



ANDROID BASED APPLICATION ON CLOTHES SHOP

Ganesh Bhosale^a, Sushar Salokhe^a, Siddhi Jadhav^a, Yashwardhan Gaikwad^a, D. J. Dattawadkar^b

^a*Student, Sanjay Ghodawat Polytechnic, Atigre, India*

^b*faculty (C.S.E), Sanjay Ghodawat Polytechnic, Atigre, India*

ABSTRACT

This Project will exhibit an outline of the basic features of Mobile Augmented Reality (MAR) and the key concepts of this technology in the Fashion Retail Industry. Several obstacles such as the necessity of putting on the garments to experience its actual view have been tackled by customers throughout the process of clothes shopping. The goal is to constringe these obstacles by creating an Augmented Reality based application that will directly solve these problems. The methodology was conducted by questioning shoppers about the different hold backs faced during shopping.

Solutions for the issues are expected to be achieved by developing an easily utilizable MAR application for targeted customers. Shoppe(Application name developed by us) is an android application where user can buy fashion products online. fashion products are displayed in effective graphical user interface. This system will be like a virtual supermarket. User can select the clothes he wants to buy. User can search for the product through category like Kids, Men's, and Women's etc. After selecting products user can make payment through secure online payment gateway by using Card, UPI apps, Net-banking, Wallet or Cash on Delivery. After payment, products will be delivered to his specified address. Since this system is made in android it is easily available in smartphones. Customer who has phones supporting android can easily use the application and can buy cloths. There is no need for waiting in long queue. No need of carrying heavy bags and struggling with trolleys. Customer can purchase products sitting at home by viewing images of different fashion products.

Keywords: *Android app*

1. INTRODUCTION

Fashion may not be considered an important aspect of life, but in general, fashion is important because it reflects every culture in the world. It was a way to create different social groups and be differentiate according to your status.

There is a need of clothes which is changing seasonally. Eventually, it slowly turned into an industry - from jewellery to shoes and clothes. We do not wear the same clothes through the centuries which indicates that fashion trends have always been enathing. Fashion is a way of self-expression - one can either choose to play with it or let others decide what their outfit tells about them. But more than self-expression, fashion is a medium of self-empowerment and confidence. The correct outfit can make a person feel very satisfied, can make him/her feel good about themselves eventually radiating a sense of confidence and boosting their self- esteem. The importance of a neat and presentable dressing sense can prove to be very crucial during interviews, business meetings, social gatherings or events where it is important to make a positive first impression.

2. LITERATURE REVIEW

The traditional try on clothes in fashion retail shops has been known as one of the most time consuming tasks that includes long waiting periods of the fitting rooms and taking on and off garments. However, this survey will express the use of Augmented Reality in the Virtual Try-On to help reduce time and effort. Virtual Try-On innovation utilizing Augmented Reality can give many fascinating use cases and be deployed in various situations as in fashion retail stores. With features such as price comparison, product photos and user reviews, consumers can shop easily and smartly without even going to the stores and having such a hard time looking for the products they want.

2.1 Fashion Android Application using Augmented Reality 07, July-2020

In spite of the fact that we are in 21st century with loads of advancement in the field of innovation and improved periods of utilizing applications inside the scope of our fingertips, the desire towards a novel and better innovation for a life of simplicity ne'er comes to stop. One such new innovation is expanded reality. This is an application where clients can shop online in the most astute manner by taking a stab at various embellishments like caps, shades, pieces of jewellery and so forth simply utilizing our versatile camera to. This application gives an easy to understand frontal picture of face with no/light cosmetics and short/bound hair. The Beauty specialist's framework won't just propose the client appropriate cosmetics and hairdo, it even shows the manufactured impacts.

2.2 A fashion community and outfit recommender Android application May 2020

In this project, we propose a complete fashion community in the form of a mobile application called "Shoppe", aimed to provide a platform for fashion enthusiasts and youngsters in general, wherein users can connect, gain fashion advice and even interact with people having the same fashion interests as them. This mobile community is inspired by street fashion and blogs which will provide advice and inspiration to users in various forms such as outfit images, blogs, latest fashion news, links to online fashion stores, polls, etc. and keep them updated on the recent trends in the fashion industry. Users can upload their outfit images and ask for queries or advice from fashion experts on the community. Apart from external fashion suggestions, the user can also create apparels similar to the ones they already own in the application to create a 'virtual closet' which will be used to provide daily outfit combinations to the user. The fashion application can be used to effectively put together trendy outfits from the existing apparel already owned by the user for a more presentable and trendy appearance.

3. OBJECTIVE AND SCOPE

3.1 Objective of Project

- To cut off effort by testing the garment product in the real world environment through the virtual world on top of it.
- The application gives all the information about the garment products and provides better services for customer.
- To resides on a great effortless experience and entirely benefitting from the testing technique which also provides full evaluation of the product without the need of the real world testing (putting on the garment/accessory).
- Provides facility to the customer to payment by the cash or Debit/Credit card or through Net banking.
- It's providing full details about the Fashion product and related information about the product like cost, quality, quantity etc.
- By this system we can advertise and send procure to a lot of customer by sending email.

3.2 Scope of Project

This scope will be evaluating the prospected advantages of the utilization of Augmented Reality in Fashion Retailing from both a consumer and a business perspective. The purpose of the Mobile Augmented Reality Application in the fashion retail shopping is to help enhance the customer's in-store experience. The developed application is intended to help diminish problems faced by customers in fashion shopping experience by the valuable utilization of the Augmented Reality Technology. This system allows to the customers to maintain cart for add or remove the product over the internet.

