Jersey

(An Android Application)

Project Report submitted to

St. Xavier's College – Autonomous Mumbai

For the partial fulfillment for the award of the degree of

Bachelor of Science (BSc) in

Information Technology

By

Vivek Ramayan Chaurasia

(UID : 205102)

Under the Supervision of

Prof. Miss. Lydia Fernandes

INFORMATION TECHNOLOGY

ST. XAVIER'S COLLEGE (AUTONOMOUS),

MUMBAI-400001, INDIA

April 2023

PROJECT CERTIFICATE

This is to certify that the project entitled **Jersey - An Android Application** undertaken at the Information Technology Department of St. Xavier's College – Autonomous Mumbai has been submitted by: **Vivek Ramayan Chaurasia (UID No: 205102)**, in partial fulfillment of Bachelor's in Information Technology degree (Semester VI) Examination. It is further certified that he has completed all required phases of the project.

(Internal Guide) **(Internal Examiner)** **(External Examiner)**

(HOD – Information Technology department) **College Seal**

Declaration

I, **Vivek Ramayan Chaurasia (UID No: 205102)**, do hereby, certify that:

1) that the project report titled, "Jersey - An Android Application" which is being submitted in partial fulfillment of the requirements for the Degree of Bachelor of Science with a specialization in Information Technology is the result of the **original work** carried out by us under the guidance of the Ms, Lydia Fernandes, faculty of Information Technology Department, St. Xavier's College, Mumbai-01.
2) This project has not previously formed the basis for the award of any degree, diploma, or certificate of this college or of any other college or university.
3) Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them by providing references of them in the project documentation.
4) From the plagiarism test, it is found that the similarity index of whole submission is within ___%. [optional]

Date:

Place:

_____

Vivek Ramayan Chaurasia
UID: 205102

Acknowledgement

I would like to thank the professors of the IT department for their guidance throughout the three years of my studies in St.Xavier's College - Autonomous, Mumbai.

I would also like to thank the principal for giving a fair environment to study in and the people who have helped me most throughout my project.

I am grateful to my Prof. Miss. Lydia Fernandes for nonstop support for the project. I can't say thank you enough for her tremendous support and help.

I owe my deep gratitude to our HOD of Information Technology Department Mr. Roy Thomas who took keen interest in our project work and guided me all along, till the completion of our project work by providing all the necessary information for developing a good system.

At last, but not the least I want to thank all my friends who helped/treasured me out in completing the project, where they all exchanged their own interesting ideas, thoughts and made this possible to complete my project with all accurate information. I wish to thank my parents for their personal support or attention who inspired/encouraged me to go my own way.

Abstract

Jersey is an Android e-commerce application that allows sports enthusiasts to purchase jerseys of their favourite teams online. The app features a shopping cart that enables users to select and add the items they wish to purchase. Once the user has completed their selections, they can proceed to check out.

Upon successful payment, users will receive an order confirmation email within the next 24 hours. If they have any queries related to their order, they can contact the support team via email at jerseysupport@gmail.com.

.

TABLE OF CONTENTS

7. CONCLUSION

List of Figures

1. Introduction

1.1.Background

Jersey is an Android e-commerce application designed to assist sports enthusiasts in purchasing jerseys of their favourite teams online. E-commerce is a business model that uses the internet to sell goods and services to customers. With the proliferation of smartphones and mobile devices, e-commerce is becoming increasingly popular, and mobile e-commerce or m-commerce is gaining more traction.

Jersey's primary goal is to provide a convenient and seamless experience for users who want to purchase sports jerseys online. The app features a user-friendly interface that enables users to select and add the items they wish to purchase to their shopping cart. Payment for the selected items can be made quickly and easily through the app's integrated payment gateway.

E-payment technologies have made it possible for customers to make payments online with just a few clicks. Jersey takes advantage of this technology to provide a fast and secure payment process for its users. Upon successful payment, users will receive an order confirmation email within the next 24 hours, and they can contact the support team via email at jerseysupport@gmail.com if they have any queries related to their order.

Overall, Jersey provides a convenient and hassle-free way for sports enthusiasts to purchase their favourite teams' jerseys online, making it an excellent choice for anyone looking for a reliable and user-friendly e-commerce app for Android.

2. Objectives

The objective of Jersey - an Android e-commerce application is to provide users with a convenient and user-friendly platform for purchasing sports jerseys online. The app aims to achieve the following objectives:

✔ Easy Product Navigation and Selection

✔ Simple and Secure Payment Gateway

✔ User-Friendly Interface and Time-Effective Purchase Process

✔ Automated Order Confirmation Email

✔ Efficient Order Tracking and Management

✔ Options to Update Products, Categories, and Website Settings at the backend on nameCheap server with an actual domain

✔ User-Centric Features like Individual Product Quantity Modification, Best Selling Products List, and News Info for Sliders.

3. Purpose and Scope

3.1. Purpose

The main purpose of Jersey Android Application is to provide a user-friendly platform for sports enthusiasts to purchase sports jerseys online. The app aims to make the shopping experience hassle-free and time-effective. It also provides an opportunity for sports lovers to purchase their favourite jerseys from anywhere at any time.

3.2. Scope

The scope of the Jersey Android Application is to facilitate the purchase of sports jerseys online. It offers a range of sports team options and best-selling product lists for users to choose from. The app also provides an option to add products to the cart, view the cart, and make payments online. Additionally, it allows users to track their order and receive support via email.

