he/she can see the registration page. Then user can provide the necessary information and sign up for the app. Then the user will get a verification email to his email account and he must verify his email in order to get access to the app.

2.2 Technical information

- i. App is connected to the Firebase and activated the Authentication with email and password authentication.

The screenshot shows a step-by-step guide for setting up Firebase Authentication. It includes sections for connecting the app to Firebase, adding authentication, checking current auth state, and signing up new users. Each section contains explanatory text and code snippets.

Email and password authentication

You can use Firebase Authentication to let your users sign in with their email addresses and passwords, and to manage your app's password-based accounts. This tutorial helps you set up an email and password system and then access information about the user.

[Launch in browser](#)

- ① Connect your app to Firebase
Connected
- ② Add Firebase Authentication to your app
[Add Firebase Authentication to your app](#)

To use an authentication provider, you need to enable it in the [Firebase console](#). Go to the Sign-in Method page in the Firebase Authentication section to enable Email/Password sign-in and any other identity providers you want for your app.
- ③ Check current auth state
Declare an instance of `FirebaseAuth`

```
private FirebaseAuth mAuth;
```

In the `onCreate()` method, initialize the `FirebaseAuth` instance.

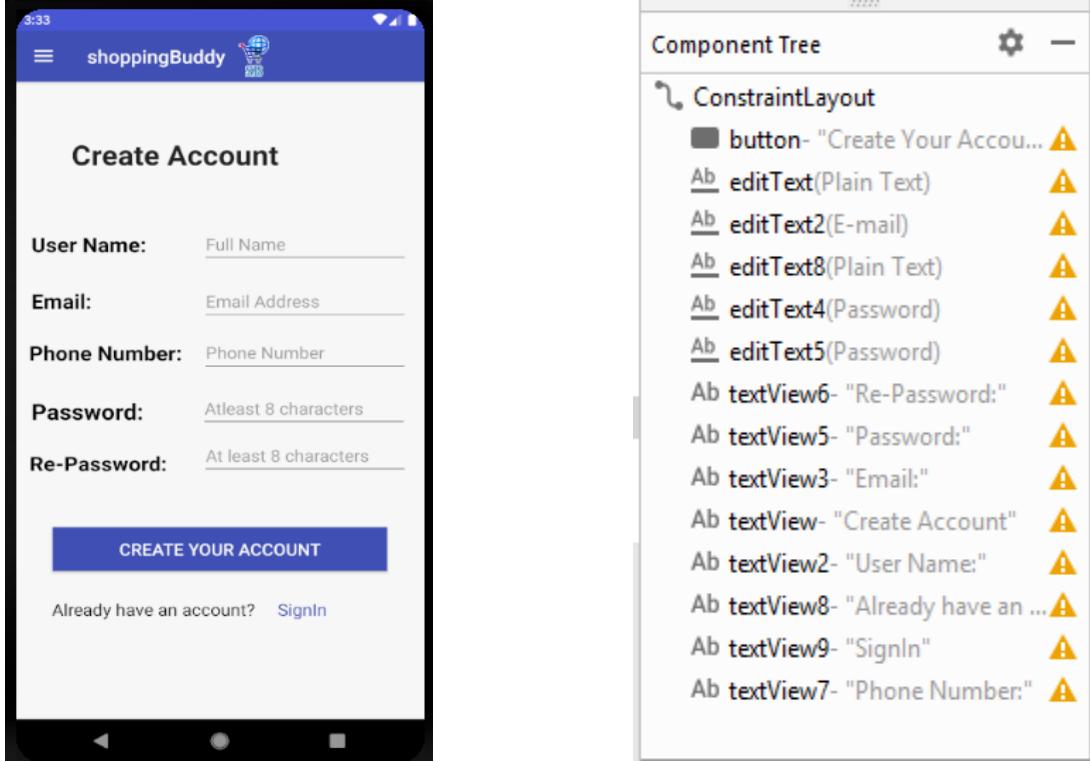
```
mAuth = FirebaseAuth.getInstance();
```

When initializing your Activity, check to see if the user is currently signed in.

```
@Override
public void onStart() {
    super.onStart();
    // Check if user is signed in (non-null) and update UI accordingly
    FirebaseUser currentUser = mAuth.getCurrentUser();
    updateUI(currentUser);
}
```
- ④ Sign up new users
Create a new `createAccount` method which takes in an email address and password, validates them and then creates a new user with the `createUserWithEmailAndPassword` method.

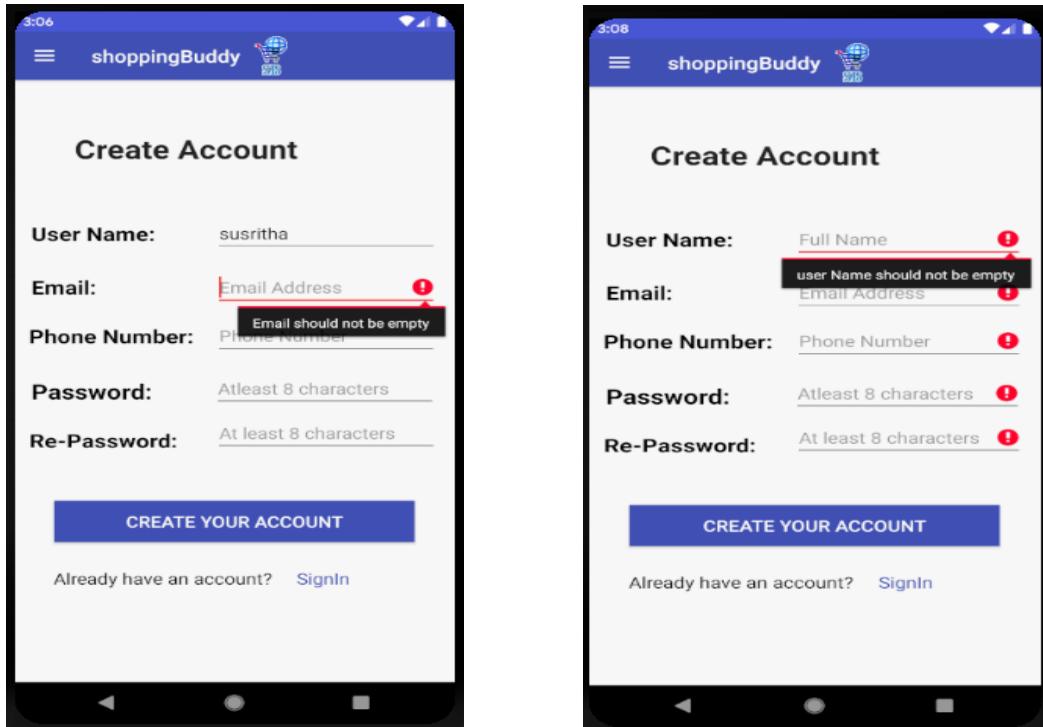
Shopping Buddy

ii. Signup activity is created on Android studio



iii. User need to fill all the information on signup page, otherwise it will send message stating that field should not be empty.

Shopping Buddy

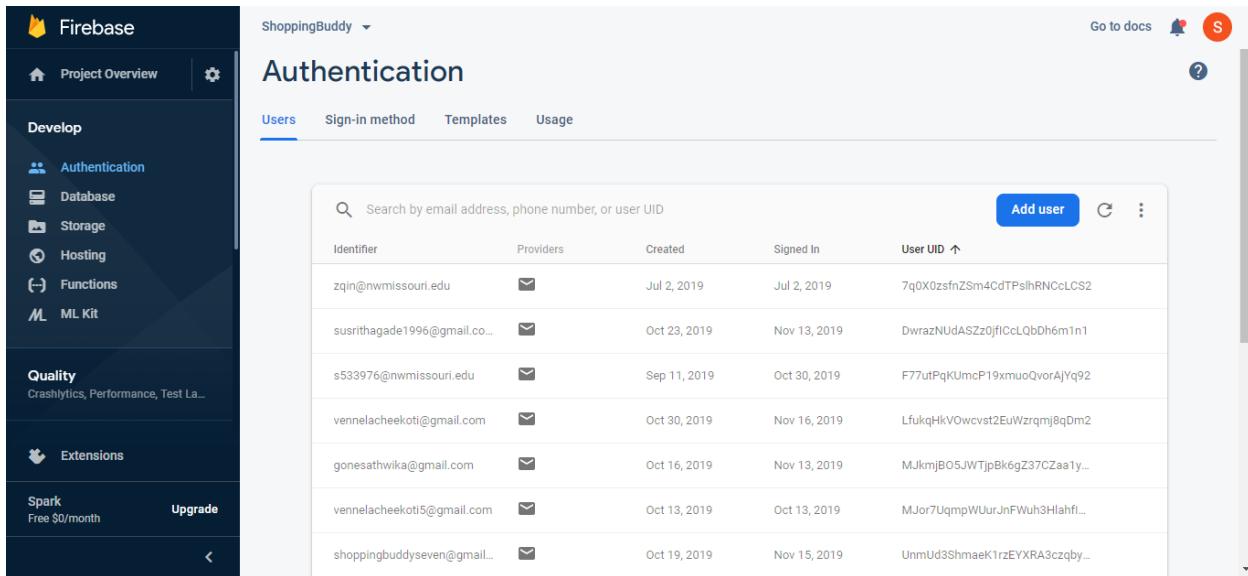


iv. Once user fill all the information and click on continue, it will create a record on the firebase authentication and will send an email to the user for verification.

```
firebaseAuth.createUserWithEmailAndPassword(em, pw).addOnCompleteListener( activity: this, (task) -> {
    if (task.isSuccessful()) {
        FirebaseAuth auth = FirebaseAuth.getInstance();
        FirebaseUser user = auth.getCurrentUser();
        user.sendEmailVerification()
            .addOnCompleteListener((task) -> {
                if (task.isSuccessful()) {
                    Toast.makeText( context: MainActivity.this, text: "Email sent", Toast.LENGTH_LONG).show();
                    Log.d( tag: "success" , msg: "email sent");
                    Intent i = new Intent( packageContext: MainActivity.this, SigninActivity.class);
                    startActivity(i);
                }
            });
    } else {
        Toast.makeText( context: MainActivity.this, text: "Could not register please try again", Toast.LENGTH_LONG).show();
    }
});
db = FirebaseFirestore.getInstance();
itemCollection = db.collection( collectionPath: "users");
Map<String, Object> adduser = new HashMap<>();
adduser.put("Username",uname );
adduser.put("Email", em);
adduser.put("Phonenumber", phone);
itemCollection.document().set(adduser);
```

Shopping Buddy

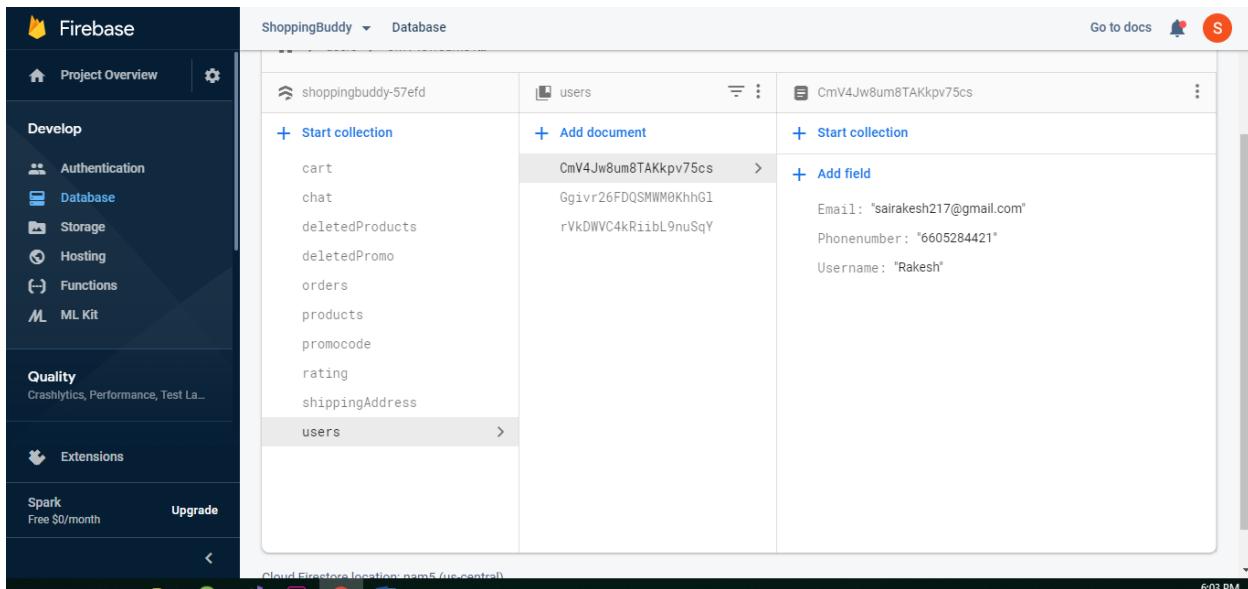
vi. Authentication data will store in the firebase authentication and personal data will store on firebase firestore.



The screenshot shows the Firebase Authentication console for the 'ShoppingBuddy' project. The left sidebar includes 'Project Overview', 'Develop' (with 'Authentication' selected), 'Database', 'Storage', 'Hosting', 'Functions', 'ML Kit', 'Quality', 'Extensions', and 'Spark' (Free \$0/month). The main 'Authentication' tab is active, showing the 'Users' section. A table lists users with columns: Identifier, Providers, Created, Signed In, and User UID. The table contains several rows of user data, such as zqin@nwmissouri.edu, susrithagade1996@gmail.com, s533976@nwmissouri.edu, vennelacheekoti@gmail.com, gonesathwika@gmail.com, vennelacheekoti5@gmail.com, and shoppingbuddyseven@gmail.com.

In firestore, user details will store as follows.