4. CORE TECHNOLOGY

2.1 Project structure

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

- Manifests: Contains the AndroidManifest.xml file.
- Java: Contains the Java source code files, including JUnit test code.
- Res: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

2.2 Features of Android Studio

- It has a flexible Gradle-based build system.
- It has a fast and feature-rich emulator for app testing.
- Android Studio has a consolidated environment where we can develop for all Android devices.

- Apply changes to the resource code of our running app without restarting the app.
- Android Studio provides extensive testing tools and frameworks.
- It supports C++ and NDK.
- It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google Cloud Messaging and App Engine.

2.3 Gradle builds system

Gradle build used as the foundation of the build system in Android Studio. It uses more Android-specific capabilities provided by the Android plugin for Gradle. This build system runs independently from the command line and integrated tool from the Android Studio menu. We can use build features for the following purpose:

- Configure, customize, and extend the build process.
- We can create multiple APKs from our app, with different features using the same project and modules.
- Reuse resource and code across source sets.

5. XML LANGUAGE

XML stands for Extensible Markup Language, which gives us a clue to what it does. A markup language is slightly different from a programming language. Whereas a programming language (C#, C++, Java, Kotlin, Python, BASIC) will allow you to define behaviors, interactions, and conditions; a markup language is used more to describe data, and in this case, layouts. Programming languages create dynamic interactions, whereas markup languages generally handle things like static user interfaces. When you create a new project in Android Studio, you will be greeted by a hierarchy of different files and folders, which can be a little daunting for complete beginners. It's a rather jarring introduction to XML.

5.1 Xml in Android App

XML describes the views in your activities, and Java tells them how to behave. To make changes to the layout of your app then, you have two main options. The first is to use the Design view. Open up the activity main.xml file in Android Studio and get your first introduction to XML. You'll notice there are two tabs at the bottom of that window: Design and Text. The Text view will show you the actual XML code, but the Design view will let you manually edit the layout by dragging and dropping elements into the render of your activity. XML files can also help store strings. Using the Design view is easier for beginners, though it can lead to complications. For one, you will run into the limitations of XML early on when the designer refuses to let you drop items into specific places. Without the knowledge of why, this can make designing your app an exercise in frustration.

5.2 Used Layouts and Containers

Android Layout is used to define the user interface that holds the UI controls or widgets that will appear on the screen of an android application or activity screen. Generally, every application is a combination of View and View Group. As we know, an android application contains a large number of activities and we can say each activity is one page of the application. All the elements in a layout are built using a hierarchy of View and View Group objects.

- **Relative Layout:** RelativeLayout is a view group that displays child views in relative positions. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout area (such as aligned to the bottom, left or center).
- **Linear Layout:** LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. You can specify the layout direction with the android:orientation attribute.
- **androidx.cardview.widget.CardView:** Apps often need to display data in similarly styled containers. These containers are often used in lists to hold each item's information. The system provides the CardView API as an easy way for you to show information inside cards that have a consistent look across the platform.
- **Frame Layout:** FrameLayout is a ViewGroup subclass, used to specify the position of View elements it contains on the top of each other to display only a single View inside the FrameLayout.
- **Constraint Layout:** ConstraintLayout is a ViewGroup subclass, used to specify the position of layout constraints for every child View relative to other views present. A ConstraintLayout is similar to a RelativeLayout, but having more power.
- **View:** A View is defined as the user interface which is used to create interactive UI components such as TextView, ImageView, EditText, RadioButton, etc., and is responsible for event handling and drawing. They are Generally Called Widgets.

5.3 Used Components

- **TextView:** A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor, however the basic class is configured to not allow editing.
- **TextInputLayout:** TextInputLayout is a view container that is used to add more features to an EditText. It acts as a wrapper for EditText and has some features like: Floating hint. Animation that can be disabled or enabled.
- **EditText:** A EditText is an overlay over TextView that configures itself to be editable. It is the predefined subclass of TextView that includes rich editing capabilities.
- **Button:** A button consists of text or an icon (or both text and an icon) that communicates what action occurs when the user touches it.
- **ImageButton:** Image Button is a user interface control that is used to display a button with an image and to perform an action when a user clicks or taps on it.
- **ImageView:** Displays image resources, for example Bitmap or Drawable resources. ImageView is also commonly used to apply tints to an image and handle image scaling.

6. DEVELOPMENT

6.1 Java

Android development is a popular buzz in the Java programming world. It's Android, which keeps Java at the forefront of the last couple of years. reach a mass, Android offers that opportunity to you. Millions of Android phones are available and they keep increasing, with pace, much higher than iPhone or iOS. What all this means is, it does make a lot of sense to learn Android programming for Java programmers, and this article is about that, but this is also one of the good reasons to learn Java programming. This tutorial will give you a basic idea of How Android works? not detailed but a good overview.