3.3. Applicability

The Jersey Android Application can be used by anyone who wishes to purchase sports jerseys online. It is designed to cater to sports enthusiasts who wish to shop for sports merchandise online. The application provides an easy-to-use interface that is user-friendly and convenient. It also offers an option for users to receive support via email, making it more accessible for people who face difficulties in navigating the app.

4. Achievements

The project objectives have been successfully met by developing an Android app that enables users to purchase sports products with ease. The app features a user-friendly interface that includes a navigation bar to search products and eight sports categories from which users can select products. Each product comes with a detailed description, and users can add products to their carts and proceed to checkout. It allows users and has the freedom to email us if they have any queries on jerseysupport@gmail.com.

5. Organization of Report

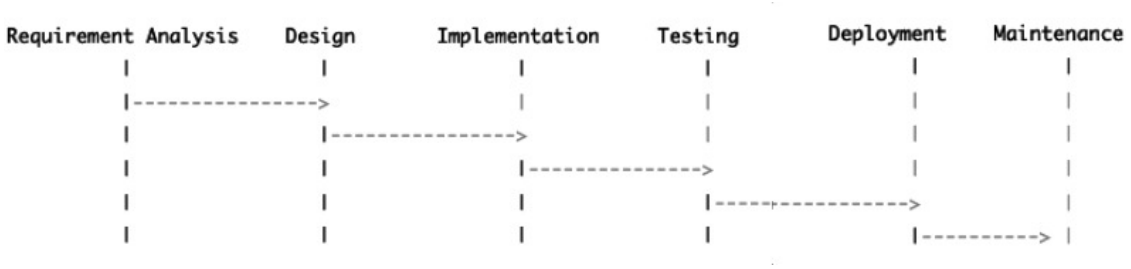


Figure 1: Waterfall Model

The report will be divided into several chapters that cover the entire software development process using **waterfall model**. The chapters include:

- **Introduction:** The introduction chapter will provide an overview of the project, including its purpose, objectives, and scope.
- **Requirement Analysis:** In this chapter, we will analyze the requirements for the Android app, including the hardware and software requirements.
- **Design:** In this chapter, we will describe the UI/UX design of the Android app, including the database schema design.
- **Technology Survey:** This chapter will compare and decide on a certain technology suitable for the Android application. The app will be developed using Android Studio, and the programming language used will be Java. For the backend db, MySQL will be used.
- **Implementation:** This chapter will include the design and programming of the modules according to the specified requirements.
- **Testing:** In this chapter, we will test all the modules and specifications thoroughly to ensure that the app is fully functional and can meet all its objectives efficiently.
  Note: Random data from google of e-commerce was used first to insert into the database after which we had updated our jerseys content on it which you can experience it on GitHub raw file.
- **Deployment:** After testing, the app will be shared through a Google Drive link where authorized users can download the APK file and install the app on their Android devices.
- **Maintenance:** This stage will involve providing ongoing support, maintenance, and updates to the app to ensure its continued functionality and relevance to the users.
- **Conclusion:** The conclusion chapter will summarize the project's achievements, including the challenges and lessons learned during the development process.

2. Survey of Technologies

Front-end Technologies

- HTML: The standard markup language used to create web pages and applications.
- CSS: Used to add visual style and layout to web pages created with HTML.
- JavaScript: A scripting language used to create interactive web pages and dynamic user interfaces.
- jQuery: A JavaScript library used to simplify HTML document traversal and manipulation, event handling, and animation.
- Android Studio: The official integrated development environment (IDE) for Android app development.

Back-end Technologies

- Java: A popular object-oriented programming language used for creating web applications and Android apps.
- MySQL: An open-source relational database management system used for storing and managing data.
- PHP: A server-side scripting language used for creating dynamic web pages and web applications.

API Testing and Debugging Tools

- Postman: A popular API testing tool used for testing and debugging APIs.
- Visual Studio Code: A lightweight, open-source code editor used for developing web and cloud applications.

Payment Gateway Integration

- Razorpay: A payment gateway integration platform that allows businesses to accept online payments.
- PayPal: A widely used payment gateway for online payments.

Web Hosting Service

- Name Cheap: A web hosting service provider that offers a range of hosting options, including shared hosting, dedicated hosting, and VPS hosting. It also provides domain registration and management services. The service uses PHPMyAdmin and MySQL for database management.

3. Requirement and analysis

3.1.Problem Definition

The problem with the existing system is that users have to physically go to sports stores to purchase sports products. There is no online platform for purchasing sports products that is easy to use and efficient.

Existing System

In the existing system, the user will have to do the following:

1) users have to physically visit sports stores to purchase sports products
2) Hard Payment Methods and Not so User-Friendly Interface
3) there is no online platform available with easy purchase modes for purchasing only jerseys, which makes it difficult for users to purchase products without visiting the stores

3.2.Requirement Specification

- **Functional Requirements**

The system will allow users to purchase sports products online through an easy-to-use interface and clear payment methods.

- **Non-Functional Requirements**

1) **Availability**

The system will be easily accessible to users.

2) **Reliability**

The system will be highly accurate, reliable, and secure.

3) **Maintainability**

The system is easily maintainable and provides an easy access to resources.

4) **Scalability**

The system will be designed in such a way that future enhancements can be easily introduced.