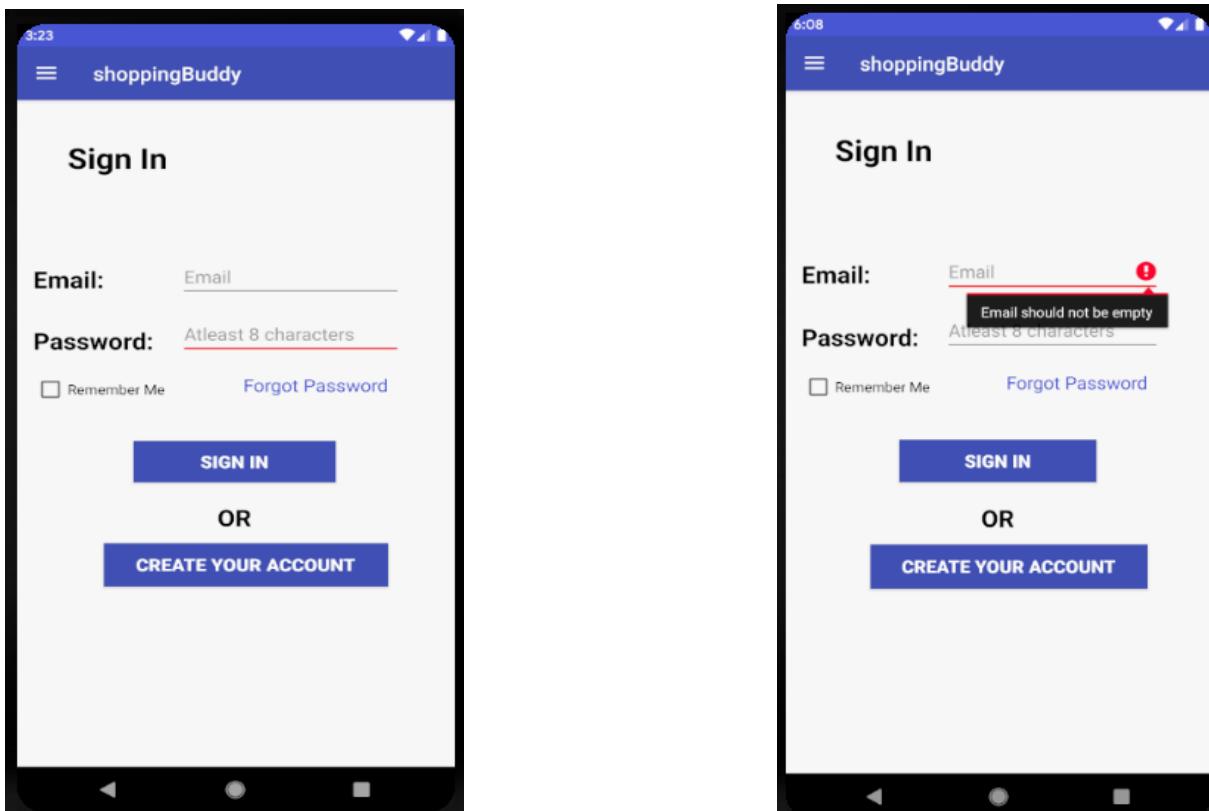
Here document id would be the user's email address.



The screenshot shows the Firestore database console for the 'ShoppingBuddy' project. The left sidebar is identical to the previous screenshot. The main view shows the 'users' collection under the 'shoppingbuddy-57efd' database. A document named 'CmV4Jw8um8TAKkpv75cs' is selected, showing its fields: Email: "sairakesh217@gmail.com", Phonenumber: "6605284421", and Username: "Rakesh". Other documents in the collection include Ggivr26FDQSMWM0KhhG1 and rVkdWVC4kRiibL9nuSqY.

3. User Login

If the user haven't entered username or password, it will display the message in toast.



Login Functionality:

```

if(em.matches( regex: "[a-zA-Z0-9._-]+@[a-z]+.[a-z]+")) {
    if(pwd.length()>=6) {
        firebaseAuth.signInWithEmailAndPassword(em, pwd)
            .addOnCompleteListener( activity: SigninActivity.this, (task) -> {
                if (task.isSuccessful()) {
                    user = firebaseAuth.getCurrentUser();
                    if(saveLoginCheckBox.isChecked()){
                        Sharedpreferences.Editor editor = rememberMe.edit();
                        editor.putString("userName", em);
                        editor.putString("password", pwd);
                        editor.commit();
                    }

                    if (user.isEmailVerified()) {
                        progressDialog.setMessage("Logging in...");
                        progressDialog.show();
                        if(em.equals("shoppingbuddyseven@gmail.com")){
                            Toast.makeText( context: SigninActivity.this, text: "Login sucessful", Toast.LENGTH_SHORT).show();
                            Intent i = new Intent( packageContext: SigninActivity.this, AdminHomeActivity.class);
                            startActivity(i);
                        }else{
                            Toast.makeText( context: SigninActivity.this, text: "Login sucessful", Toast.LENGTH_SHORT).show();
                            Intent i = new Intent( packageContext: SigninActivity.this, Home.class);
                            startActivity(i);
                        }
                    }
                }
            })
    }
}

```

Shopping Buddy

Password Recovery for User:

If the user forgets his/her password, he/she can request for the change of password using forgot password option.

```
private void loginAcc(){
    String em=email.getText().toString().trim();
    if(TextUtils.isEmpty(em)){
        Toast.makeText( context: this, text: "Email should not be empty",Toast.LENGTH_LONG).show();
        return;
    }
    progressDialog.setMessage(" Sending Email ...");
    progressDialog.show();
    firebaseAuth.sendPasswordResetEmail(em).addOnCompleteListener( task) -> {
        if (task.isSuccessful()) {
            Toast.makeText( context: ForgotPassword.this, text: "Email sent sucessful", Toast.LENGTH_SHORT).show();
            Intent i = new Intent( packageContext: ForgotPassword.this, SigninActivity.class);
            startActivity(i);
        } else {
            Toast.makeText( context: ForgotPassword.this, text: "Login failed", Toast.LENGTH_SHORT).show();
            return;
        }
    });
}
```

Home page:



```
promoCollection.orderBy( field: "PromoId", Query.Direction.DESCENDING).get().addOnCompleteListener( task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {
            itemListArray.add(new Container(doc.getString( field: "PromoId"), doc.getString( field: "PromoCode"), doc.getStri
            i++;
        }

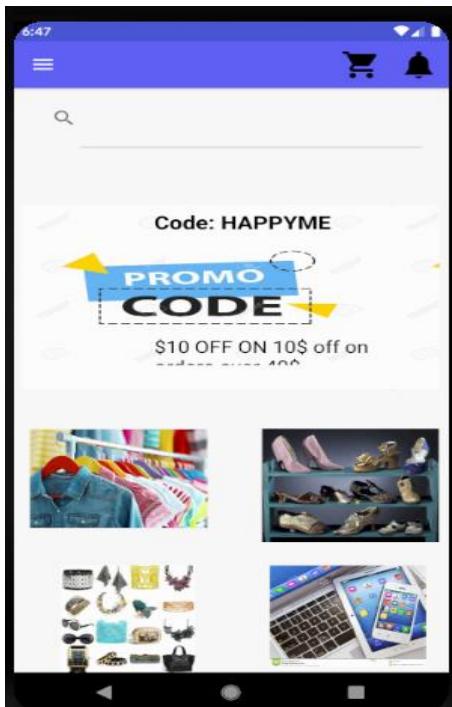
        productLV = findViewById(R.id.productsLV);
        productLV.setHasFixedSize(true);
        productLayoutManager = new LinearLayoutManager( context: Home.this,LinearLayoutManager.HORIZONTAL, reverseLayout: false);
        productsAdapter = new UserPromoAdapter(itemListArray, context: Home.this);
        productLV.setLayoutManager(productLayoutManager);
        productLV.setAdapter(productsAdapter);
    }
};

ImageView in=findViewById(R.id.imageView4);
ImageView inl=findViewById(R.id.imageView5);
ImageView in2=findViewById(R.id.imageView6);
ImageView in3=findViewById(R.id.imageView7);
ImageView in4=findViewById(R.id.imageView8);
ImageView in5=findViewById(R.id.imageView9);
in.setOnClickListener(v) -> {
    Intent ini=new Intent( packageContext: Home.this,ClothingActivity.class);
    startActivity(ini);
};
inl.setOnClickListener(v) -> {
    Intent ini=new Intent( packageContext: Home.this,Footwear.class);
    startActivity(ini);
};
```

Shopping Buddy

Xml Code:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
8     tools:context=".Home"
9     tools:showIn="@layout/app_bar_home">
10
11     <ImageView
12         android:id="@+id/imageView4"
13         android:layout_width="168dp"
14         android:layout_height="120dp"
15         app:layout_constraintBottom_toBottomOf="parent"
16         app:layout_constraintEnd_toEndOf="parent"
17         app:layout_constraintHorizontal_bias="0.065"
18         app:layout_constraintStart_toStartOf="parent"
19         app:layout_constraintTop_toTopOf="parent"
20         app:layout_constraintVertical_bias="0.704"
21         app:srcCompat="@drawable/clothing" />
22
23     <ImageView
24         android:id="@+id/imageView5"
25         android:layout_width="185dp"
26         android:layout_height="135dp"
27         app:layout_constraintBottom_toBottomOf="parent"
28         app:layout_constraintEnd_toEndOf="parent"
29         app:layout_constraintHorizontal_bias="1.0"
30         app:layout_constraintStart_toStartOf="parent"
31         app:layout_constraintTop_toTopOf="parent"
```



Search Functionality:

User can search various products through this search functionality, if he/she doesn't want to go through the categories and want to find the product quickly.

Shopping Buddy

```
productCollection.orderBy( field: "itemId", Query.Direction.DESCENDING).get().addOnCompleteListener(task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {

            if (doc.getString( field: "itemName").toLowerCase().contains(query.toLowerCase()) || doc.getString( field: "category")
                itemListArray.add(new Container(doc.getString( field: "itemId"), doc.getString( field: "itemName"), doc.getDouble(
                    doc.getString( field: "category"), doc.getId(), doc.getString( field: "imageURL"))));
            i++;
        }
    }

    productLV = findViewById(R.id.productsLV);
    productLV.setHasFixedSize(true);
    productLayoutManager = new LinearLayoutManager( context: SearchResultsActivity.this);
    productsAdapter = new SearchAdapter(itemListArray, context: SearchResultsActivity.this);
    productLV.setLayoutManager(productLayoutManager);
    productLV.setAdapter(productsAdapter);
}
});
```

Adapter:

```
public class SearchAdapter extends RecyclerView.Adapter<SearchAdapter.ItemsViewHolder> {

    private ArrayList<Container> itemListArray;

    public SearchAdapter(ArrayList<Container> itemListArray, Context context) {
        this.itemListArray = itemListArray;
        this.context = context;
    }

    private Context context;

    public static class ItemsViewHolder extends RecyclerView.ViewHolder {
        public ImageView productImage;
        public TextView productName;
        public TextView cost;

        public LinearLayout linearLayout2;

        public ItemsViewHolder(@NonNull View itemView) {
            super(itemView);

            productImage = itemView.findViewById(R.id.imageIV);
            productName = itemView.findViewById(R.id.iNameTV);
            cost = itemView.findViewById(R.id.priceTV);

            linearLayout2 = itemView.findViewById(R.id.linearLayout2);
        }
    }
}
```

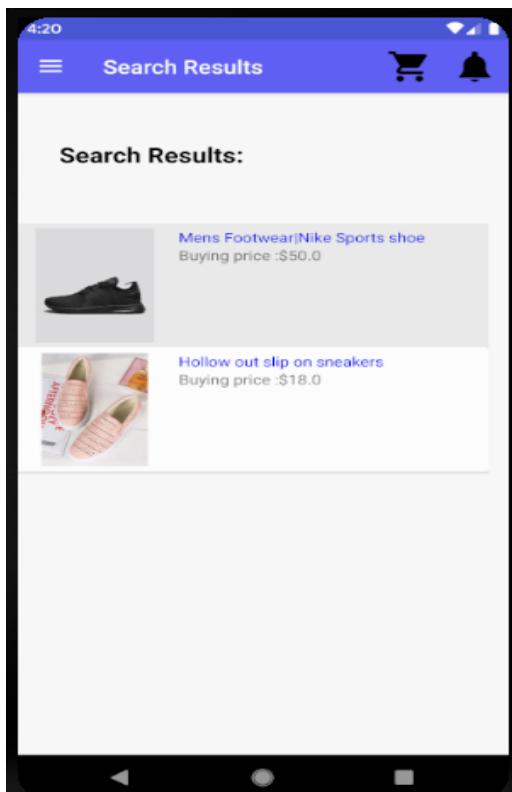
Shopping Buddy

```
public void onBindViewHolder(@NonNull final ItemsViewHolder itemsViewHolder, final int i) {
    final Container currentItem = itemListArray.get(i);

    //itemsViewHolder.itemImage.setImageResource(currentItem.getImage());
    Picasso.get().load(currentItem.getImageURL()).into(itemsViewHolder.productImage);
    itemsViewHolder.productName.setText(currentItem.getItemName());
    itemsViewHolder.cost.setText("Buying price : $" + currentItem.getCost());

    itemsViewHolder.linearLayout2.setOnClickListener((v) -> {
        if(currentItem.getCategory().equals("Men's clothing") || currentItem.getCategory().equals("Women's clothing") || currentItem.getCategory().equals("Men's Accessories") || currentItem.getCategory().equals("Women's Accessories"))
            Intent intent = new Intent(context, Clothing_Item_Details.class);
            intent.putExtra( name: "imageURL", currentItem.getImageURL());
            intent.putExtra( name: "documentId", currentItem.getDocumentId());
            intent.putExtra( name: "description", currentItem.getDescription());
            intent.putExtra( name: "itemId", currentItem.getItemId());
            intent.putExtra( name: "category", currentItem.getCategory());
            intent.putExtra( name: "itemName", currentItem.getItemName());
            intent.putExtra( name: "unitPrice", currentItem.getCost());
            intent.putExtra( name: "size",currentItem.getSize());
            context.startActivity(intent);
    }
    if(currentItem.getCategory().equals("Men's Accessories") || currentItem.getCategory().equals("Women's Accessories"))
        Intent intent = new Intent(context, ItemDetails.class);
        intent.putExtra( name: "imageURL", currentItem.getImageURL());
        intent.putExtra( name: "documentId", currentItem.getDocumentId());
        intent.putExtra( name: "description", currentItem.getDescription());
        intent.putExtra( name: "itemId", currentItem.getItemId());
        intent.putExtra( name: "category", currentItem.getCategory());
        intent.putExtra( name: "itemName", currentItem.getItemName());
        intent.putExtra( name: "unitPrice", currentItem.getCost());
        context.startActivity(intent);
    }
}
```

Phone View



Shopping Buddy

Item list Functionality:

```
Intent i = getIntent();
userEmail = i.getStringExtra("userEmail");

final ArrayList<Container> itemListArray = new ArrayList<>();

productCollection.orderBy( field: "itemId", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {

            if (doc.getString( field: "category").equals("Men's clothing")) {
                itemListArray.add(new Container(doc.getString( field: "itemId"), doc.getString( field: "itemName"), doc.getDouble( field: "price"), doc.getString( field: "category"), doc.getId(), doc.getString( field: "imageURL")));
                i++;
            }
        }

        productLV = findViewById(R.id.productsLV);
        productLV.setHasFixedSize(true);
        productLayoutManager = new LinearLayoutManager( context: ClothingProductList.this);
        productsAdapter = new ClothingProductsAdapter(itemListArray, context: ClothingProductList.this);
        productLV.setLayoutManager(productLayoutManager);
        productLV.setAdapter(productsAdapter);
    }
});
```

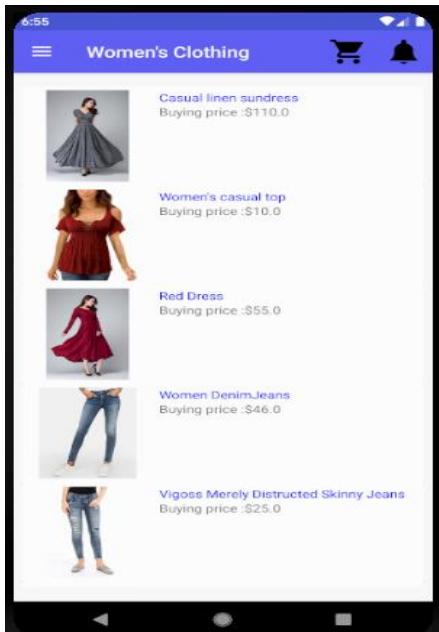
Xml Code:

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".ClothingProductList">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/productsLV"
        android:layout_width="395dp"
        android:layout_height="659dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Shopping Buddy

Phone View:



Products Detail Page:

```
rating1=findViewById(R.id.ratingBar);
    db = FirebaseFirestore.getInstance();
    itemCollection = db.collection( collectionPath: "cart");
ratingCollection=db.collection( collectionPath: "rating");
    ordersCollection=db.collection( collectionPath: "orders");
    productCollection=db.collection( collectionPath: "products");
    quantity=findViewById(R.id.spinner);
    quantityTV=findViewById(R.id.textView32);
    sizeTV=findViewById(R.id.textView31);
    size=findViewById(R.id.size);
    // Get data from the Intent
    Intent i = getIntent();
    docId = i.getStringExtra( name: "documentId");
    imageURL = i.getStringExtra( name: "imageURL");
itemid=i.getStringExtra( name: "itemId");
    //imageIV.setImageResource(i.getIntExtra("image",0));
    Picasso.get().load(imageURL).into(imageIV);
    itemNameTV.setText(i.getStringExtra( name: "itemName"));
    priceTV.setText("$" + i.getDoubleExtra( name: "unitPrice", defaultValue: 0));
    // Get an instance of the items
itemRef = db.collection( collectionPath: "products").document(docId);
    itemRef.get().addOnSuccessListener((OnSuccessListener) (documentSnapshot) → {
        detailsTV.setText(documentSnapshot.get("itemDetails").toString());
    });
    spinner = (Spinner)findViewById(R.id.spinner);
    ArrayAdapter<String> adapter = new ArrayAdapter<>( context: Clothing_Item_Details.this,
        android.R.layout.simple_spinner_item,paths);
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spinner.setAdapter(adapter);
    spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
```

Shopping Buddy

```
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        Log.v( tag: "quantity", (String) parent.getItemAtPosition(position));
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {
    }
}
});
rating1.setOnRatingBarChangeListener((ratingBar, rating, fromUser) -> {
    r=rating;
}
)
ratingCollection.orderBy( field: "user", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc1 : task.getResult()) {
            Log.d( tag: "rate", msg: "item: "+doc1.getString( field: "item")+"itemid: "+itemid);
            if(doc1.getString( field: "user").equals(user.getEmail())) {
                if ( doc1.getString( field: "item").equals(itemid)) {
                    Toast.makeText( context: Clothing_Item_Details.this, text: "you already rated this item", Toast.LENGTH_SHORT).show();
                }
                else{
                    Map<String, Object> addproduct = new HashMap<>();
                    addproduct.put("item", itemid);
                    addproduct.put("rating", r);
                    addproduct.put("user", user.getEmail());
                    ratingCollection.document().set(addproduct);
                    Toast.makeText( context: Clothing_Item_Details.this, text: "rating added to the list", Toast.LENGTH_SHORT).show();
                }
            }
        }
    }
}
)
```

Xml Code:

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".Clothing_Item_Details">

    <ImageView
        android:id="@+id/imageIV"
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_gravity="center_horizontal"
        android:padding="5dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.431"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <TextView
        android:id="@+id/itemNameTV"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="5dp"
        android:text="Product Name:"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
```

Shopping Buddy

```
<TextView  
    android:id="@+id/priceTV"  
    android:layout_width="110dp"  
    android:layout_height="wrap_content"  
    android:padding="5dp"  
    android:textColor="@color/common_google_signin_btn_text_dark_focused"  
    android:textSize="20sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.239"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.433" />  
  
<TextView  
    android:id="@+id/detailHeadTV"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:padding="7dp"  
    android:text="Product Description:"  
    android:textColor="@color/common_google_signin_btn_text_dark_focused"  
    android:textSize="22sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.499" />  
  
<TextView  
    android:id="@+id/detailsTV"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:padding="7dp"  
    android:textColor="@color/common_google_signin_btn_text_dark_focused"  
    android:textSize="20sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="1.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.566" />  
  
<Button  
    android:id="@+id/button16"  
    android:layout_width="130dp"  
    android:layout_height="wrap_content"  
    android:background="@color/colorPrimary"  
    android:text="Add to cart"  
    android:textColor="@color/common_google_signin_btn_text_dark_default"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.95"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.937" />
```

Shopping Buddy

```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="100dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.678" />

<TextView
    android:id="@+id/textView32"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Enter Quantity"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.668" />

<RadioButton
    android:id="@+id/xs"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="XS"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="536dp" />

<RadioButton
    android:id="@+id/s"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="S"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="536dp" />

<RadioButton
    android:id="@+id/m"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="M"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="536dp" />

    app:layout_constraintVertical_bias="0.429" />
<ImageButton
    android:id="@+id/imageButton"
    android:layout_width="85dp"
    android:layout_height="55dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.819"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.109"
    app:srcCompat="@drawable/ic_menu_share" />

<RatingBar
    android:id="@+id/ratingBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.093"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.923" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Shopping Buddy

```
public ClothingProductsAdapter.ItemsViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
    View v = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.activity_container, viewGroup, attachToRoot: false);
    ClothingProductsAdapter.ItemsViewHolder itemsVH = new ClothingProductsAdapter.ItemsViewHolder(v);
    return itemsVH;
}

@Override
public void onBindViewHolder(@NonNull final ClothingProductsAdapter.ItemsViewHolder itemsViewHolder, final int i) {
    final Container currentItem = itemListArray.get(i);

    //itemsViewHolder.itemImage.setImageResource(currentItem.getImage());
    Picasso.get().load(currentItem.getImageURL()).into(itemsViewHolder.productImage);
    itemsViewHolder.productName.setText(currentItem.getItemName());
    itemsViewHolder.cost.setText("Buying price : $" + currentItem.getCost());

    itemsViewHolder.linearLayout2.setOnClickListener((v) -> {
        Intent intent = new Intent(context, Clothing_Item_Details.class);
        intent.putExtra( name: "imageURL", currentItem.getImageURL());
        intent.putExtra( name: "documentId", currentItem.getDocumentId());
        intent.putExtra( name: "description", currentItem.getDescription());
        intent.putExtra( name: "itemId", currentItem.getItemId());
        intent.putExtra( name: "category", currentItem.getCategory());
        intent.putExtra( name: "itemName", currentItem.getItemName());
        intent.putExtra( name: "unitPrice", currentItem.getCost());
        context.startActivity(intent);
    });
}

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

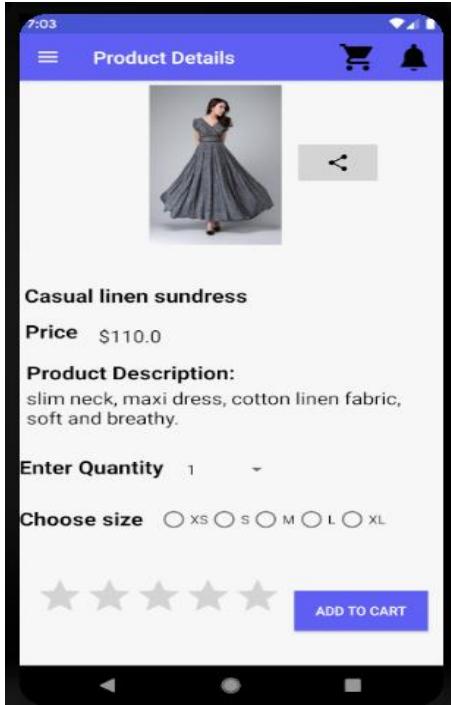
```
package com.example.shoppingbuddy;

import ...;

public class Container{
    private String itemID;
    private String itemName;
    private double cost;
    private String description;
    private String category;
    private String documentId;
    private String imageURL;
    private Long quantity;
    private String size;
    private String email;
    private String promoid;
    private String promocode;
    private String price;
    private String pdes;
    private String message, sender, reciever;

    public Container(String itemID, String itemName, double cost, String description, String category, String documentId, String imageURL) {
        this.itemID = itemID;
        this.itemName = itemName;
        this.cost = cost;
        this.description = description;
        this.category = category;
        this.documentId = documentId;
        this.imageURL = imageURL;
    }
}
```

Shopping Buddy



Cart Functionality:

Whenever user adds the products to the cart they are stored and he/she can buy the products by clicking on checkout and can also delete the products from cart if he/she doesn't like the product anymore. All the products are added to the firebase database

```
cartCollection.orderBy( field: "itemId", Query.Direction.DESCENDING).get().addOnCompleteListener(task) -> {
    if (task.isSuccessful()) {
        for (final QueryDocumentSnapshot doc : task.getResult()) {
            if(user.getEmail().equals(doc.getString( field: "user"))){
                Log.d( tag: "click", msg: "inside users verification");
                productCollection.orderBy( field: "itemId",Query.Direction.DESCENDING).get().addOnCompleteListener(task) -> {
                    int i=0;
                    for(QueryDocumentSnapshot docl: task.getResult()){
                        Log.d( tag: "click", msg: "inside products collection");
                        if(doc.getString( field: "itemId").equals(docl.getId())){
                            Log.d( tag: "click", msg: "inside products verification");
                            itemListArray.add(new Container(docl.getString( field: "itemId"), docl.getString( field: "itemName"), docl.getString( field: "category"), docl.getId(), docl.getString( field: "imageURL"),doc.getLong( field: "cost")));
                            Log.d( tag: "click", msg: "cost: "+docl.getDouble( field: "cost"));
                            totalcost=totalcost+(docl.getDouble( field: "cost")*doc.getLong( field: "quantity"));
                            i++;
                        }
                    }
                    productLV = findViewById(R.id.productsLV);
                    productLV.setHasFixedSize(true);
                    productLayoutManager = new LinearLayoutManager( context: CartActivity.this);
                    productsAdapter = new CartAdapter(itemListArray, context: CartActivity.this);
                    productLV.setLayoutManager(productLayoutManager);
                    productLV.setAdapter(productsAdapter);
                    TextView totaltv=findViewById(R.id.textView41);
                    totaltv.setText("Total cost: "+totalcost);
                });
            }
        }
    }
}
```

Shopping Buddy

```
public CartAdapter.ItemsViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
    View v = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.activity_cart_container, viewGroup, false);
    CartAdapter.ItemsViewHolder itemsVH = new CartAdapter.ItemsViewHolder(v);
    return itemsVH;
}

@Override
public void onBindViewHolder(@NonNull final CartAdapter.ItemsViewHolder itemsViewHolder, final int i) {
    final Container currentItem = itemListArray.get(i);

    //itemsViewHolder.itemImage.setImageResource(currentItem.getImage());
    Picasso.get().load(currentItem.getImageURL()).into(itemsViewHolder.productImage);
    itemsViewHolder.productName.setText("Product Name: " + currentItem.getItemName());
    itemsViewHolder.cost.setText("Price: $" + currentItem.getCost());
    itemsViewHolder.quantity.setText("Quantity: " + currentItem.getQuantity());
    itemsViewHolder.size.setText("Size: " + currentItem.getSize());

    itemsViewHolder.delete.setOnClickListener((v) -> {
        CartActivity ct=new CartActivity();
        ct.deleteprod(currentItem.getDocumentId());
        Intent in=new Intent(context,CartActivity.class);
        context.startActivity(in);
    });
}
}
```

CartAdapter > ItemsViewHolder

Xml code:

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".CartActivity">

    <TextView
        android:id="@+id/textView42"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Promo Code(if any)"
        android:textColor="#0F0FOF"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.108"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.155" />
```

Shopping Buddy

The screenshot shows the Android Studio interface with the XML layout file for the Shopping Buddy application. The code is organized into three main sections:

```
<Button
    android:id="@+id/button18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:text="Checkout"
    android:textColor="@color/common_google_signin_btn_text_dark_default"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.904"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.041" />

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="412dp"
    android:layout_height="509dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.903" />
```

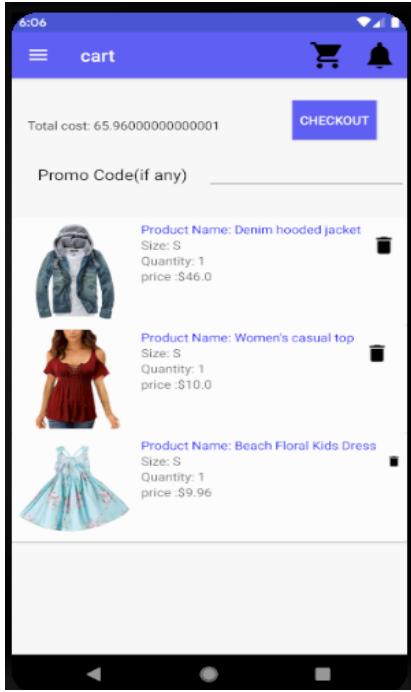


```
<TextView
    android:id="@+id/textView41"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Total Cost: 0"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.081"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.07" />

<EditText
    android:id="@+id/editText18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.139" />
```

The right side of the screen shows the "Preview" tab, which displays a visual representation of the layout with a button labeled "Checkout", a recycler view, a text view showing "Total Cost: 0", and an edit text field.

Shopping Buddy



Shipping Address:

User need to enter the address to which he wants his order to be delivered and shipping address is added to the firebase database.

```
shippingCollection.orderBy( field: "orderId", Query.Direction.DESCENDING) .get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        for (final QueryDocumentSnapshot doc : task.getResult()) {

            if (doc.getString( field: "orderId").equals("")) {
                orderId = lname + Math.random();
            } else if (doc.getString( field: "orderId").equals(orderId)) {
                orderId = lname + Math.random();
            } else {
                orderId = lname + Math.random();
            }
        }
    });
}

if(fname.equals("") || lname.equals("") || address.equals("") || city.equals("") || state.equals("") || zipcode.equals("")){
    Toast.makeText( context: ShippingAddressActivity.this, text: "All the field needs to be filled ", Toast.LENGTH_SHORT).
}
else {
    Map<String, Object> addproduct = new HashMap<>();
    addproduct.put("User", user);
    addproduct.put("OrderId", orderId );
    addproduct.put("First Name", fname);
    addproduct.put("Last Name", lname);
    addproduct.put("Address", address);
    addproduct.put("City", city);
    addproduct.put("State", state);
    addproduct.put("Zipcode", zipcode);
}
```

Shopping Buddy

Xml code

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"  
    tools:context=".ShippingAddressActivity">  
  
    <TextView  
        android:id="@+id/textView47"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Total cost"  
        android:textColor="#0F0FOF"  
        android:textSize="18sp"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.123"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.792" />  
  
    <TextView  
        android:id="@+id/textView45"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Zip Code"  
        android:textColor="#0F0FOF"  
        android:textSize="18sp"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.105"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent"  
        app:layout_constraintVertical_bias="0.589" />  
  
    <TextView  
        android:id="@+id/textView44"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="State"  
        android:textColor="#0F0FOF"  
        android:textSize="18sp"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintEnd_toEndOf="parent"  
        app:layout_constraintHorizontal_bias="0.593"  
        app:layout_constraintStart_toStartOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
/>
```

Phone view:



Shopping Buddy

Payment:

After entering the shipping details then user can do his/her payment. Payment details will be stored in the firebase.



```
FirebaseAuth auth = FirebaseAuth.getInstance();
user = auth.getCurrentUser();
//Intent i=getIntent();
orderid=in.getStringExtra( name: "orderid");
cartCollection.orderBy( field: "itemId", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        for (final QueryDocumentSnapshot doc : task.getResult()) {
            if(user.getEmail().equals(doc.getString( field: "user"))){
                Log.d( tag: "click", msg: "inside users verification");
                Map<String, Object> addproduct = new HashMap<>();
                addproduct.put("OrderId",orderid);
                addproduct.put("user",user.getEmail());
                addproduct.put("productId",doc.getString( field: "itemId"));
                addproduct.put("quantity",doc.getLong( field: "quantity"));
                addproduct.put("size",doc.getString( field: "size"));
                addproduct.put("total cost",totcost);
                addproduct.put("item",doc.getString( field: "item"));
                orderCollection.document().set(addproduct);
                Toast.makeText( context: PaymentActivity.this, text: "check your orders", Toast.LENGTH_SHORT).show();
                //finish();
                itemDoc = db.collection( collectionPath: "cart").document(doc.getId());
                itemDoc.delete();
                Log.d( tag: "itemclick", msg: ""+doc.getId());
                Intent in=new Intent( packageContext: PaymentActivity.this,Home.class);
                startActivity(in);
            }
        }
    }
});
```

Xml code



```
<EditText
    android:id="@+id/editText22"
    android:layout_width="78dp"
    android:layout_height="56dp"
    android:ems="10"
    android:hint="MM/YY"
    android:inputType="textPersonName"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.912"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.475" />

<TextView
    android:id="@+id/textView53"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:text="Expiry Date:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.609"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499" />
```

Shopping Buddy

The image shows two screenshots of the Android Studio interface, each displaying an XML layout file for a mobile application named "Shopping Buddy".