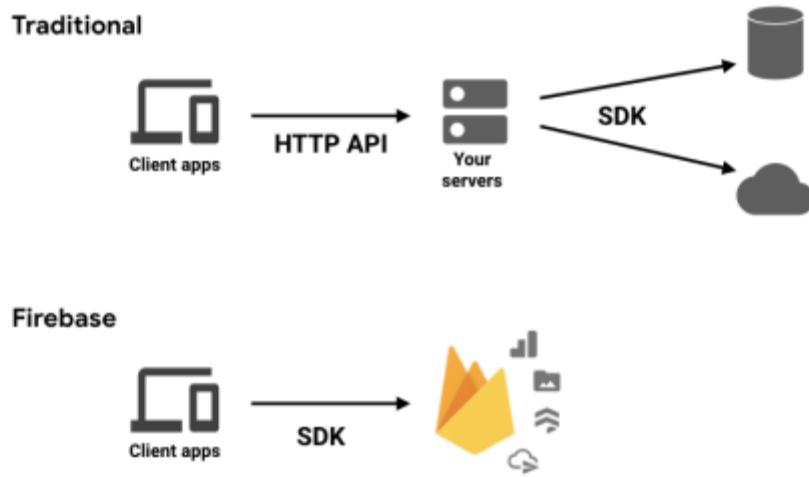
6.2 Used Listener

- **OnClickListener:** In Android, the OnClickListener() interface has an onClick(View v) method that is called when the view (component) is clicked. The code for a component's functionality is written inside this method, and the listener is set using the setOnClickListener() method.
- **ValueEventListener:** public interface ValueEventListener. Classes implementing this interface can be used to receive events about data changes at a location. Attach the listener to a location us addValueEventListener(ValueEventListener) .
- **OnSuccessListener:** public abstract void onSuccess (ResultT result). Called when the Task completes successfully.
- **addOnFailureListener:** Listener called when a Task fails with an exception
- **OnCompleteListener:** public abstract void onComplete (Task<ResultT> task). Called when the Task completes

7. FIREBASE DATABASE

7.1 Introduction to Firebase

Firebase is a toolset to "build, improve, and grow your app", and the tools it gives you cover a large portion of the services that developers would normally have to build themselves, but don't really want to build, because they'd rather be focusing on the app experience itself. This includes things like analytics, authentication, databases, configuration, file storage, push messaging, and the list goes on. The services are hosted in the cloud, and scale with little to no effort on the part of the developer. The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our Apple platforms, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.



Fig(7.1)