3.3.Planning and Scheduling

Task	Start Date	End Date	Duration
Requirement Analysis	01-Nov-22	07-Nov-22	7 days
System Design	08-Nov-22	22-Nov-22	15 days
Database Design	23-Nov-22	29-Nov-22	7 days
Front-end Development	30-Nov-22	19-Dec-22	20 days
Back-end Development	20-Dec-22	19-Jan-23	31 days
Payment Gateway Integration	20-Jan-23	02-Feb-23	14 days
Testing and Debugging	03-Feb-23	16-Feb-23	14 days
Deployment	17-Feb-23	28-Feb-23	12 days
User Acceptance Testing	01-Mar-23	08-Mar-23	8 days
Bug Fixing and Final Touches	09-Mar-23	20-Mar-23	12 days
Project Completion and Documentation	21-Mar-23	31-Mar-23	11 days

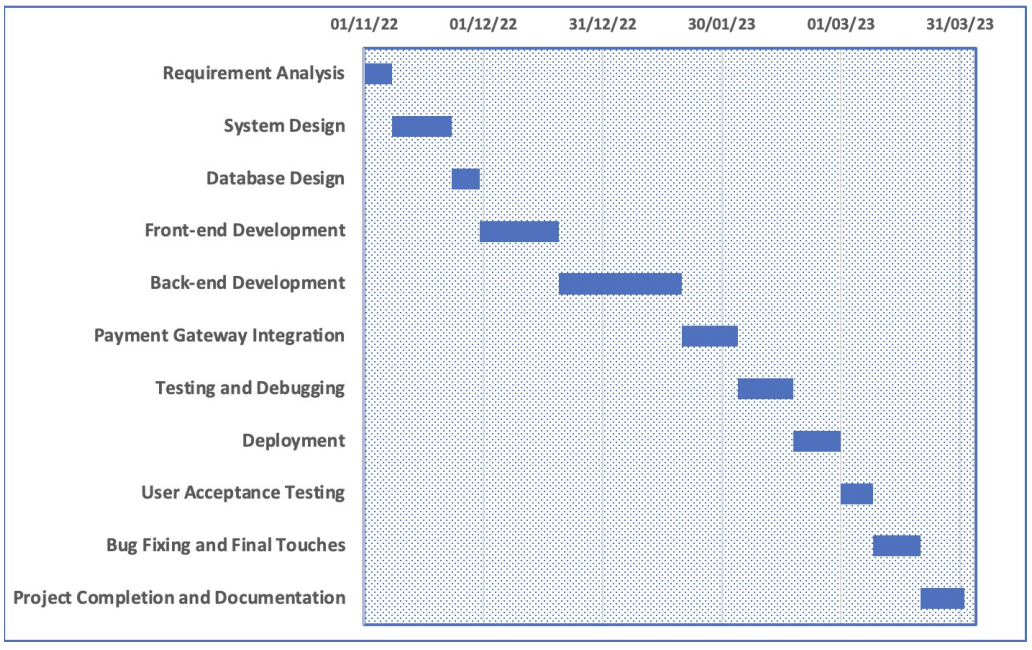


Figure 2: Gantt Chart

3.4.Software and Hardware Requirements

- **Hardware**

1) Any computer or laptop running on Windows, MacOS or Linux with at least 4 GB RAM.

2) Minimum 500 GB storage capacity.

- **Software**

1) VS Code, Android Studio

2) Microsoft SQL Server management Studio, Postman, NameCheap for server

3) Languages: HTML, JavaScript, PHP, MySql, Java, xml

3.5.Preliminary Product Description

The application will contain the following **modules:**

1. **Home Page**

   The landing page of the jersey e-commerce selling app where users can navigate to different sections of the app.

2. **Search bar**

   Allows users to search for specific jerseys by keywords, such as team name or player name.

3. **Categories**

   A section on the home page that displays different categories of jerseys, such as by team or by sport.

4. **Product page**

   Displays detailed information about a particular jersey, including images, sizes, colors, and pricing.

5. **Shopping Cart**

   A section of the app where users can view and manage the items they have added to their cart before proceeding to checkout.

6. **Check Out**

   The process of submitting an order and providing payment and shipping information.

7. **Payment**

   The section where users can choose a payment method such as Razor Pay, bank transfer and PayPal and provide payment information i.e with order code to complete their purchase.