Top Screenshot: This screenshot shows the XML code for a fragment or activity layout. It includes an `<EditText>` and a `<TextView>`. The `<EditText>` has an ID of `@+id/editText21`, a width of `78dp`, and a height of `56dp`. It features a font size of `10`, a text color of `@color/common_google_signin_btn_text_dark_focused`, and a text size of `20sp`. The `<TextView>` has an ID of `@+id/textView52`, a wrap content width, and a height of `40dp`. Its text is `"CVV"`, with a bold text style, a font color of `@color/common_google_signin_btn_text_dark_focused`, and a text size of `20sp`.

Bottom Screenshot: This screenshot shows the XML code for another part of the application. It includes two `<TextView>` elements. The first `<TextView>` has an ID of `@+id/textView50`, a wrap content width, and a height of `40dp`. Its text is `"Card Number:"`, with a bold text style, a font color of `@color/common_google_signin_btn_text_dark_focused`, and a text size of `20sp`. The second `<TextView>` has an ID of `@+id/textView48`, a wrap content width, and a height of `40dp`. Its text is `"Total Amount:"`, with a bold text style, a font color of `@color/common_google_signin_btn_text_dark_focused`, and a text size of `20sp`.

Both screenshots include standard Android Studio toolbars on the right side, including "Gradle", "Preview", and "Device File".

```
<EditText
    android:id="@+id/editText21"
    android:layout_width="78dp"
    android:layout_height="56dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.261"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.475" />

<TextView
    android:id="@+id/textView52"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:text="CVV"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.056"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.499" />
```



```
<TextView
    android:id="@+id/textView50"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:text="Card Number:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.056"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.386" />

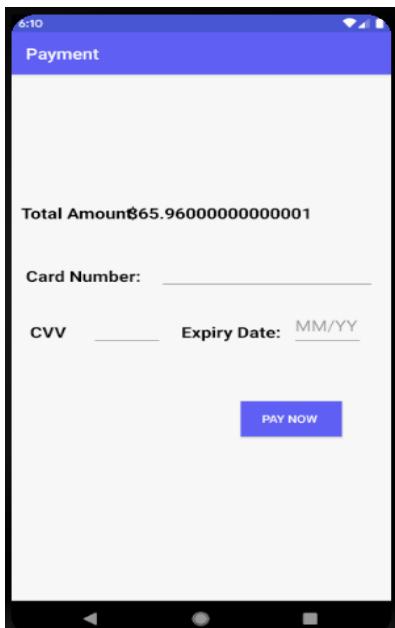
<TextView
    android:id="@+id/textView48"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:text="Total Amount:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.039"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.259" />
```

Shopping Buddy

```
        <EditText
            android:id="@+id/editText20"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName"
            android:textColor="@color/common_google_signin_btn_text_dark_focused"
            android:textSize="20sp"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.906"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.367" />

        <Button
            android:id="@+id/button19"
            android:layout_width="110dp"
            android:layout_height="47dp"
            android:background="@color/colorPrimary"
            android:text="pay now"
            android:textColor="@color/common_google_signin_btn_text_dark_default"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.826"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintVertical_bias="0.666" />
```

Phone view:



Orders:

After the payment is done then user can look at the orders he/she had made and can cancel or return products.

Shopping Buddy

```
ordersCollection.orderBy( field: "productId", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        for (final QueryDocumentSnapshot doc : task.getResult()) {
            if(user.getEmail().equals(doc.getString( field: "user"))){
                Log.d( tag: "click", msg: "inside users verification");
                productCollection.orderBy( field: "itemId",Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
                    int i=0;
                    for(QueryDocumentSnapshot doc1: task.getResult()){
                        Log.d( tag: "click", msg: "inside products collection");
                        if(doc.getString( field: "productId").equals(doc1.getId())){
                            Log.d( tag: "click", msg: "inside products verification");
                            itemListArray.add(new Container(doc1.getString( field: "itemId"), doc1.getString( field: "itemName"), doc1.getDocId(), doc1.getString( field: "category"), doc1.getId(), doc1.getString( field: "imageURL"),doc1.getLong( field: "quantity")));
                            i++;
                        }
                    }
                    productLV = findViewById(R.id.productsLV);
                    productLV.setHasFixedSize(true);
                    productLayoutManager = new LinearLayoutManager( context: Orders_Product_List.this);
                    productsAdapter = new OrdersAdapter(itemListArray, context: Orders_Product_List.this);
                    productLV.setLayoutManager(productLayoutManager);
                    productLV.setAdapter(productsAdapter);
                });
            }
        }
    });
}

private ArrayList<Container> itemListArray;

public OrdersAdapter(ArrayList<Container> itemListArray, Context context) {
    this.itemListArray = itemListArray;
    this.context = context;
}

private Context context;

public static class ItemsViewHolder extends RecyclerView.ViewHolder {
    public ImageView productImage;
    public TextView productName;
    public TextView cost;
    public Button cancel,return;

    public LinearLayout linearLayout2;

    public ItemsViewHolder(@NonNull View itemView) {
        super(itemView);

        productImage = itemView.findViewById(R.id.imageIV);
        productName = itemView.findViewById(R.id.iNameTV);
        cost = itemView.findViewById(R.id.priceTV);
        cancel=itemView.findViewById(R.id.button24);
        return=itemView.findViewById(R.id.button25);

        linearLayout2 = itemView.findViewById(R.id.linearlayout2);
    }
}
```

Shopping Buddy

Xml code:



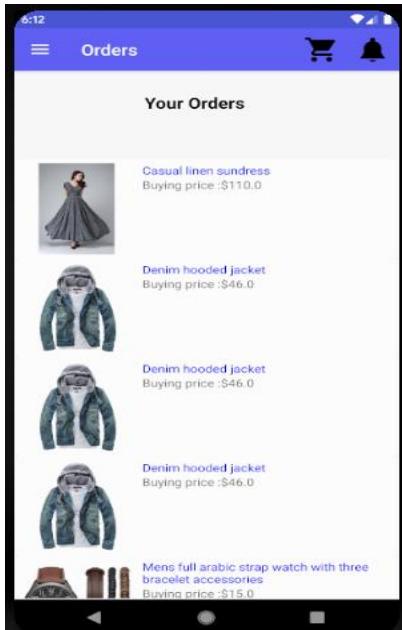
```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="407dp"
    android:layout_height="577dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.989">

</androidx.recyclerview.widget.RecyclerView>

<TextView
    android:id="@+id/textView11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Your Orders"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="20sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.46"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.043" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Phone view



User Settings:

User can change his privacy settings like changing password and can also contact us.

Shopping Buddy

Privacy:

```
public void changePassword() {
    et1=findViewById(R.id.editText24);
    et2=findViewById(R.id.editText25);
    String newpassword = et1.getText().toString();
    String confirmpass = et2.getText().toString();
    Log.d( tag: "check",newpassword);
    Log.d( tag: "check",confirmpass);
    /////
    ////
    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    user.updatePassword(newpassword).addOnCompleteListener((task) -> {
        if (task.isSuccessful()) {
            Toast.makeText( context: ChangePassword.this, text: "Password updated", Toast.LENGTH_SHORT).show();
            Log.d( tag: "success", msg: "password updated");
            Intent i = new Intent( packageContext: ChangePassword.this, SigninActivity.class);
            startActivity(i);
        } else {
            Toast.makeText( context: ChangePassword.this, text: "Password updated", Toast.LENGTH_SHORT).show();
        }
    });
//}
}
```

Xml code:



The screenshot shows the Android Studio XML layout editor for a profile screen. The layout consists of two main TextView elements. The first TextView has the following attributes:

```
<TextView
    android:id="@+id/textView13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Your Profile"
    android:textSize="26sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.025" />
```

The second TextView has the following attributes:

```
<TextView
    android:id="@+id/textView14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Email"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.044"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.134" />
```

On the right side of the interface, there are tabs for 'Gradle' and 'Preview'.

Shopping Buddy

The screenshot shows the Android Studio XML layout editor with the code for a password update screen. The code uses ConstraintLayout with various constraints and dimensions. The interface includes an EditText for the new password, a TextView for the change password instruction, another TextView for the confirm password instruction, and a Button labeled "UPDATE". The code is color-coded for syntax highlighting.

```
<EditText  
    android:id="@+id/editText23"  
    android:layout_width="259dp"  
    android:layout_height="48dp"  
    android:ems="10"  
    android:inputType="textPersonName"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="1.0"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.105" />  
  
<TextView  
    android:id="@+id/textView15"  
    android:layout_width="wrap_content"  
    android:layout_height="60dp"  
    android:text="Change Your Password"  
    android:textSize="20sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.079"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.209" />  
  
- - -  
  
<TextView  
    android:id="@+id/textView17"  
    android:layout_width="128dp"  
    android:layout_height="63dp"  
    android:text="Confirm Password"  
    android:textSize="20sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.056"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.366" />  
  
<Button  
    android:id="@+id/button22"  
    android:layout_width="118dp"  
    android:layout_height="63dp"  
    android:text="UPDATE"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />  
/>
```

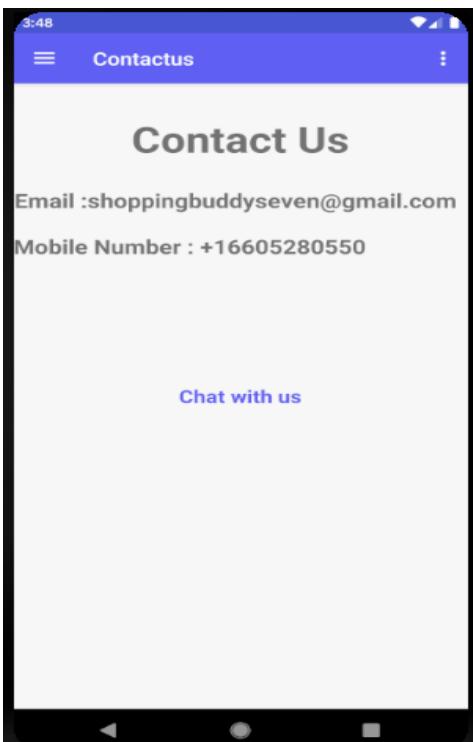
Shopping Buddy

Phone view



Contact us:

User can contact admin either by mailing or he/she can directly chat with admin.



Shopping Buddy

Chat:

```
sendmsg.setOnClickListener(v -> {
    msg=message.getText().toString();
    Map<String, Object> messages = new HashMap<>();
    messages.put("From", email);
    messages.put("To", "shoppingbuddyseven@gmail.com");
    messages.put("Message", msg);
    messages.put("Date",new Timestamp(new Date()) );

    chatCollection.document().set(messages);
    Toast.makeText( context: UserChatActivity.this, text: "Message sent", Toast.LENGTH_SHORT).show();
    message.setText("");

    final ArrayList<Container> itemListArray = new ArrayList<>();
    chatCollection.orderBy( field: "Date", Query.Direction.ASCENDING).get().addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            int i = 0;
            for (QueryDocumentSnapshot doc : task.getResult()) {

                if (doc.getString( field: "From").equals(email) || doc.getString( field: "To").equals(email)) {
                    // if(doc.getDate("Date").after(d.toDate())){
                    itemListArray.add(new Container(doc.getId(),doc.getString( field: "Message"), doc.getString( field: "From")));
                    i++;
                    // }
                }
            }
        }

        productLV = findViewById(R.id.recyclerview);
        productLV.setHasFixedSize(true);
        productLayoutManager = new LinearLayoutManager( context: UserChatActivity.this);
        productsAdapter = new ChatAdapter(itemListArray, context: UserChatActivity.this);
        productLV.setLayoutManager(productLayoutManager);
    })
}

private ArrayList<Container> itemListArray;
public static final int MSG_TYPE_LEFT=0;
public static final int MSG_TYPE_RIGHT=1;

FirebaseUser user;
public String email;
public ChatAdapter(ArrayList<Container> itemListArray, Context context) {
    this.itemListArray = itemListArray;
    this.context = context;
}

public ChatAdapter() {
}

private Context context;

public static class ItemsViewHolder extends RecyclerView.ViewHolder {

    public TextView message;
    public LinearLayout linearLayout2;

    public ItemsViewHolder(@NonNull View itemView) {
        super(itemView);

        message = itemView.findViewById(R.id.showmsg);
        linearLayout2 = itemView.findViewById(R.id.linearLayout2);
    }
}
```

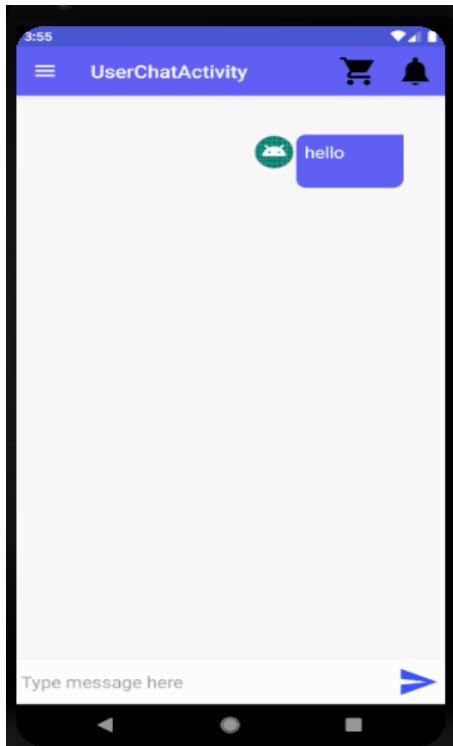
Shopping Buddy

Xml code

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerview"
    android:layout_width="match_parent"
    android:layout_height="566dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="5dp"
    android:background="#fff"
    android:layout_gravity="bottom">
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/message"
        android:layout_toLeftOf="@+id/send"
        android:layout_centerVertical="true"
        android:background="@android:color/transparent"
        android:hint="Type message here"/>
    <ImageButton
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:id="@+id/send"
        android:layout_alignParentEnd="true"
        android:background="@+id/icon_send" />
</RelativeLayout>
```