8. UML DIAGRAMS

8.1 Activity Diagrams

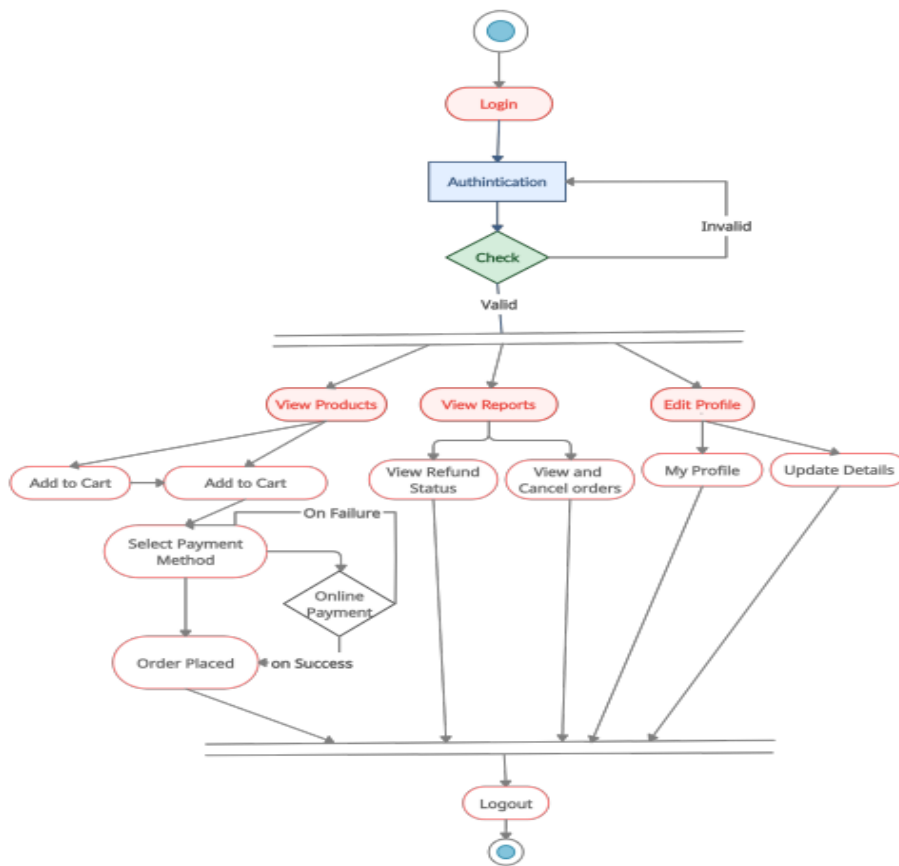


Figure 8.1.1 Activity Diagram Customer's Panel



Figure 8.1.2 Activity Diagram Admin Panel

8.2 Use Case Diagram

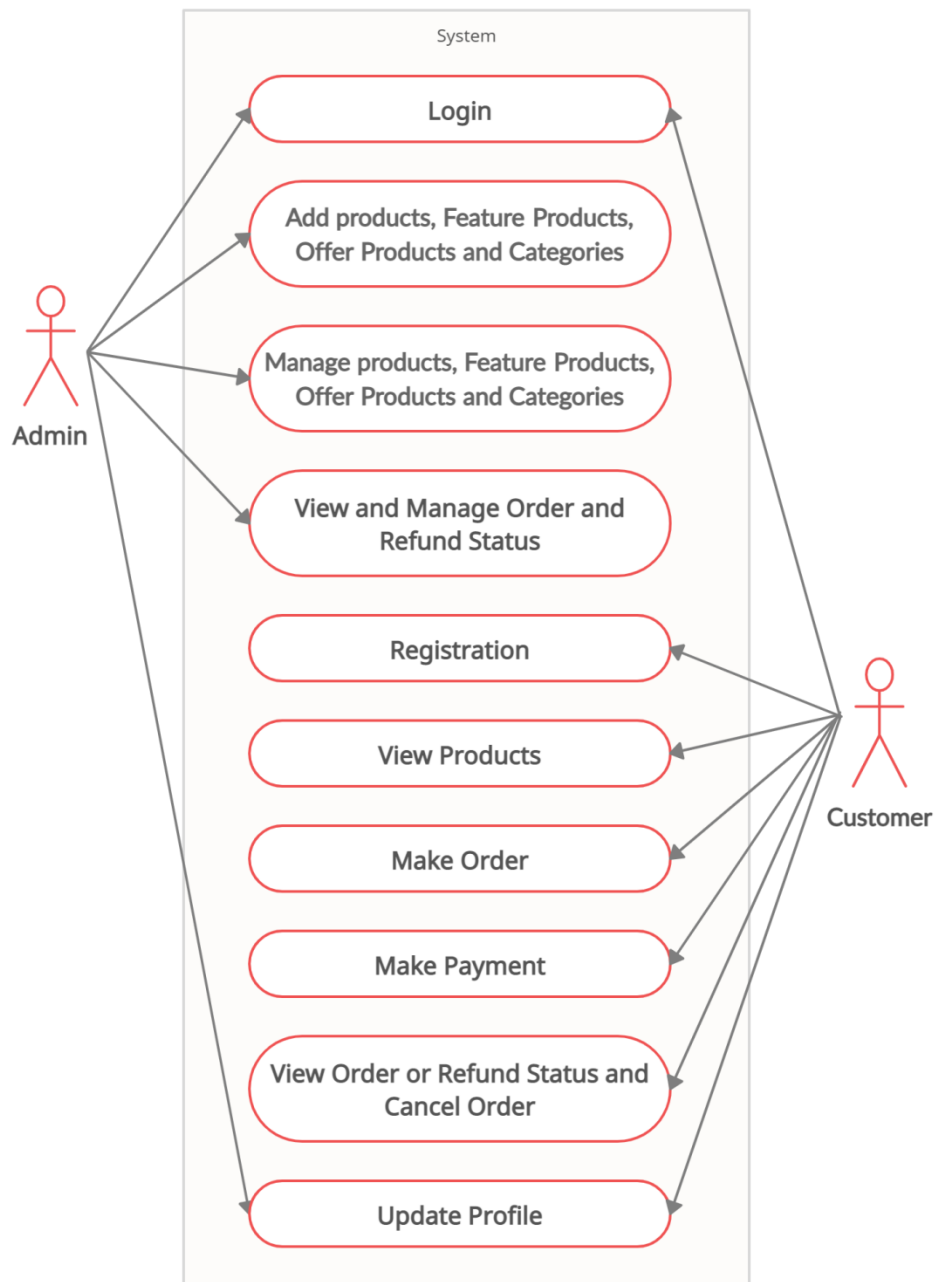


Figure 8.2.1 Use Case Diagram

8.3 Data Flow Diagrams



Figure 8.3.1 DFD Level 0

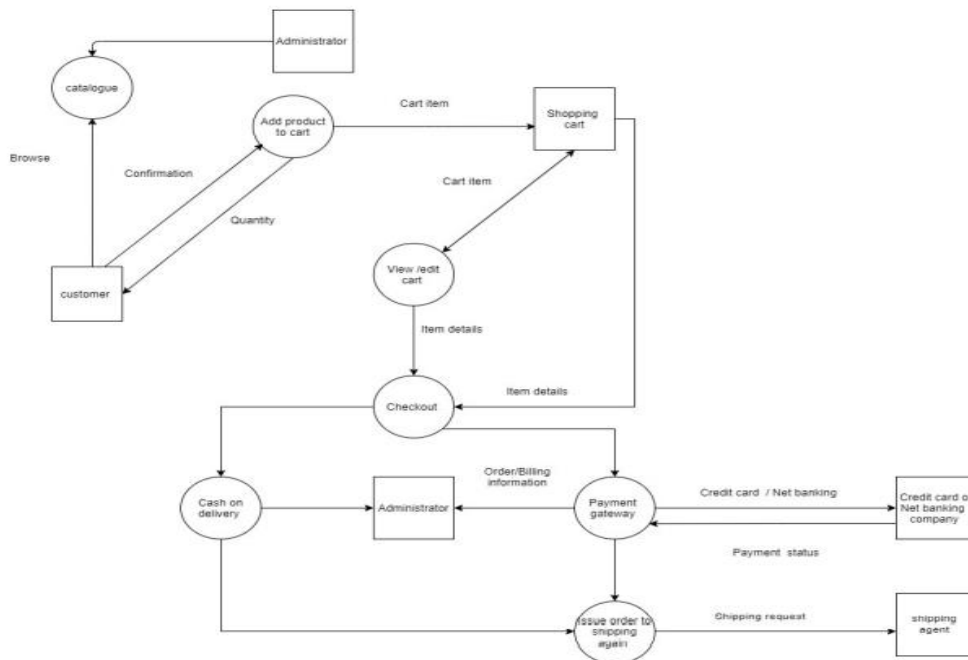


Figure 8.3.2 DFD Level 1

9. SYSTEM CODING

9.1 splash Screen.java

```

package com.example.digital_docs;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class Splash_Screen extends AppCompatActivity {
    TextView label;
    FirebaseAuth mAuth;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        int SPLASH_SCREEN_TIME_OUT=3000;
        super.onCreate(savedInstanceState);
        label=findViewById(R.id.label);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_splash__screen);
        Animation a = AnimationUtils.loadAnimation(this, R.anim.zoom);
        a.reset();
        TextView tv = (TextView) findViewById(R.id.label);
        tv.clearAnimation();
        tv.startAnimation(a);

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                mAuth = FirebaseAuth.getInstance();
                FirebaseUser user = mAuth.getInstance().getCurrentUser();
                if (user == null) {
                    startActivity(new Intent(getApplicationContext(), LoginActivity.class));
                }
                else {
                    startActivity(new Intent(getApplicationContext(), MainActivity.class));
                }

                finish();
            }
        }, SPLASH_SCREEN_TIME_OUT);
    }
}