3.6.Conceptual Models

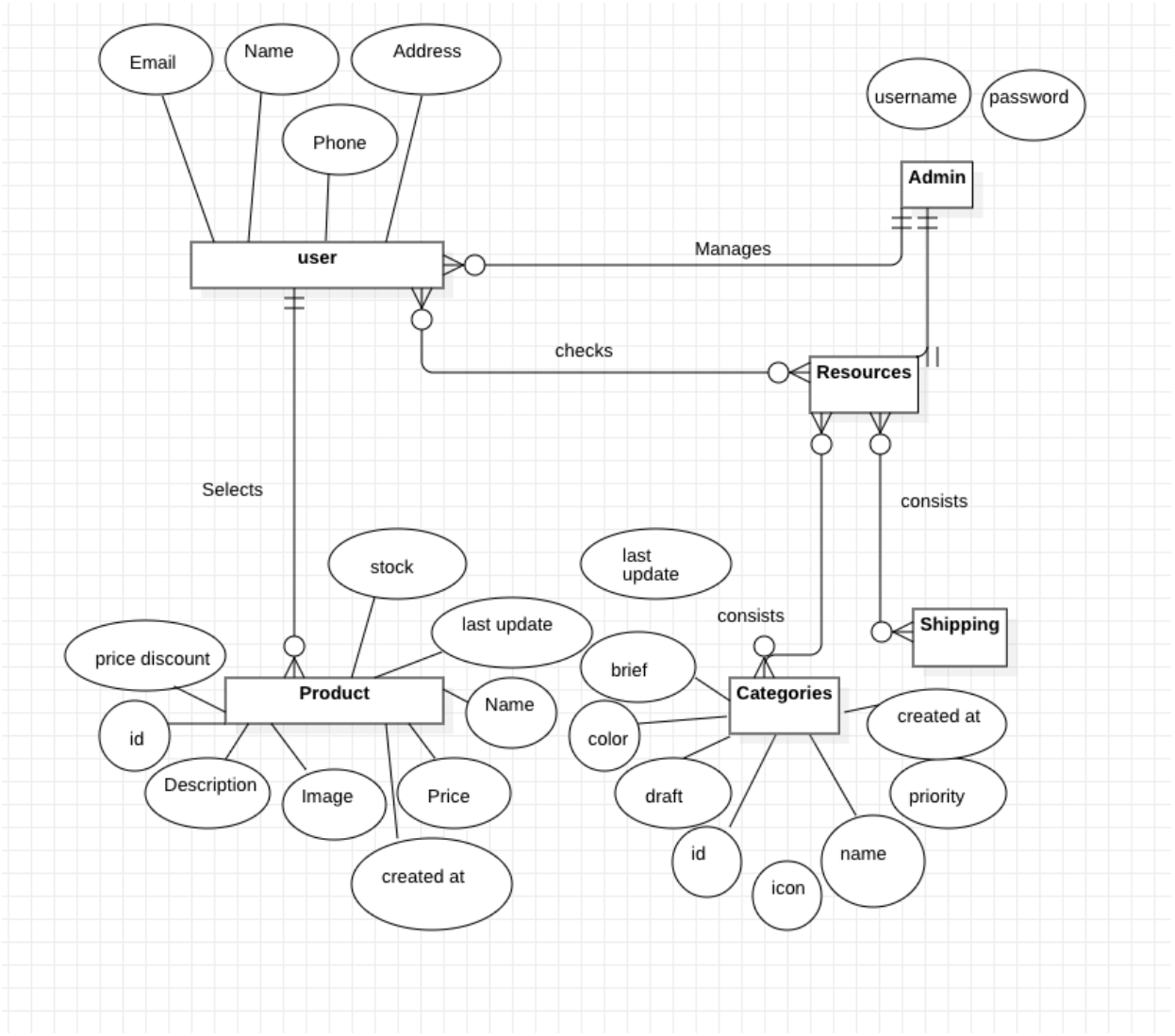


Figure 3: E-R Diagram

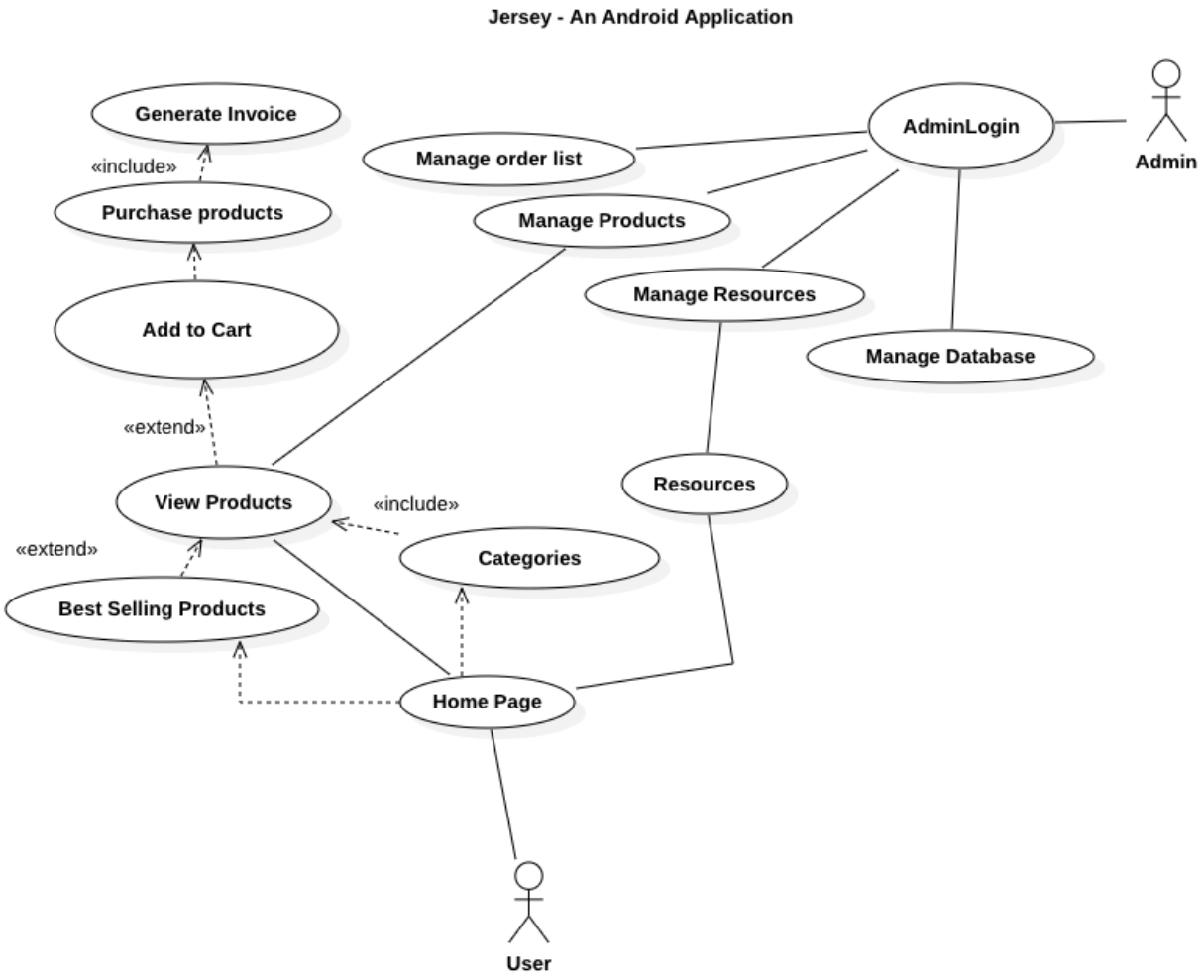