Phone View:



Shopping Buddy- Android App

Version 2.0.0

Technical Manual-Admin

Team : Android Team

Course: 44692-04

Class : GDP 02 Spring 2019

TEAM MEMBERS

Sathwika Gone

Vennela Cheekoti

Rakesh Avirineni

Yaswanth Yarram

Mehar Choudhry Meenavilli

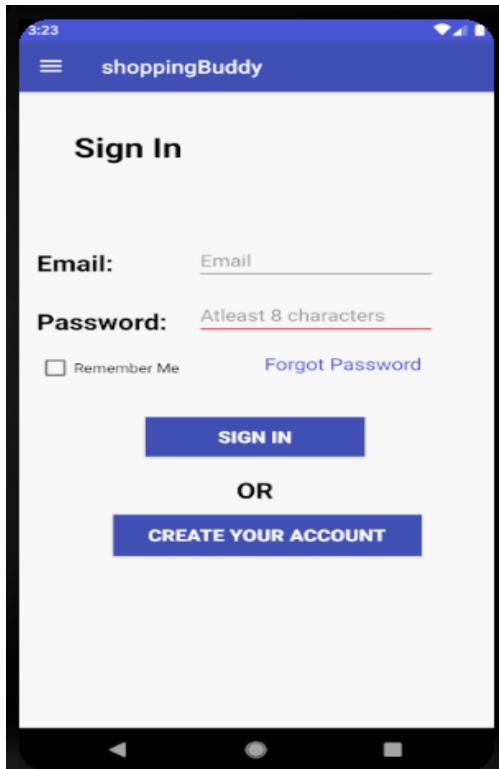
Sandeep Devineni

Susritha Gade

Admin Sign In functionality:

Shopping Buddy

Admin can sign in into the application with the email that he/she has been assigned and by entering password.



```
if(em.matches( regex: "[a-zA-Z0-9._-]+@[a-z].[a-z]+")) {
    if(pwd.length()>=6) {
        firebaseAuth.signInWithEmailAndPassword(em, pwd)
            .addOnCompleteListener( activity: SigninActivity.this, (task) -> {
                if (task.isSuccessful()) {
                    user = firebaseAuth.getCurrentUser();
                    if(saveLoginCheckBox.isChecked()){
                        Sharedpreferences.Editor editor = rememberMe.edit();
                        editor.putString("userName", em);
                        editor.putString("password", pwd);
                        editor.commit();
                    }

                    if (user.isEmailVerified()) {
                        progressDialog.setMessage("Logging in...");
                        progressDialog.show();
                        if(em.equals("shoppingbuddyseven@gmail.com")){
                            Toast.makeText( context: SigninActivity.this, text: "Login sucessful", Toast.LENGTH_SHORT).show();
                            Intent i = new Intent( packageContext: SigninActivity.this, AdminHomeActivity.class);
                            startActivity(i);
                        }else{
                            Toast.makeText( context: SigninActivity.this, text: "Login sucessful", Toast.LENGTH_SHORT).show();
                            Intent i = new Intent( packageContext: SigninActivity.this, Home.class);
                            startActivity(i);
                        }
                    }
                }
            })
    }
}
```

Admin Home:

Shopping Buddy

```
        Button addProduct=findViewById(R.id.button5);
        addProduct.setOnClickListener((v) -> {
            Intent i = new Intent( packageContext: AdminHomeActivity.this, AddProductActivity.class);
            startActivity(i);
        });
        Button editProduct=findViewById(R.id.button9);
        editProduct.setOnClickListener((v) -> {
            Intent i = new Intent( packageContext: AdminHomeActivity.this, SelectProduct.class);
            startActivity(i);
        });
        Button addPromo=findViewById(R.id.button10);
        addPromo.setOnClickListener((v) -> {
            Intent i = new Intent( packageContext: AdminHomeActivity.this, AdminPromoActivity.class);
            startActivity(i);
        });
        Button editPromo=findViewById(R.id.button11);
        editPromo.setOnClickListener((v) -> {
            Intent i = new Intent( packageContext: AdminHomeActivity.this, SelectPromocodeActivity.class);
            startActivity(i);
        });
    }
}
```

Xml code



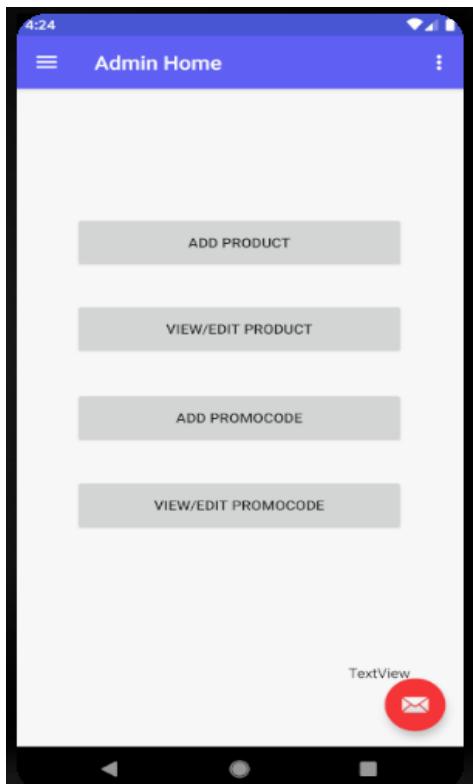
Shopping Buddy



The screenshot shows the Android Studio XML layout editor. On the left is the XML code for a ConstraintLayout containing two buttons. On the right is a preview window showing a mobile phone screen with a blue header bar labeled "Admin Home". Below the header are four grey rectangular buttons with white text: "ADD PRODUCT", "VIEW/EDIT PRODUCT", "ADD PROMOCODE", and "VIEW/EDIT PROMOCODE". At the bottom center is a red circular button with a white envelope icon. The preview window has a title bar at the top with "Preview" and a toolbar on the right.

```
<ConstraintLayout>
    ...
    <Button
        android:id="@+id/button10"
        android:layout_width="303dp"
        android:layout_height="57dp"
        android:text="Add PromoCode"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/button11"
        android:layout_width="303dp"
        android:layout_height="57dp"
        android:text="View/Edit Promocode"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.645" />
</ConstraintLayout>
```

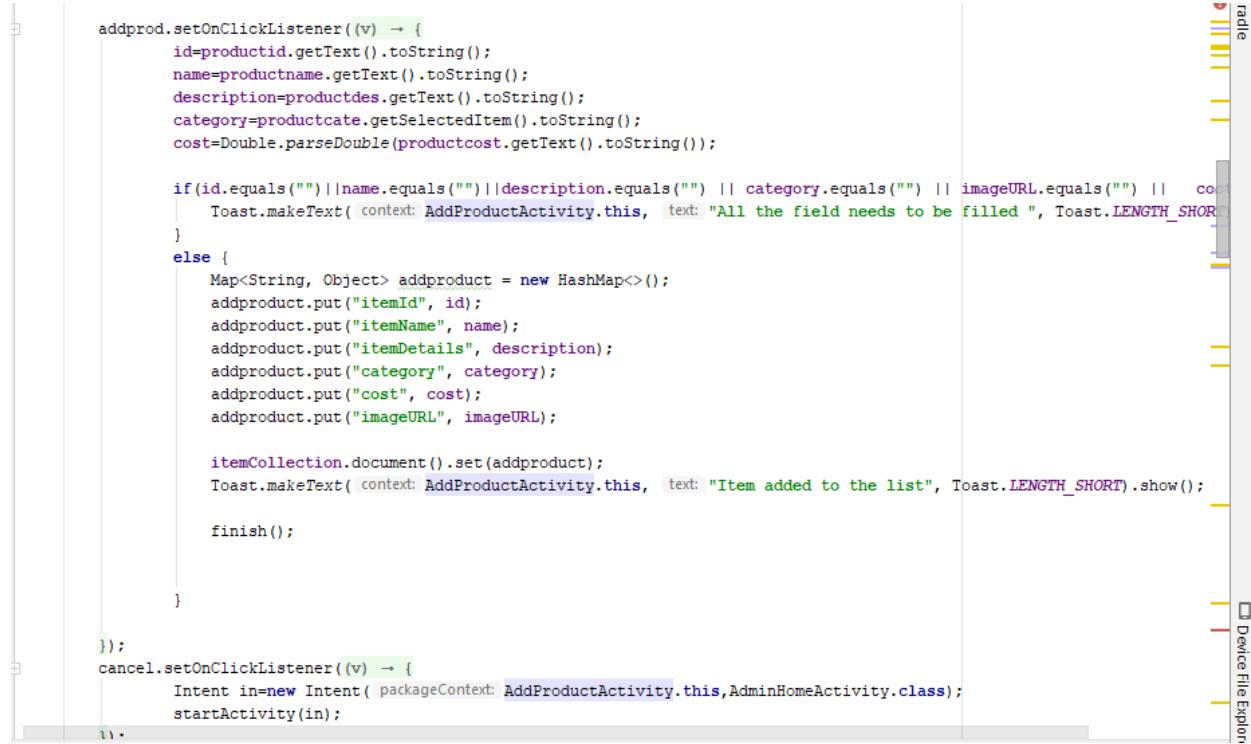
Phone view



Shopping Buddy

Add Product:

Admin has to add the products so that the user can buy products and all these products are stored in the firebase database.



```
addprod.setOnClickListener(v) -> {
    id=productid.getText().toString();
    name=productname.getText().toString();
    description=productdes.getText().toString();
    category=productcate.getSelectedItem().toString();
    cost=Double.parseDouble(productcost.getText().toString());

    if(id.equals("")||name.equals("")||description.equals("") || category.equals("") || imageURL.equals("") || cost.equals("")){
        Toast.makeText(context: AddProductActivity.this, text: "All the field needs to be filled ", Toast.LENGTH_SHORT).show()
    }
    else {
        Map<String, Object> addproduct = new HashMap<>();
        addproduct.put("itemId", id);
        addproduct.put("itemName", name);
        addproduct.put("itemDetails", description);
        addproduct.put("category", category);
        addproduct.put("cost", cost);
        addproduct.put("imageURL", imageURL);

        itemCollection.document().set(addproduct);
        Toast.makeText(context: AddProductActivity.this, text: "Item added to the list", Toast.LENGTH_SHORT).show();

        finish();
    }
};

cancel.setOnClickListener(v) -> {
    Intent in=new Intent(packageContext: AddProductActivity.this, AdminHomeActivity.class);
    startActivity(in);
}
```

Xml Code:



```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".AddProductActivity">

    <Button
        android:id="@+id/button12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:text="ADD"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.95"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.974" />

    <TextView
        android:id="@+id/textView25"
        android:layout_width="150dp"
        android:layout_height="40dp"
        android:text="Add Product"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
```

Shopping Buddy

The screenshot shows the Android Studio interface with the XML code for a layout file. The code defines a vertical stack of views: two TextViews at the top, followed by a Spinner, and finally a Button at the bottom. Each view has specific attributes for its appearance and position within the layout.

```
<TextView  
    android:id="@+id/textView26"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Product ID"  
    android:textColor="@color/common_google_signin_btn_text_dark_focused"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.018"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.208" />  
  
<TextView  
    android:id="@+id/textView27"  
    android:layout_width="wrap_content"  
    android:layout_height="20dp"  
    android:text="Product Name"  
    android:textColor="@color/common_google_signin_btn_text_dark_focused"  
    android:textSize="18sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.023"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.328" />  
  
<Spinner  
    android:id="@+id/spinner"  
    android:layout_width="200dp"  
    android:layout_height="50dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.924"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.598" />  
  
<Button  
    android:id="@+id/button13"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Upload Image"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.066"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.822" />
```

Shopping Buddy



```
<Spinner
    android:id="@+id/spinner"
    android:layout_width="200dp"
    android:layout_height="50dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.924"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.598" />

<Button
    android:id="@+id/button13"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Upload Image"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.066"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.822" />

<ImageView
    android:id="@+id/imageView3"
    android:layout_width="123dp"
    android:layout_height="111dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.597"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.898"
    app:srcCompat="@color/color" />

<Button
    android:id="@+id/button14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary"
    android:text="Cancel"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.068"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.974" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Phone View



View/Edit Product:

Admin can Search for the product that he/she wants to edit or view the product after the products get displayed admin can edit the product. Admin can also remove product.

Shopping Buddy

```
saveChangesBTN.setOnClickListener((v) -> {
    prododes = productdes.getText().toString();
    prodcost = Double.parseDouble(productcost.getText().toString());
    AlertDialog.Builder builder = new AlertDialog.Builder(context: EditorRemoveProduct.this);
    builder.setMessage("Are you sure you want to save the changes").setPositiveButton(text: "Yes", (dialog, which) ->
        itemDoc.update(field: "itemDetails", prododes, ...moreFieldsAndValues: "cost", prodcost);
        Toast.makeText(context: EditorRemoveProduct.this, text: "Product updated successfully", Toast.LENGTH_LONG);
        Intent j = new Intent(packageContext: EditorRemoveProduct.this, AdminHomeActivity.class);
        j.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(j);
        EditorRemoveProduct.this.finish();
    ).setNegativeButton(text: "No", (dialog, which) -> {
        finish();
    }).show();
});

cancelBTN.setOnClickListener((v) -> {
    Intent i = new Intent(packageContext: EditorRemoveProduct.this, SelectProduct.class);
    startActivity(i);
});
removeBTN.setOnClickListener((v) -> {
    AlertDialog.Builder builder = new AlertDialog.Builder(context: EditorRemoveProduct.this);
    builder.setMessage("Are you sure you want to delete the product").setPositiveButton(text: "Yes", (dialog, which) ->
        Map<String, Object> deleteproduct = new HashMap<>();
        deleteproduct.put("itemId", prodid);
        deleteproduct.put("itemName", name1);
        deleteproduct.put("cost", prodcost);
        deleteproduct.put("imageURL", imageURL);
        itemCollection.document().set(deleteproduct);
        itemDoc.delete();
        finish();
    );
});
```

Xml code

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".EditorRemoveProduct">

    <Button
        android:id="@+id/button12"
        android:layout_width="125dp"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:text="Save Changes"
        android:textColor="@color/common_google_signin_btn_text_dark_default"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.95"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.974" />

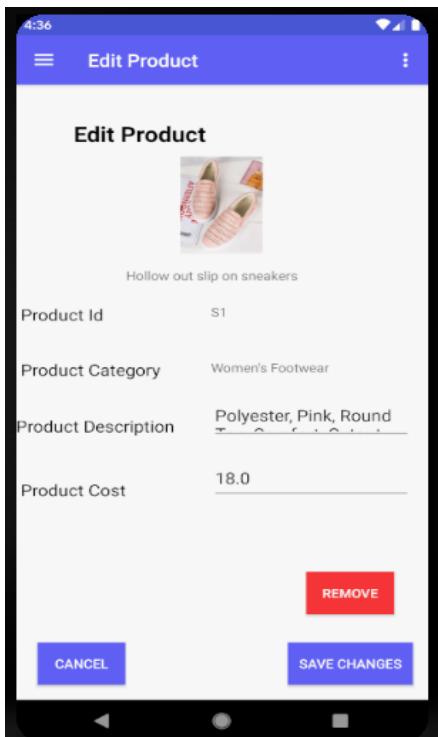
    <TextView
        android:id="@+id/textView25"
        android:layout_width="150dp"
        android:layout_height="40dp"
        android:text="Edit Product"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="24sp"
        android:textStyle="bold" />
```