```

9.2 activity splash screen.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"

```

```
android:layout_height="match_parent"
android:orientation="horizontal"
android:background="@drawable/splash_1"
tools:context=".Splash_Screen">

<TextView
    android:id="@+id/label"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:layout_marginTop="260dp"
    android:clickable="false"
    android:fontFamily="sans-serif-black"
    android:gravity="center"
    android:lineSpacingExtra="20dp"
    android:text="WELCOME\n TO \n SHOPPE"
    android:textColor="@color/white"
    android:textSize="25dp"
    android:textStyle="bold"
    android:layout_centerInParent="true"/>

</RelativeLayout>
```

9.3 MainActivity.java

```
package com.example.digital_docs;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.SearchView;
import androidx.appcompat.widget.Toolbar;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.example.digital_docs.Admin.Admin_Main;
import com.example.digital_docs.domain.orders;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.protobuf.StringValue;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
```

```

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;
    DatabaseReference databaseUsers;
    FirebaseAuth mAuth;
    NavController navController;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mAuth = FirebaseAuth.getInstance();
        FirebaseAuth mAuth = FirebaseAuth.getInstance().getCurrentUser();
        if (user == null) {
            startActivity(new Intent(getApplicationContext(), LoginActivity.class));
        }
        NavigationView navigationView = findViewById(R.id.nav_view);
        View headerView = navigationView.getHeaderView(0);
        TextView name = (TextView) headerView.findViewById(R.id.Name);
        TextView email = (TextView) headerView.findViewById(R.id.Email);
        if (user != null) {
            String emaila = user.getEmail();
            email.setText(emaila);
            FirebaseDatabase database = FirebaseDatabase.getInstance();
            databaseUsers = database.getReference("user").child(mAuth.getUid()).child("User Details");
            databaseUsers.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {

                    for(DataSnapshot ds: snapshot.getChildren()){
                        String username = ""+ds.child("fname").getValue();
                        name.setText(username);
                    }
                }
                @Override
                public void onCancelled(@NonNull DatabaseError databaseError) {
                }
            });
        }
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        mAppBarConfiguration = new AppBarConfiguration.Builder(
            R.id.nav_home,R.id.nav_orders, R.id.nav_gallery, R.id.nav_slideshow, R.id.logout)
            .setDrawerLayout(drawer)
            .build();
        navController = Navigation.findNavController(this, R.id.nav_host_fragment);
        Bundle extras=getIntent().getExtras();
        if(extras != null) {
            String s = getIntent().getStringExtra("order_details");
            if (s.equals("nav_orders")) {
                navController.navigate(R.id.nav_orders);
            }
        }
        NavigationUI.setupActionBarWithNavController(this, navController, mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        getMenuInflater().inflate(R.menu.admin_panel, menu);

        if ( mAuth.getInstance().getCurrentUser().getUid().equals("IzKaO636exb6L7X3huGP3yR7xD93")) {
            MenuItem admin_panel = menu.findItem(R.id.admin_panel);
            admin_panel.setVisible(true);
        }
    }
}

```

```

admin_panel.setOnMenuItemClickListener(new MenuItem.OnMenuItemClickListener() {
    @Override
    public boolean onOptionsItemSelected(MenuItem menuItem) {
        startActivity(new Intent(getApplicationContext(), Admin_Main.class));
        return false;
    }
});
}
return true;
}

@Override
public boolean onSupportNavigateUp() {
    NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
    return NavigationUI.navigateUp(navController, mAppBarConfiguration)
        || super.onSupportNavigateUp();
}

public void profile(MenuItem item) {
    startActivity(new Intent(getApplicationContext(), UpdateProfile.class));
}

public void cart(MenuItem item) {
    startActivity(new Intent(getApplicationContext(), CartActivity.class));
}

@Override
public void onBackPressed() {
    String s= String.valueOf(navController.getCurrentDestination().getLabel());
    System.out.println(s);
    if (s.equals("Home")) {
        new AlertDialog.Builder(this)
            .setTitle("Really Exit?")
            .setMessage("Are you sure you want to exit?")
            .setNegativeButton("No", null)
            .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface arg0, int arg1) {
                    MainActivity.super.onBackPressed();
                    moveTaskToBack(true);
                }
            })
            .create().show();
    } else {
        navController.navigate(R.id.nav_home);
    }
}
}
}

```

9.4 Activity_main.xml