Figure 4: Use Case Diagram

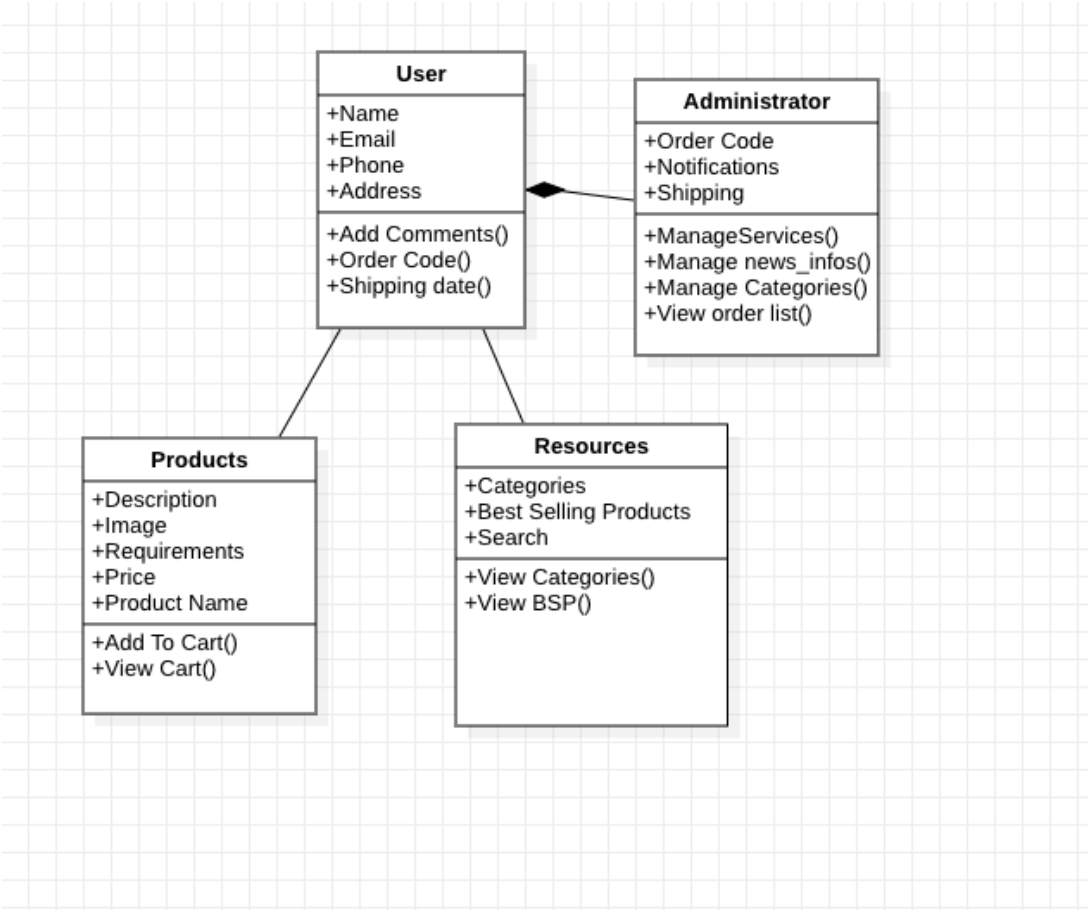


Figure 5: Class Diagram