Shopping Buddy

```
<EditText  
    android:id="@+id/editText13"  
    android:layout_width="200dp"  
    android:layout_height="50dp"  
    android:ems="10"  
    android:inputType="textPersonName"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.924"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.651" />  
  
<ImageView  
    android:id="@+id/imageView3"  
    android:layout_width="123dp"  
    android:layout_height="111dp"  
    android:layout_constraintBottom_toBottomOf="parent"  
    android:layout_constraintEnd_toEndOf="parent"  
    android:layout_constraintStart_toStartOf="parent"  
    android:layout_constraintTop_toTopOf="parent"  
    android:layout_constraintVertical_bias="0.138"  
    app:srcCompat="@color/color" />  
  
<Button  
    android:id="@+id/button14"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@color/colorPrimary"  
    android:text="Cancel"  
    android:textColor="@color/common_google_signin_btn_text_dark_default"  
    android:drawable="..."/> android:drawable="..."/> android:background="..."/> android:background="..."/> android:background="..."/> android:background="..."/>  
<TextView  
    android:id="@+id/textView39"  
    android:layout_width="200dp"  
    android:layout_height="32dp"  
    android:text="TextView"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.924"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.468" />  
  
<Button  
    android:id="@+id/button15"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@color/colorAccent"  
    android:text="Remove"  
    android:textColor="@color/common_google_signin_btn_text_dark_default"  
    android:layout_constraintBottom_toBottomOf="parent"  
    android:layout_constraintEnd_toEndOf="parent"  
    android:layout_constraintHorizontal_bias="0.897"  
    android:layout_constraintStart_toStartOf="parent"  
    android:layout_constraintTop_toTopOf="parent"  
    android:layout_constraintVertical_bias="0.851" />
```

Shopping Buddy

Phone View



Add promo Code

Admin can add the promo code so that users can use the promo code and get some discount when the deal is active.

```
db = FirebaseFirestore.getInstance();
promoCollection = db.collection( collectionPath: "promocode");
final ArrayList<Container> itemListArray = new ArrayList<>();
promoCollection.orderBy( field: "PromoId", Query.Direction.DESCENDING).get().addOnCompleteListener( task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {
            itemListArray.add(new Container(doc.getString( field: "PromoId"), doc.getString( field: "PromoCode"), doc.get
            i++;
        }

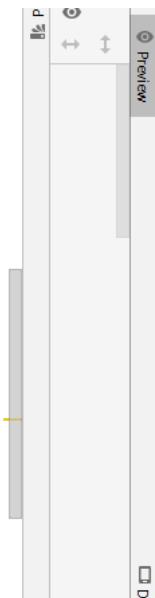
        productLV = findViewById(R.id.productsLV);
        productLV.setHasFixedSize(true);
        promoLayoutManager = new LinearLayoutManager( context: AdminHistoryAPromoActivity.this);
        promoAdapter = new AdminPromoAdapter(itemListArray, context: AdminHistoryAPromoActivity.this);
        productLV.setLayoutManager(promoLayoutManager);
        productLV.setAdapter(promoAdapter);
    }
});
```

Xml Code

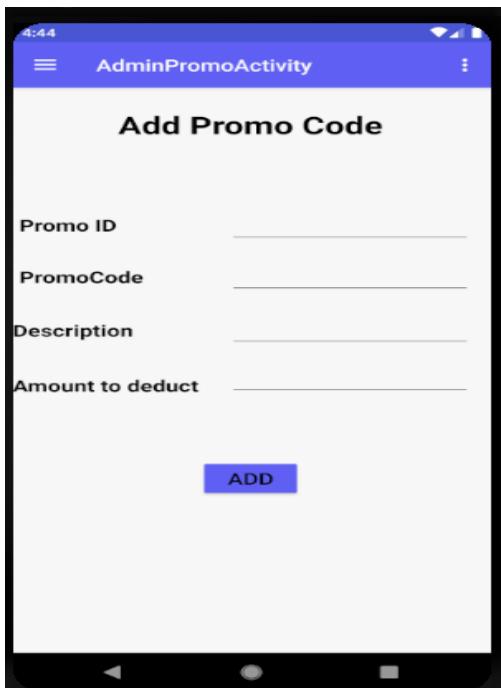
Shopping Buddy

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="395dp"
    android:layout_height="579dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<TextView
    android:id="@+id/textView63"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Added Promo code:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.125"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.053" />
```



Phone View



View/Edit Promo Code

Admin can edit the promo code and can also remove the promo code.

Shopping Buddy

```
saveChangesBTN.setOnClickListener((v) -> {
    pcode = promocode.getText().toString();
    pcost = cost.getText().toString();
    pdes=description.getText().toString();
    AlertDialog.Builder builder = new AlertDialog.Builder(context: EditorRemovePromo.this);
    builder.setMessage("Are you sure you want to save the changes").setPositiveButton(text: "Yes", (dialog, which) ->
        itemDoc.update(field: "PromoCode", pcode, ...moreFieldsAndValues: "amount to dedcut", pcost, "description", pdes);
        Toast.makeText(context: EditorRemovePromo.this, text: "Promocode updated successfully", Toast.LENGTH_LONG);
        Intent j = new Intent(packageContext: EditorRemovePromo.this, AdminHomeActivity.class);
        j.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(j);
        EditorRemovePromo.this.finish();
    ).setNegativeButton(text: "No", (dialog, which) -> {
        finish();
    }).show();
});
cancelBTN.setOnClickListener((v) -> {
    Intent i = new Intent(packageContext: EditorRemovePromo.this, SelectPromocodeActivity.class);
    startActivity(i);
});
removeBTN.setOnClickListener((v) -> {
    AlertDialog.Builder builder = new AlertDialog.Builder(context: EditorRemovePromo.this);
    builder.setMessage("Are you sure you want to delete the promocode").setPositiveButton(text: "Yes", (dialog, which) ->
        Map<String, Object> deletepromo = new HashMap<>();
        deletepromo.put("itemId", pid);
        deletepromo.put("itemName", deletePromo);
        deletepromo.put("cost", pcost);
        deletepromo.put("description", pdes);
        deletedPromo.document().set(deletepromo);
        itemDoc.delete();
        finish();
    );
});
```

Xml Code

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".EditorRemovePromo">

    <EditText
        android:id="@+id/editText30"
        android:layout_width="147dp"
        android:layout_height="54dp"
        android:ems="10"
        android:inputType="textPersonName"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.882"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.637" />

    <TextView
        android:id="@+id/textView62"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Amount to deduct:"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="22sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.097"
        app:layout_constraintStart_toStartOf="parent"
```

Shopping Buddy

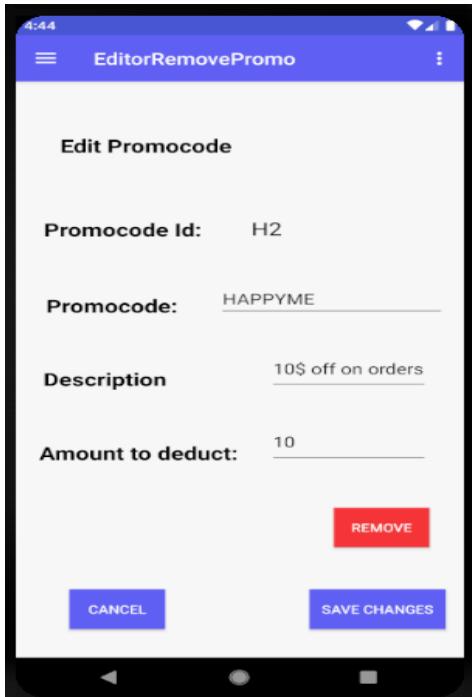


```
        app:layout_constraintVertical_bias="0.947" />

    <Button
        android:id="@+id/button14"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:text="Cancel"
        android:textColor="@color/common_google_signin_btn_text_dark_default"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.154"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.947" />

    <Button
        android:id="@+id/button15"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/colorAccent"
        android:text="Remove"
        android:textColor="@color/common_google_signin_btn_text_dark_default"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.904"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.795" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Phone View



User History:

Admin can view the list of the users for the application.

Adapter:

Shopping Buddy

```
public class UsersAdapter extends RecyclerView.Adapter<UsersAdapter.ItemsViewHolder> {
    private ArrayList<Container> itemListArray;
    private Context context;

    public UsersAdapter(ArrayList<Container> itemListArray, Context context) {
        this.itemListArray = itemListArray;
        this.context = context;
    }

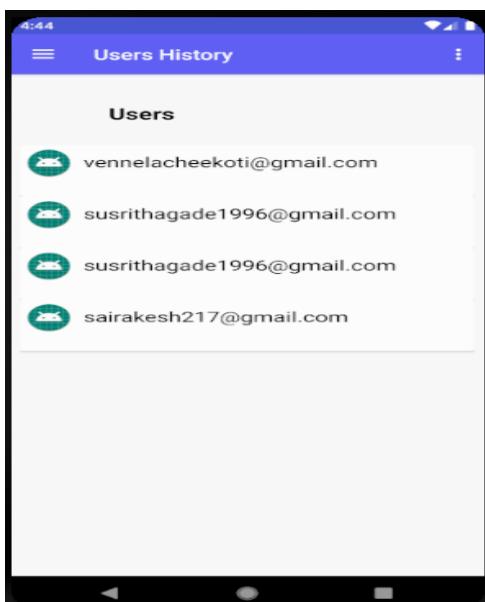
    public static class ItemsViewHolder extends RecyclerView.ViewHolder {
        public TextView userEmail;

        public LinearLayout linearLayout2;
        public ItemsViewHolder(@NonNull View itemView) {
            super(itemView);
            userEmail = itemView.findViewById(R.id.iNameTV);
            linearLayout2 = itemView.findViewById(R.id.linearLayout2);

        }
    }
    @NonNull
    @Override
    public UsersAdapter.ItemsViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        View v = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.activity_userhistory_container, viewGroup, false);
        UsersAdapter.ItemsViewHolder itemsVH = new UsersAdapter.ItemsViewHolder(v);
        return itemsVH;
    }

    @Override
    public void onBindViewHolder(@NonNull final UsersAdapter.ItemsViewHolder itemsViewHolder, final int i) {
        final Container currentItem = itemListArray.get(i);
        itemsViewHolder.userEmail.setText(currentItem.getSender());
        itemsViewHolder.linearLayout2.setOnClickListener((v) -> {
            Intent intent = new Intent(context, UserOrdersActivity.class);
            intent.putExtra("email", currentItem.getSender());
            context.startActivity(intent);
        });
    }
    @Override
    public int getItemCount() { return itemListArray.size(); }
}
```

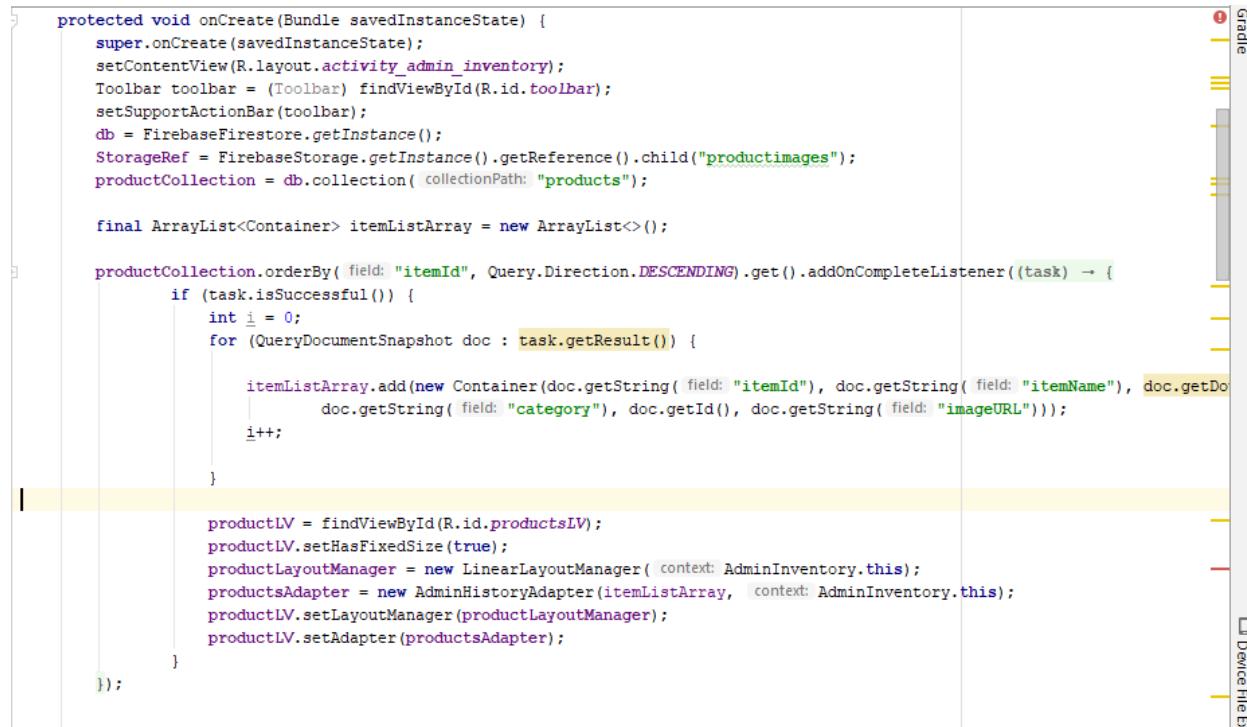
Phone View



Shopping Buddy

Admin Inventory:

Admin can view all the products that have been added to the application.