```

10 <?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

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

```

```
<com.google.android.material.navigation.NavigationView
  android:id="@+id/nav_view"
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:layout_gravity="start"
  android:fitsSystemWindows="true"
  app:headerLayout="@layout/nav_header_main"
  app:menu="@menu/activity_main_drawer" />
</androidx.drawerlayout.widget.DrawerLayout>
```

9.5 Screenshots

Starting Screen

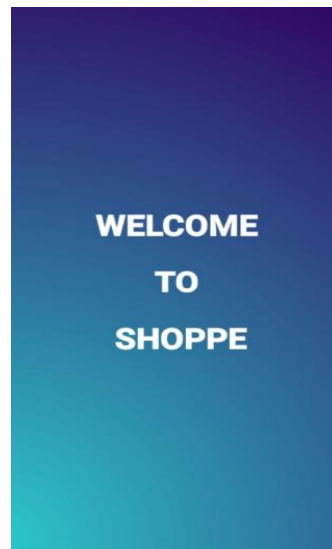


Figure 9.5.1

This is a Starting Screen

Registration

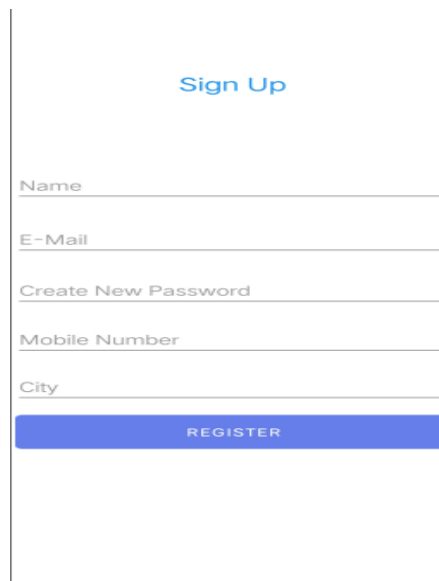
A registration form with a white background and a thin vertical line on the left. At the top, the text "Sign Up" is displayed in blue. Below it are five input fields with labels: "Name", "E-Mail", "Create New Password", "Mobile Number", and "City". At the bottom, there is a blue button with the text "REGISTER" in white, uppercase letters.

Figure 9.5.2

This page Provides Registration for new User or Customer.

Login Page

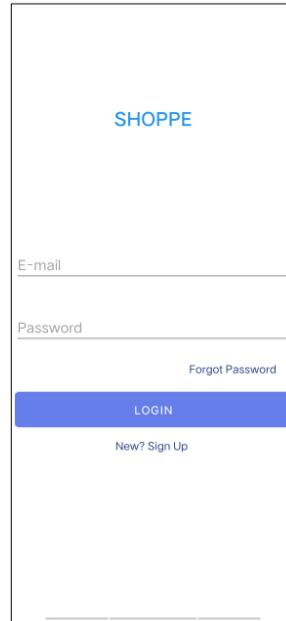


Figure 9.5.3

This page Provides Login for Registered User or Customer.

Home page

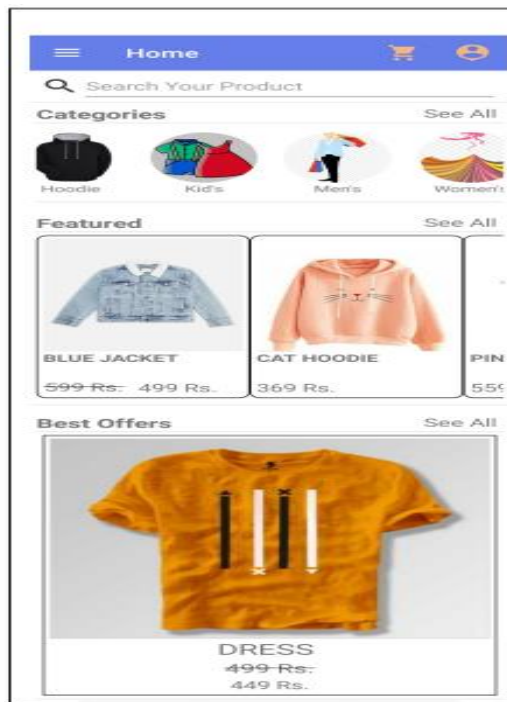


Figure 9.5.4

This Page Provides short information about our store and its products.

Product Details Page

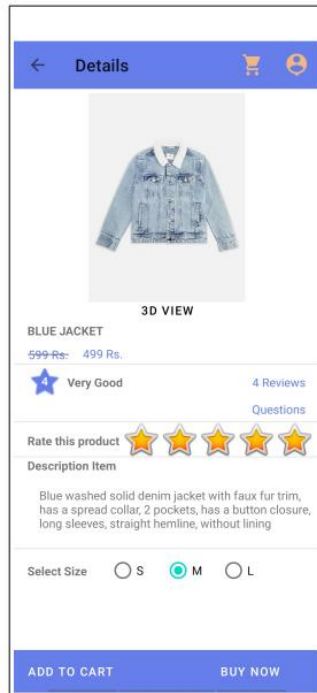


Figure 9.5.5

Here User or Customer can see the products details of particular product.

Cart Page

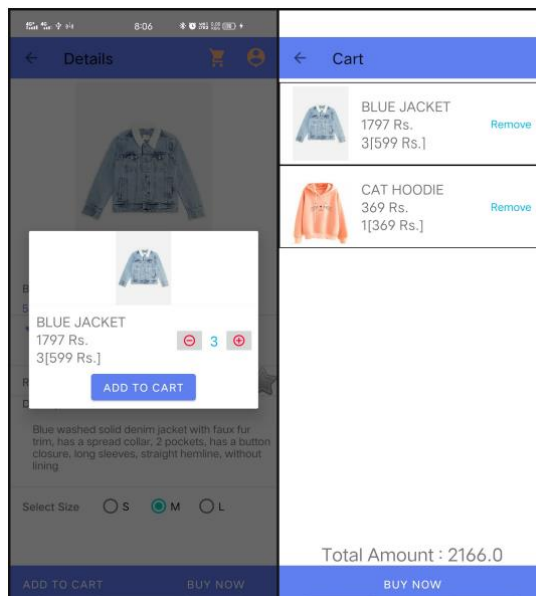


Figure 9.5.6

Here User or Customer can see the products which are added to cart.

Checkout Page

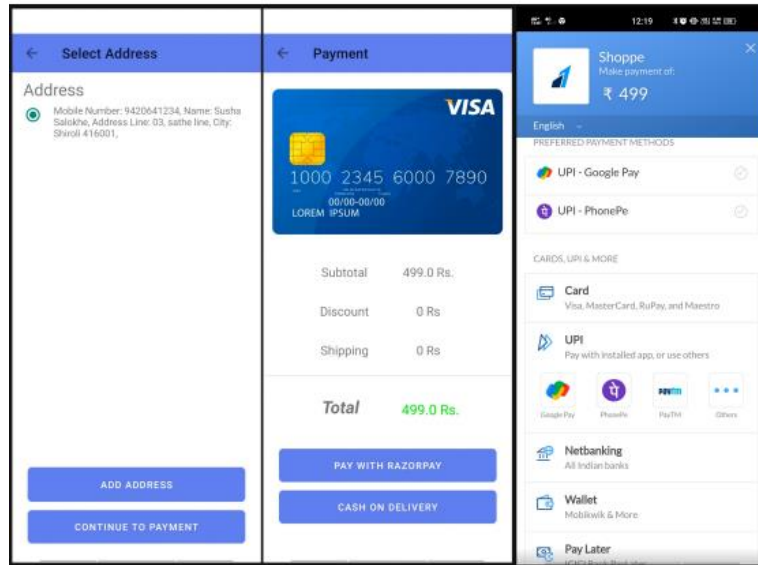


Figure 9.5.7

This is a checkout page here user or customer fills shipping/billing details and make payments.

Order Details

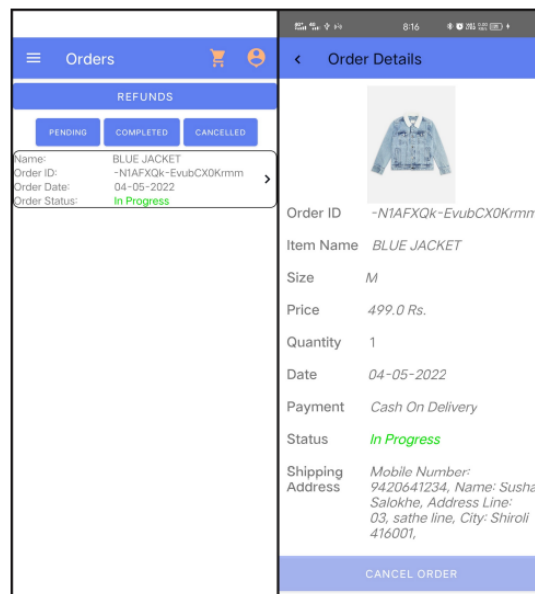


Figure 9.5.8

Here User or Customer can see his order details after checkout.

Add and Manage Product

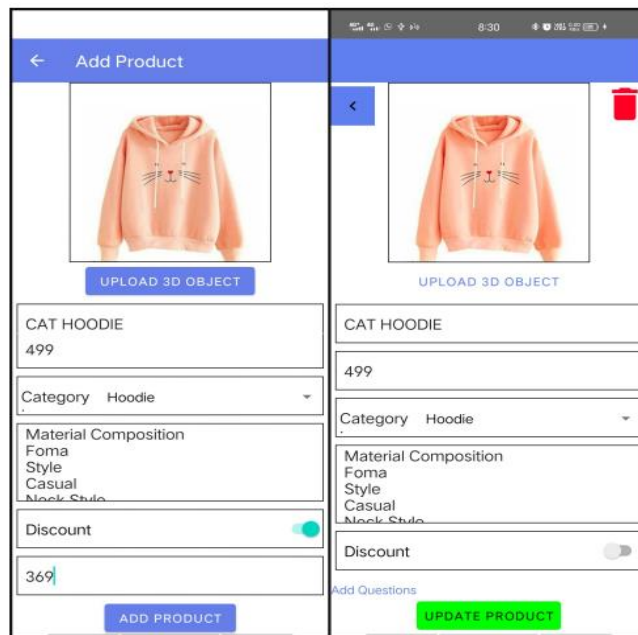


Figure 9.5.9

Here Admin can see add new products and manage it accordingly.

10. TESTING AND RESULTS OF PROJECT

10.1 Testing Methodologies

1. Black box Testing
2. White box Testing

10.2 Levels of Testing

1. Unit Testing
2. Integration Testing
3. System Testing

10.3 Testing Methodologies

1. Black box Testing

It is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.

Usually Test Engineers are involved in the black box testing.

2. White box Testing

It is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually The Developers are involved in white box testing.

10.4 Levels of Testing