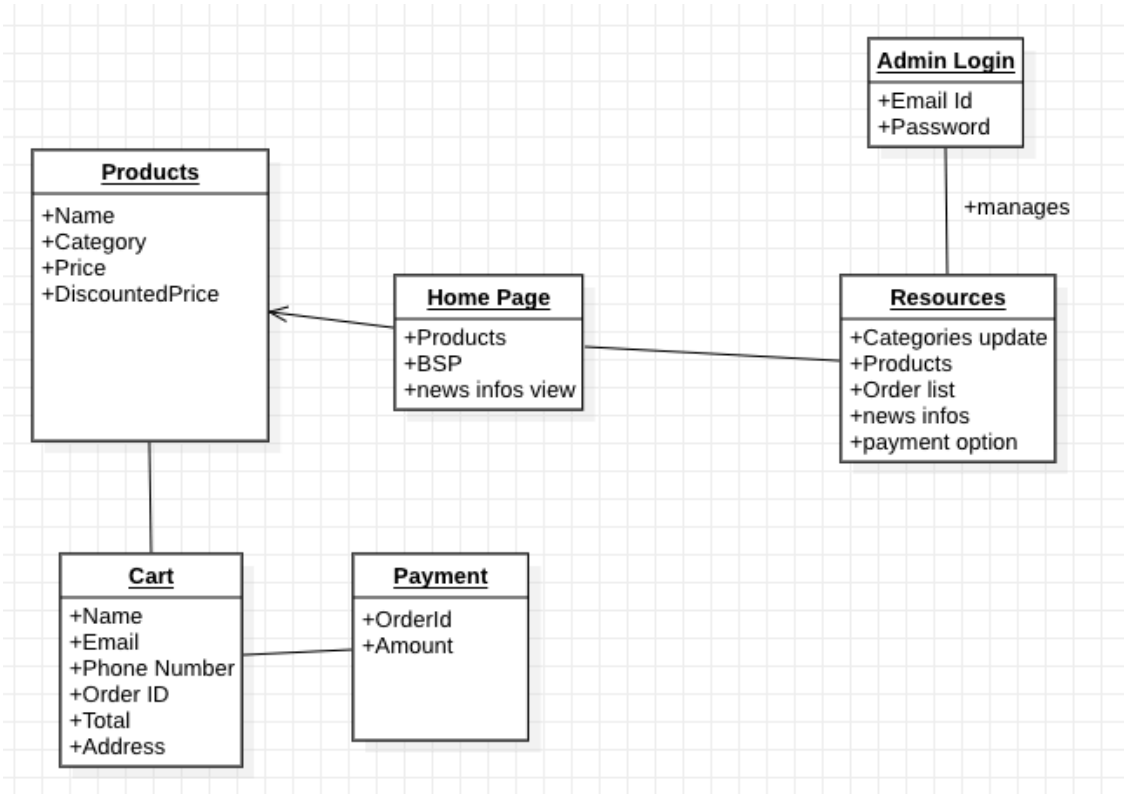


Figure 6: Object Diagram

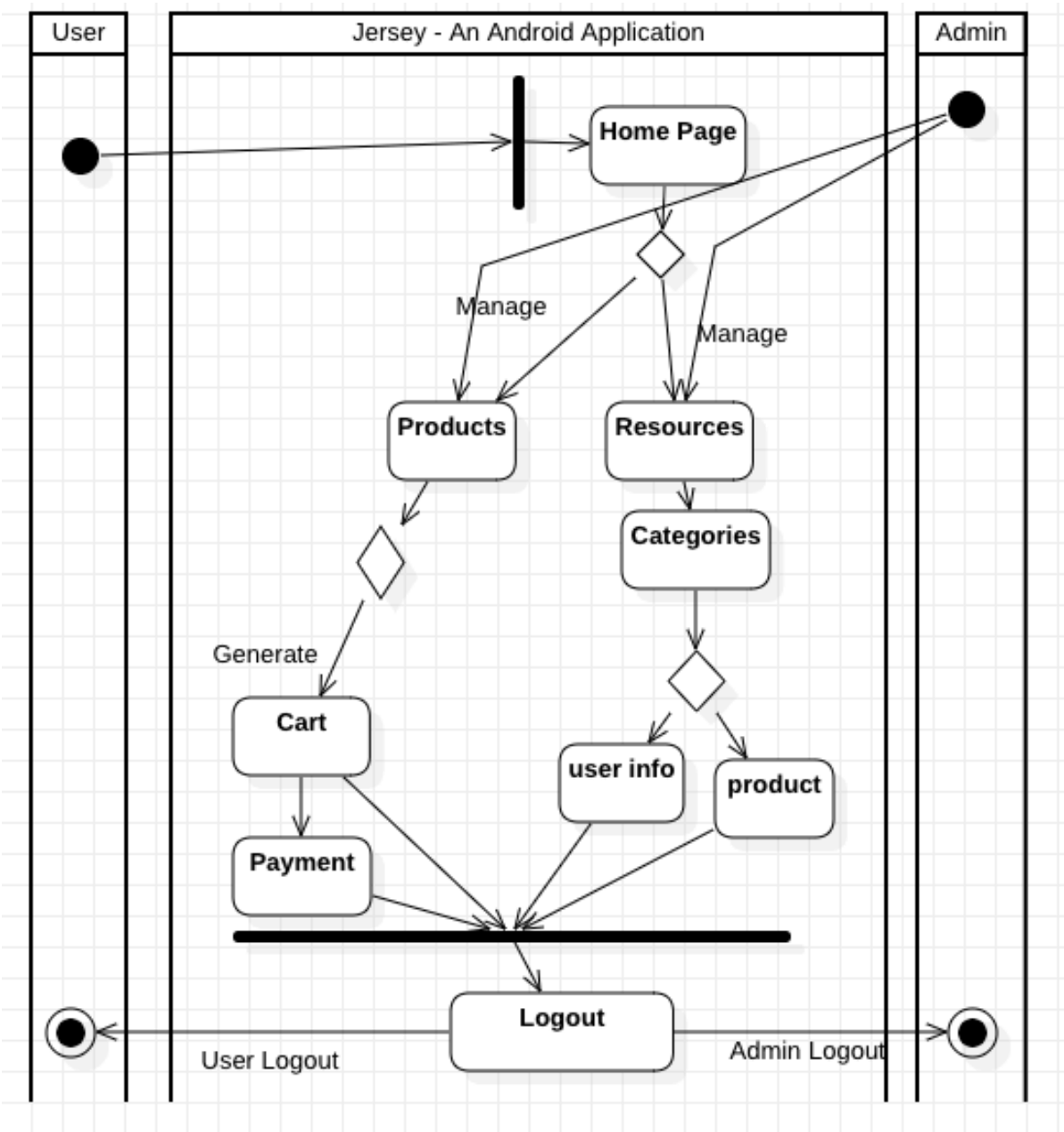