```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_admin_inventory);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    db = FirebaseFirestore.getInstance();
    StorageRef = FirebaseStorage.getInstance().getReference().child("productimages");
    productCollection = db.collection(collectionPath: "products");

    final ArrayList<Container> itemListArray = new ArrayList<>();

    productCollection.orderBy(field: "itemId", Query.Direction.DESCENDING).get().addOnCompleteListener(task) -> {
        if (task.isSuccessful()) {
            int i = 0;
            for (QueryDocumentSnapshot doc : task.getResult()) {

                itemListArray.add(new Container(doc.getString(field: "itemId"), doc.getString(field: "itemName"), doc.getString(field: "category"), doc.getId(), doc.getString(field: "imageURL")));
                i++;
            }
        }
    }

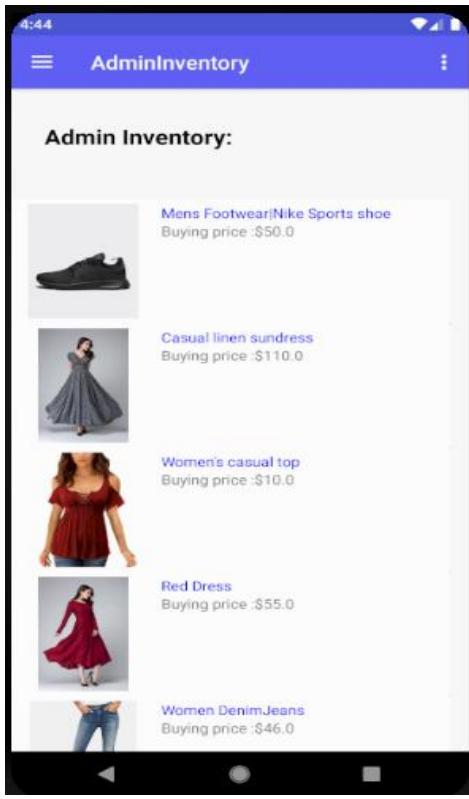
    productLV = findViewById(R.id.productsLV);
    productLV.setHasFixedSize(true);
    productLayoutManager = new LinearLayoutManager(context: AdminInventory.this);
    productsAdapter = new AdminHistoryAdapter(itemListArray, context: AdminInventory.this);
    productLV.setLayoutManager(productLayoutManager);
    productLV.setAdapter(productsAdapter);
}
};
```

Xml Code



Shopping Buddy

Phone View



Admin History:

Admin can see the history like added products, deleted products, added promo codes and deleted promo codes.

Added Products:

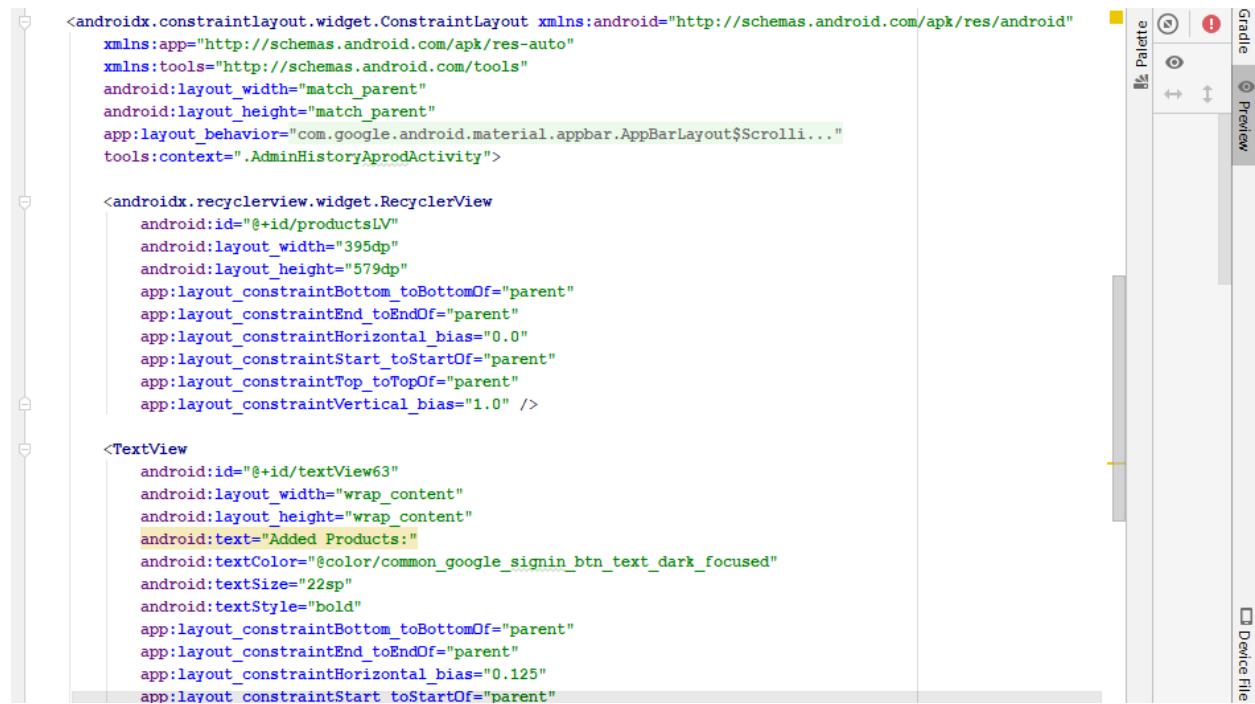
```
        productCollection.orderBy( field: "itemId", Query.Direction.DESCENDING) .get() .addOnCompleteListener((task) -> {
            if (task.isSuccessful()) {
                int i = 0;
                for (QueryDocumentSnapshot doc : task.getResult()) {

                    itemListArray.add(new Container(doc.getString( field: "itemId"), doc.getString( field: "itemName"), doc.getString( field: "category"), doc.getId(), doc.getString( field: "imageURL")));
                    i++;
                }

                productLV = findViewById(R.id.productsLV);
                productLV.setHasFixedSize(true);
                productLayoutManager = new LinearLayoutManager( context: AdminHistoryAprodActivity.this);
                productsAdapter = new AdminHistoryAdapter(itemListArray, context: AdminHistoryAprodActivity.this);
                productLV.setLayoutManager(productLayoutManager);
                productLV.setAdapter(productsAdapter);
            }
        });
    }
```

Xml Code:

Shopping Buddy



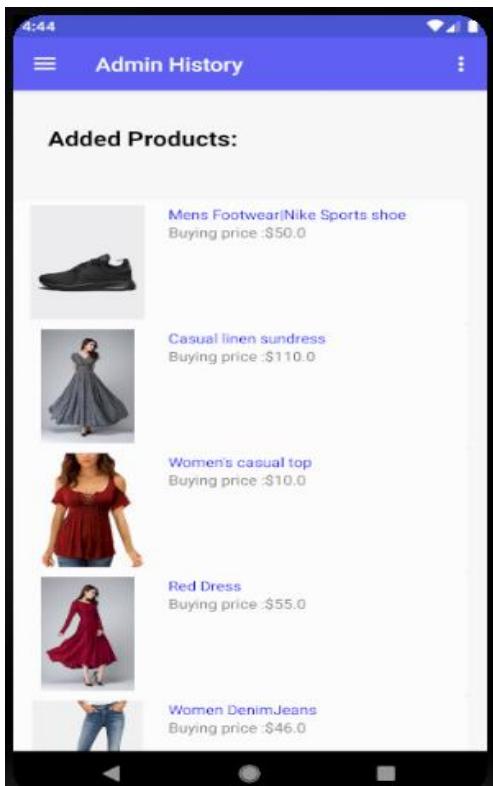
```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".AdminHistoryActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/productsLV"
        android:layout_width="395dp"
        android:layout_height="579dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="1.0" />

    <TextView
        android:id="@+id/textView63"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Added Products:"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="22sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.125"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintVertical_bias="1.0" />

```

Phone View



Shopping Buddy

Deleted Products:

```
    /**
     * final ArrayList<Container> itemListArray = new ArrayList<>();

    productCollection.orderBy( field: "itemId", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
        if (task.isSuccessful()) {
            int i = 0;
            for (QueryDocumentSnapshot doc : task.getResult()) {

                itemListArray.add(new Container(doc.getString( field: "itemId"), doc.getString( field: "itemName"), doc.getId(), doc.getString( field: "imageURL")));
                i++;
            }
        }

        productLV = findViewById(R.id.productsLV);
        productLV.setHasFixedSize(true);
        productLayoutManager = new LinearLayoutManager( context: AdminHistoryDProdActivity.this);
        productsAdapter = new AdminHistoryAdapter(itemListArray, context: AdminHistoryDProdActivity.this);
        productLV.setLayoutManager(productLayoutManager);
        productLV.setAdapter(productsAdapter);
    });
});
```

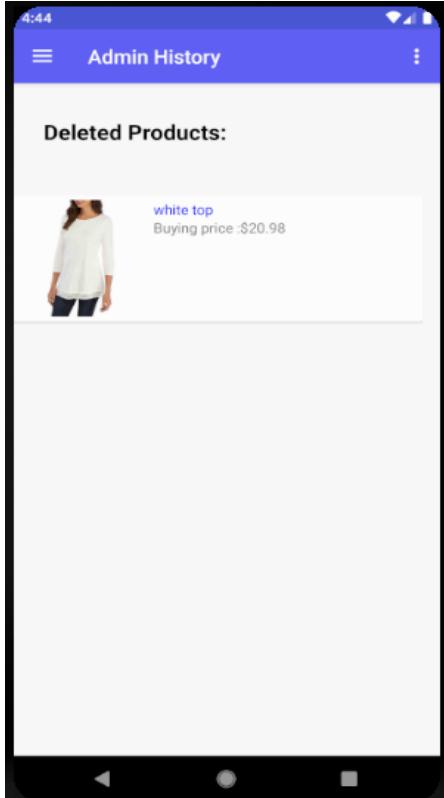
Xml Code:

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="395dp"
    android:layout_height="579dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<TextView
    android:id="@+id/textView63"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Deleted Products:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.125"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.053" />
```

Phone View

Shopping Buddy



Added Promo Codes:

```
setSupportActionBar(toolbar);

db = FirebaseFirestore.getInstance();
promoCollection = db.collection( collectionPath: "deletedPromo");
final ArrayList<Container> itemListArray = new ArrayList<>();
promoCollection.orderBy( field: "promoId", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {
            itemListArray.add(new Container(doc.getString( field: "promoId"), doc.getString( field: "promocode"), doc.get
            i++;
        }

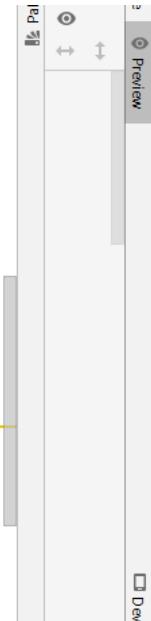
        productLV = findViewById(R.id.productsLV);
        productLV.setHasFixedSize(true);
        promoLayoutManager = new LinearLayoutManager( context: AdminHistoryDPromoActivity.this);
        promoAdapter = new AdminPromoAdapter(itemListArray, context: AdminHistoryDPromoActivity.this);
        productLV.setLayoutManager(promoLayoutManager);
        productLV.setAdapter(promoAdapter);
    }
});
```

Shopping Buddy

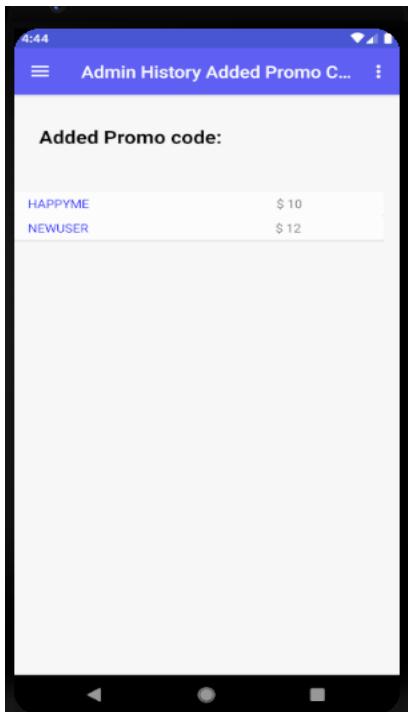
Xml Code:

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="395dp"
    android:layout_height="579dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

<TextView
    android:id="@+id/textView63"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Added Promo code:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.125"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.053" />
```



Phone View



Deleted Promo Code

```

db = FirebaseFirestore.getInstance();
promoCollection = db.collection( collectionPath: "deletedPromo");
final ArrayList<Container> itemListArray = new ArrayList<>();
promoCollection.orderBy( field: "promoId", Query.Direction.DESCENDING).get().addOnCompleteListener((task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {
            itemListArray.add(new Container(doc.getString( field: "promoId"), doc.getString( field: "promocode"), doc.get
            i++;
        }

        productLV = findViewById(R.id.productsLV);
        productLV.setHasFixedSize(true);
        promoLayoutManager = new LinearLayoutManager( context: AdminHistoryDPromoActivity.this);
        promoAdapter = new AdminPromoAdapter(itemListArray, context: AdminHistoryDPromoActivity.this);
        productLV.setLayoutManager(promoLayoutManager);
        productLV.setAdapter(promoAdapter);
    }
});

```

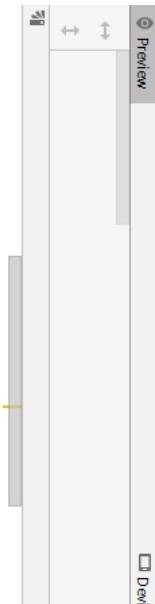
Xml Code:

```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="395dp"
    android:layout_height="579dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />

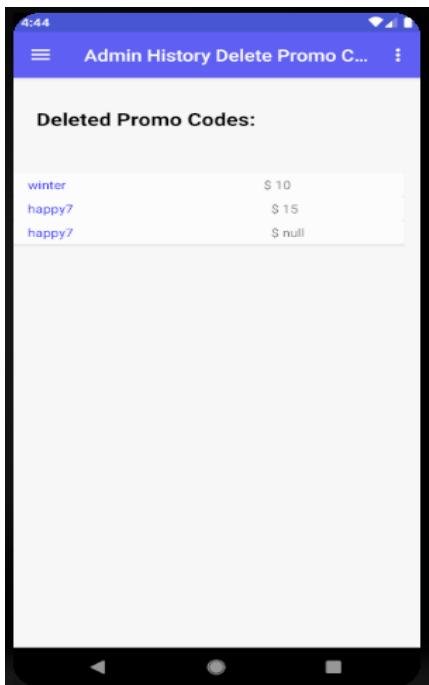
<TextView
    android:id="@+id/textView63"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Deleted Promo Codes:"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.125"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.053" />

```



Shopping Buddy

Phone View



Admin Chat:

Admin can reply to the users who have him/her in case of any queries.