1. Unit Testing

Unit Testing concentrates on the verification of the smallest element of the program i.e. Module. In this testing all control paths are tested to identify errors within the bounds of the module. The important goal of unit testing is to isolate each part of the program and show individual parts are correct. It is very easy to perform and requires less amount of time because the modules are smaller in size. In unit testing it is possible that the outputs produced by one unit become input for another unit hence, if incorrect output produced by one unit is provided as input to the second unit then it also produces wrong output. If this process is not corrected, the entire software may produce unexpected outputs. To avoid this, all the units in the software are tested independently using unit testing. In unit testing, the units are tested to ensure that they operate correctly. In software engineering the unit testing is not just performed once during software development, but repeated whenever the software is modified.

2. Integration Testing

When unit testing is complete, integration testing begins. In integration testing the tested units are combined together to form system as whole. The aim of this testing is to ensure that all modules are working properly according to user's requirements when they are combined. The integration test takes all tested individual modules, integrate them, test them again and develop the software. It ensures that all modules work together properly and transfer accurate data across their interfaces.

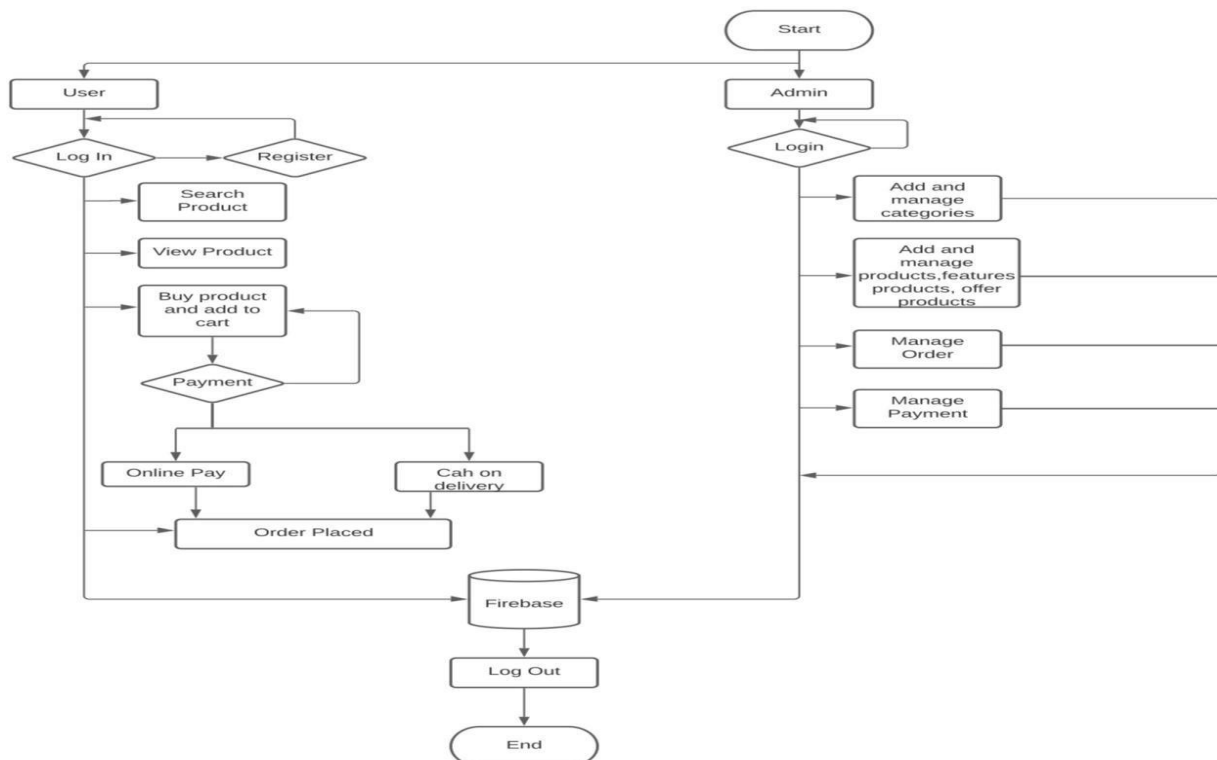
3. System Testing

System testing is the next level in the testing and tests the system as a whole. Once, all the components are integrated, the application as a whole is tested to see that it meets Quality Standards. This type of testing is performed by a specialized testing team. System testing can be defined as "a testing conducted on a complete, integrated system to ensure that the system is according to its specified requirement".

11. METHODOLOGY

System Architecture

There are two main components in the fashion application, Admin module and Shop module. Admin can add Categories, Feature Products, Offers or Products and manage it accordingly also admin can manage Orders. In Shop module customer can see Products, Feature Products, Offers and he can buy product he wants also customer can see order summery.



Login

Login module will authenticate the admin or customer which has to be login. The Login Module is a portal module that allows users to type a Email and password to log in.