Figure 7: Activity Diagram

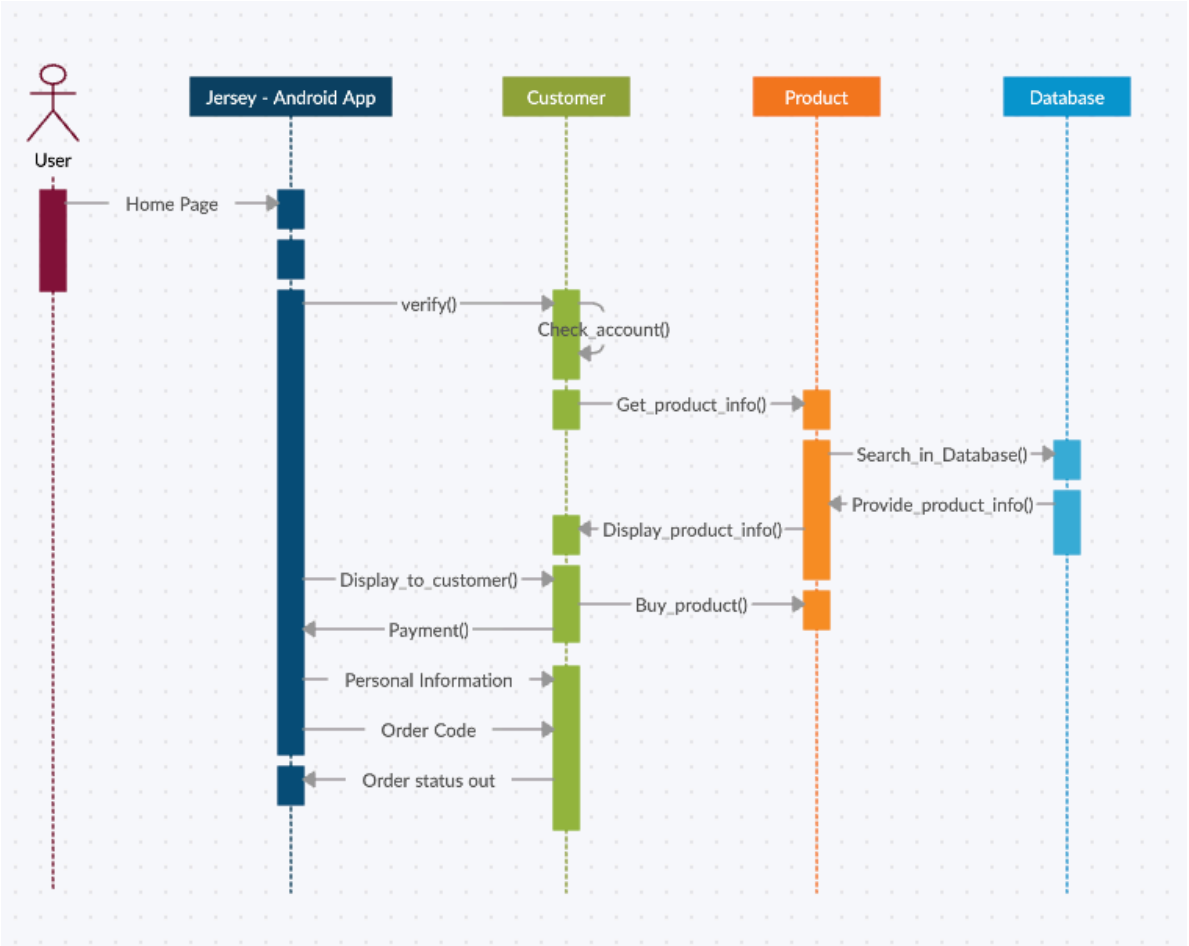


Figure 8: Sequence Diagram

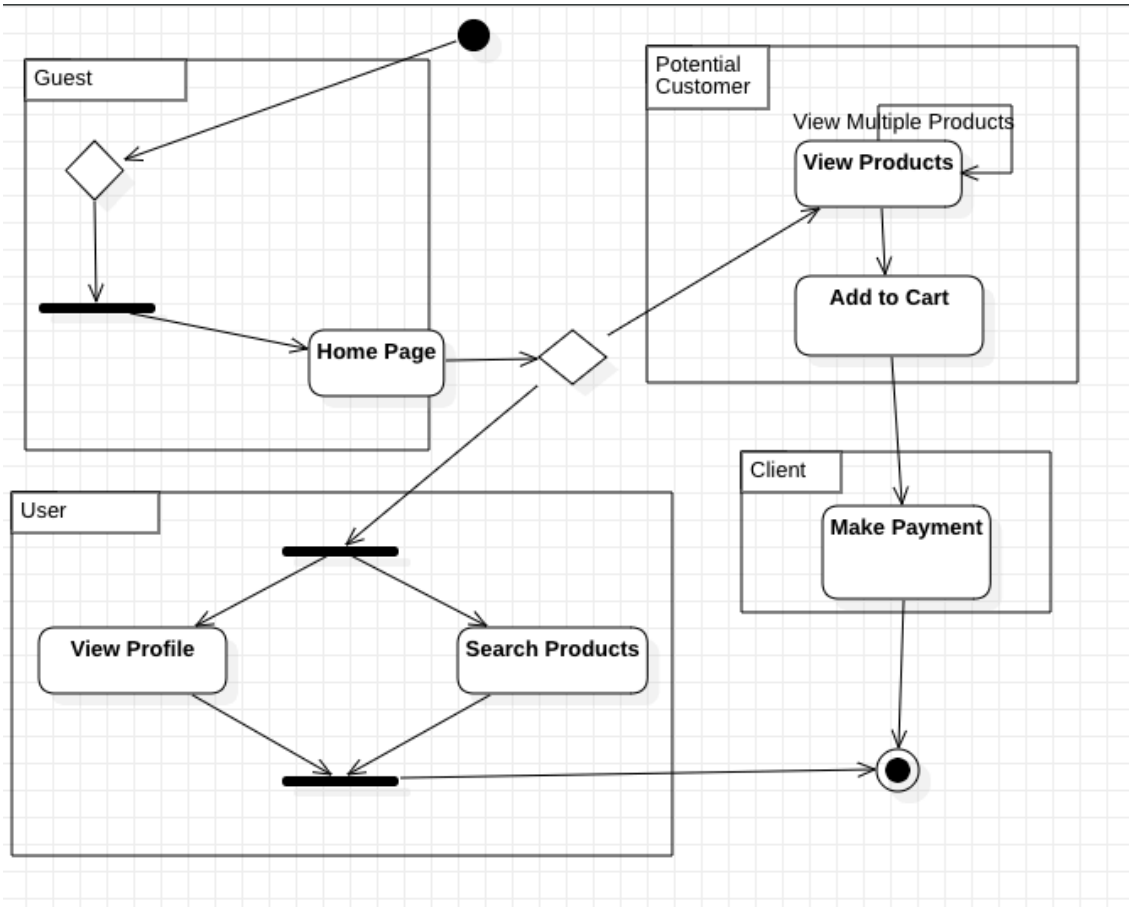