```
usersCollection.orderBy( field: "Date", Query.Direction.DESCENDING).get().addOnCompleteListener(task) -> {
    if (task.isSuccessful()) {
        int i = 0;
        for (QueryDocumentSnapshot doc : task.getResult()) {
            if(!(doc.getString( field: "From").equals("shoppingbuddyseven@gmail.com"))){
                if(!(itemListArray.contains(doc.getString( field: "From")))) {
                    itemListArray.add(new Container(doc.getId(),doc.getString( field: "From")));
                    itemListArray1.add(doc.getString( field: "From"));
                    Log.d( tag: "click", msg: ""+itemListArray1.toString());
                    i++;
                }
            }
        }
    }

    productLV = findViewById(R.id.productsLV);
    productLV.setHasFixedSize(true);
    productLayoutManager = new LinearLayoutManager( context: AdminChat.this);
    productsAdapter = new AdminChatAdapter(itemListArray, context: AdminChat.this);
    productLV.setLayoutManager(productLayoutManager);
    productLV.setAdapter(productsAdapter);
});
```

Shopping Buddy

Adapter

```
public class AdminChatAdapter extends RecyclerView.Adapter<AdminChatAdapter.ItemsViewHolder> {
    private ArrayList<Container> itemListArray;
    private Context context;

    public AdminChatAdapter(ArrayList<Container> itemListArray, Context context) {
        this.itemListArray = itemListArray;
        this.context = context;
    }

    public static class ItemsViewHolder extends RecyclerView.ViewHolder {
        public TextView userEmail;

        public LinearLayout linearLayout2;
        public ItemsViewHolder(@NonNull View itemView) {
            super(itemView);
            userEmail = itemView.findViewById(R.id.iNameTV);
            linearLayout2 = itemView.findViewById(R.id.linearLayout2);
        }
    }
    @NonNull
    @Override
    public AdminChatAdapter.ItemsViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        View v = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.activity_userhistory_container, viewGroup, false);
        AdminChatAdapter.ItemsViewHolder itemsVH = new AdminChatAdapter.ItemsViewHolder(v);
        return itemsVH;
    }

    @Override
    public void onBindViewHolder(@NonNull final AdminChatAdapter.ItemsViewHolder itemsViewHolder, final int i) {

        final Container currentItem = itemListArray.get(i);
        itemsViewHolder.userEmail.setText(currentItem.getSender());
        itemsViewHolder.linearLayout2.setOnClickListener((v) -> {
            Intent intent = new Intent(context, AdminMessagesActivity.class);
            intent.putExtra("name", currentItem.getSender());
            context.startActivity(intent);
        });
    }
    @Override
    public int getItemCount() { return itemListArray.size(); }
}
```

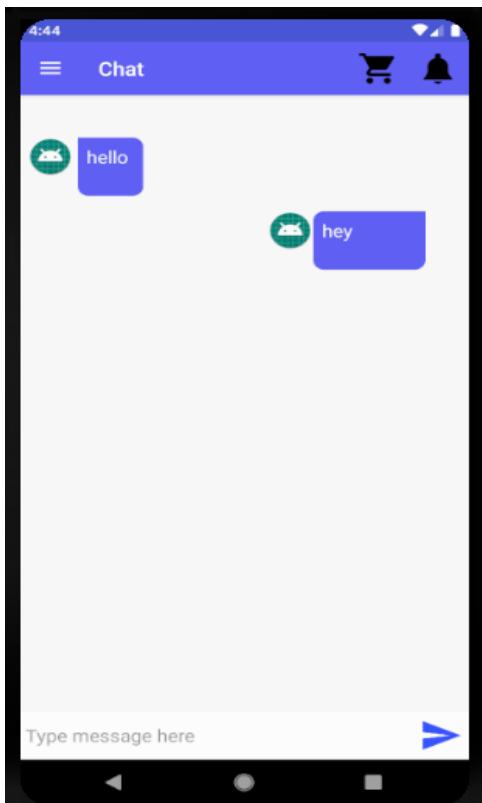
Xml Code

```
<TextView
    android:id="@+id/textView70"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Messages"
    android:textColor="@color/common_google_signin_btn_text_dark_focused"
    android:textSize="22sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.147"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.057" />

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/productsLV"
    android:layout_width="395dp"
    android:layout_height="592dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Shopping Buddy

Phone View



Admin Settings

Admin can change the password by using settings.

```
update.setOnClickListener((v) -> {
    pass = password.getText().toString();
    user.updatePassword(pass).addOnCompleteListener(task -> {
        if (task.isSuccessful()) {
            Toast.makeText(context: AdminSettingActivity.this, text: "Password updated",
            Log.d(tag: "success", msg: "password updated");
            Intent i = new Intent(packageContext: AdminSettingActivity.this, SigninActivity.class);
            startActivity(i);
        } else {
            Toast.makeText(context: AdminSettingActivity.this, text: "Password not updated");
        }
    });
});
```

Shopping Buddy

Xml Code

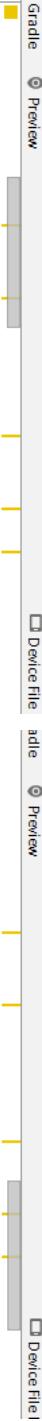
```
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior"
    tools:context=".AdminSettingActivity">

    <TextView
        android:id="@+id/textView24"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="Settings"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="22sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.102"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.062" />

    <TextView
        android:id="@+id/textView64"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.09" />

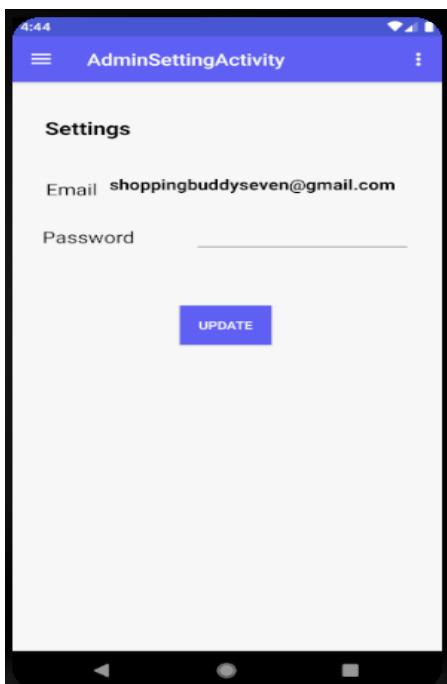
    <Button
        android:id="@+id/button23"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:text="Update"
        android:textColor="@color/common_google_signin_btn_text_dark_default"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.422" />

    <EditText
        android:id="@+id/editText29"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:textColor="@color/common_google_signin_btn_text_dark_focused"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.858"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.253" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



Shopping Buddy

Phone View



Shopping Buddy- Android App

Version 1.0.0

End User's Manual-User

Client's User Manual

Team: Android Team

Course: 44692-04

Class: GDP-02 spring 2019

TEAM MEMBERS

Sathwika Gone

Vennela Cheekoti

Rakesh Avirineni

Yaswanth Yarram

Mehar Choudhry Meenavilli

Sandeep Devineni

Susritha Gade

USER'S MANUAL

TABLE OF CONTENTS

1. General Information
2. Installation
3. Signup
4. Login
5. Forgot Password
6. Home
7. Product and its details
8. Add to cart and checkout
9. Shipping Address
10. Payment
11. Check Orders
12. Settings
13. Chat.
14. Logout

1. General Information

Shopping Buddy-Android is an android based app that allows the users to buy the products from the application that are posted by admin.

2. Installation

Download the Shopping Buddy app from online store, install it on your android device.

3. Signup

- Open the page and the first page is the signup page.
- Provide valid email address as user name
- Provide phone number.
- Provide a strong password, should have at least 6 characters
- Click continue
- Click signup button.
- User will receive an email asking for the email verification.
- Open the email and click on the link to verify the email address
- Go back to the app login page and use the email address and password to login to the app.

4. Login

- Open the App and then click on sign in link provided is the login page
- Use the registered email address in the email field
- Use the registered password in the password field
- Click on the login, user will login into the system
- It will redirect to the list of the Home page with products of different categories

5. Password Recovery Steps

- On the Login page, click on the forgot password link
- Type the registered email address in the email field shown in the redirected page
- Click the send request
- It will send a reset password email to the Users email
- In the email, click on the link to reset the password
- It will redirect to the new page and user can give the new password
- And then click on Save for the password change.
- Last step is to login by using registered email and password.

6. Home

- Login as user.
- User will be redirected to the home page which has products related to different categories and user can shop the products they like.

7. Product and its details

- After selecting the category to shop user will find the list of products with product name, price and product description.
- User can click on the product and then will be redirected to another page which shows all the details about the product like size, quantity they can choose the options and click on add to cart button which is available in the same page.

8. Cart and Checkout

- User can add all the products into the cart and he/she can click on the cart in the action bar and can see all the products added to the car.
- User can delete any products if they don't want to buy.
- After that user can check out the products which will redirect to shipping page.

9. Shipping Address

- After checkout user will be redirected to the shipping address page.
- User need to enter the details like first name, last name, address, city, state, Zip Code.
- After that user need to click on the pay button which redirects to the payment page.

10. Payment

- After user is done with entering shipping address then user will be redirected to the payment page.
- Then user need to enter the number on the card, cvv, expiry date.
- After that user can click on the pay now button with this user is done with the placing the order.

11. Orders

- Click on the menu on the action bar in any page, select the orders button
- User can view the summary of the order he/she has placed.

12. Settings

- Click on the menu on the action bar in any page, select the settings button
- User can view the privacy button where they can change the password.
- User can enter password he/she wants to change and need to confirm the password again.
- After that user can click on update button and the password has been updated successfully.

13. Chat

- Click on the menu on the action bar in any page, select the settings button.
- User can view the page which contains the contact us button and then user can click on that.
- After that user will be redirected to the contact us page where they will be displayed with the email through which they can contact.
- If user wants to contact the admin through live chat then he/she can click on the link provided for the chat.
- User will be redirected to the messaging page where they can message admin and will get reply.

14. Logout

- Click on the menu on the action bar in any page, select the logout button
- User is successfully logged out of the application.
- User will be redirected to signin page.

**Shopping Buddy - Android App
Version 2.0.0**

End User's Manual-Admin

Client's User Manual

Team: Android Team

Course: 44692-04

Class: GDP-02 spring 2019

TEAM MEMBERS

Sathwika Gone

Vennela Cheekoti

Rakesh Avirineni

Yaswanth Yarram

Mehar Choudhry Meenavilli

Sandeep Devineni

Susritha Gade

USER'S MANUAL

TABLE OF CONTENTS

1. General Information
2. Installation
3. Login
4. Add product
5. View/Edit product
6. Add promo
7. View/Edit promo
8. User History
9. Inventory
10. Settings
11. Chat
12. Logout

1. General Information

Shopping Buddy-Android is an android based app that allows the users to buy the products from the application that are posted by admin.

2. Installation

Download the Shopping Buddy app from online store, install it on your android device.

3. Login

- Open the App and then click on sign in link provided is the login page.
- Use the email address (shoppingbuddyseven@gmail.com) in the email field.
- Use the password (admin7) in the password field.
- Click on the login, admin will login into the system.
- It will redirect to the list of the Admin Home page with products of different categories.

4. Add Product

- Admin can add the products to the application.
- To add the products admin need to enter the details like product id, product name, product description and should select the category.
- Admin need to enter the price, and upload the picture of the product.
- After that click on the add product.
- Now the product has been added successfully to the application.

5. Edit Product

- Admin can edit the products in the application.
- While editing products admin can change the description of the product and price.
- After that admin can click on the save changes button then the product has been successfully edited.
- Admin can remove the product just by clicking on the remove button.
- Product has been successfully removed from application.

6. Add Promo

- Admin can add the promo codes to the application.
- To add the products admin need to enter the details like promo id, promo name, promo description, amount deducted.
- After that admin should click on add promo.
- Now the promo has been added successfully to the application.

7. Edit Promo

- Admin can edit the promo code in the application.
- While editing promo code admin can change the promo name, promo description, amount deducted and price.
- After that admin can click on the save changes button then the promo code has been successfully edited.
- Admin can remove the promo code just by clicking on the remove button.
- Promo code has been successfully removed from application.

8. User History

- Admin can click on the menu and will get a list, then click on the user history in any page.
- Admin will be redirected to the user history page where the user emails are displayed.
- After that admin can click on the any of the email then admin can see the list of the orders done by user.

9. Inventory

- Admin can click on the menu and will get a list, then click on the inventory in any page.
- Admin will be redirected to the inventory page where the list of the products added to the application are displayed.

10. Settings

- Admin can click on the navigation (top right corner) after that admin will get settings button
- Admin need to click on the settings button.
- After that admin will be redirected to the admin settings page where admin can change the password by entering password.
- After entering the password he can click on update button.
- Now the password has been successfully updated.

11. Chat

- After logging into the application admin will get the admin home page where admin can find a fab button.
- Admin need to click on that button and he/she will be redirected to the list of the users who have messaged.
- Admin can click on any of the users tab and will be redirected to the chat screen where admin can reply to the user.

12. Logout

- Click on the menu on the action bar in any page, select the logout button
- Admin is successfully logged out of the application.
- Admin will be redirected to sign in page.

.

**Shopping Buddy- Android App
Version 2.0.0
Testing User and Admin functionalities**

Team: Android Team

Course: 44692-04

Class: GDP-02 spring 2019

TEAM MEMBERS

Sathwika Gone

Vennela Cheekoti

Rakesh Avirineni

Yaswanth Yarram

Mehar Choudhry Meenavilli

Sandeep Devineni

Susritha Gade

Admin Test Cases:
Purpose and methodology

Functionality	Description	Input	Expected Output	Actual Output	Status
Login	Enter the admin name and password to get logged in.	AN: shoppingbuddyseven@gmail.com PW:admin7	Admin home Screen	Admin home screen	Passed
		AN: shoppingbuddy@gmail.com PW: admin7	Incorrect email(Remain on login page)	Login Failed. (Invalid Email ID)Remains on the same login page.	Passed
		AN: shoppingbuddyseven@gmail.com PW: admi	Password should be at least 6 characters long.	Login Failed (Invalid Password)	Passed

Shopping Buddy

Add Products	To add an product to application, admin should provide the following details: ID, picture of the item, Item name, Description, price, select category.	ID:1 Name: Lenin shirt Upload image: shirt.png Price: \$10 Category: Clothing Description: Lenin shirt with long hands and good comfort.	Product should be added to the database	Product is added to the database	Passed
---------------------	---	---	---	----------------------------------	--------

Shopping Buddy

Edit Product	To Edit/Delete a product, we first choose a product and edit the fields we want to change or delete the item.	ID:1 Name: Lenin shirt Upload image: shirt.png Price: \$9 Category: Clothing Description: Lenin shirt with long hands and good comfort.	Product should be updated to database	Product updated to database	Passed
---------------------	---	--	---------------------------------------	-----------------------------	--------

Shopping Buddy

Add promo	To add promo code admin need to add the promo id, Name, Amount to deduct	Id: 1 Name: Happy Me Amount to Deduct:\$5	Promo should be added to the database	Promo added to the database	passed
-----------	--	---	---------------------------------------	-----------------------------	--------

User application test cases

Functionality	Description	Input	Expected output	Actual output	Status
Login	Enter the user name and password to get logged in.	User name: susrithagade1996@gmail.com PW: Susritha	User Home page	User Home Page	Passed
		User name: lucy99108@gmail.com PW: 1234567	Incorrect Password	Incorrect password	Passed

Shopping Buddy

	Enter the registration details to register	Email: poojithasingham@gmail.com Password:Abc@123 Confirm Password: Abc@123 First name: Poojitha Last name:Singham Phone number:1234567890	Security questions	Security questions	Passed
Registration		Email:lohithavarma03@gmail.com Password:Abc@123 Confirm Password: Abc@124 First name: Lohitha Last name: Bhethalem Phone number:1234567890	Password doesn't match	Password doesn't match	Passed
Display product details	User clicks products to see the details	User need to select the quantity and the size of the product if the category is footwear or clothing.	User should select the quantity and size	User is able to select the quantity and size	Passed
Shipping Address	Enter the first name, last name, address, city, state, zip code	First name: Susritha Last name: Gade Address:1115 North College Dive City: Maryville State: MO Zip code: 64468	Shipping Address should be added to the database	Shipping Address added to the database	Passed
Payment Details	Enter card number , Cvv and expiry data	Card Number: 1234567890123456 Cvv: 123 Expiry data: 05/24	Order should be placed	Order placed	Passed