Registration

In this module the customer can register himself. He needs to provide his details like name, address, phone number and he needs to create a username and password.

Search

A search module is for search required products using search bar. search module helps user to find his product he wants.

12. SYSTEM CONFIGURATION**Hardware Requirement**

1. Android Device
2. RAM - 1 GB(min)
3. Storage - 1 GB(min)
4. Processor - KitKat 4.4 and above

Software Requirement

1. Operating System -Windows 7 and above, Android (Kitkat 4.4 min)
2. Languages - Java, XML
3. Tool - Android Studio

Database Requirement

1. Firebase

13. ADVANTAGES AND DISADVANTAGES OF PROPOSED SYSTEM**Advantages**

1. More variety.
2. You get an electronic record of the receipt of your purchase, which makes record-keeping much easier.
3. It saves time.
4. You get to try things on in the comfort of your own home.
5. Rare products are available.
6. Freedom of choice.

Disadvantages

1. Spending too much time online.
2. Return can be complicated.
3. No support for local retail.
4. You need a secure internet connection.
5. Cyber crime.
6. Lack of opportunity to bargain.

14. CONCLUSION

This application will be for betterment to customer for their shopping purpose. While doing this project we realized different problems occurring regarding to customers view, we are trying to solve those problems through this application . While doing such a things are learning a lot of different things, technologies which are upskilling us. If is found useful then this application can be upscaled to different institutes and solving their problem as well.

REFERENCES

- [1] An Android Augmented Reality Application for Retail Fashion Shop-ping· January 2019
- [2] Fashion Android Application using Augmented Reality 07, July-2020
- [3] MODA-A fashion community and outfit recommender Android application May 2020
- [4] “Mobile Fashion” Application Exploring design solution for the “Outfit-Centric Accessories” concept November 2011
- [5] Alexander, A. S. G., Valdi, S., Albertus, F., Heri, N., Widodo, B., Herman, T., Muhammad, (2019) “Development of Smart Trolley System Based on Android Smartphone Sensors” Jakarta: 4th International Conference on Computer Science and Computational Intelligence 2019 Google Scholar
- [6] Altarteer, S., Vassilis, C., Harrison, D., and Chan, W. (2016) “Product Customisation: Vir- tual Reality and New Opportunities for Luxury Brands Online Trading” 21st International Conference on Web3D Technology, Anaheim, California: ACM: 173-174. Google Scholar.