Figure 9: State Chart Diagram

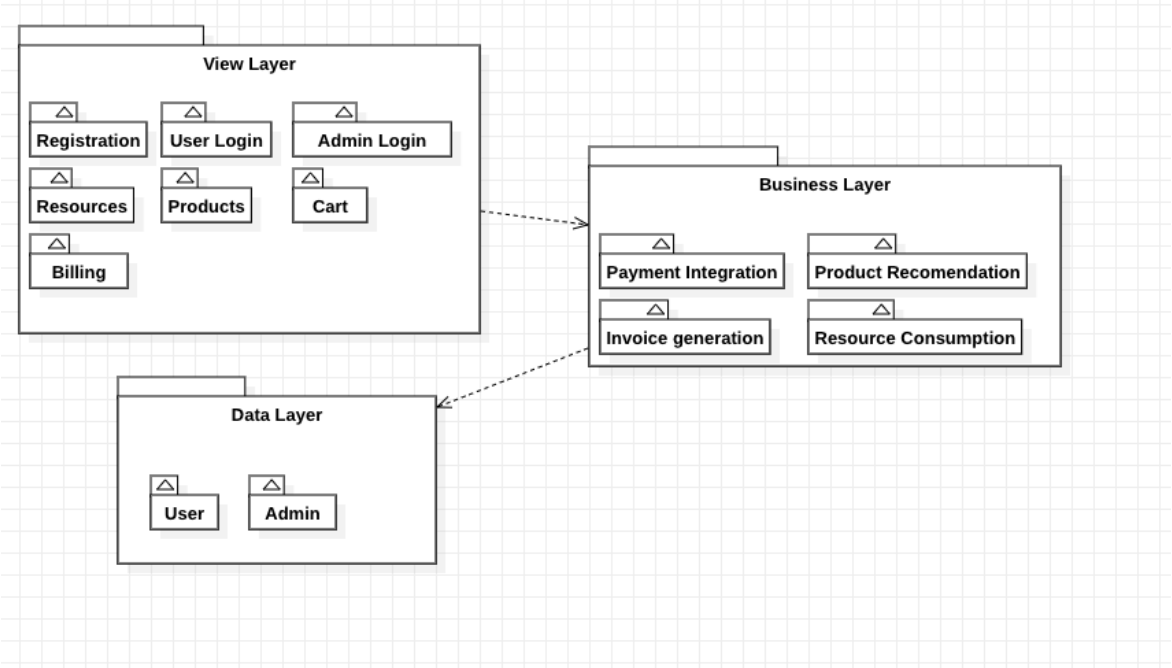


Figure 10: Package Diagram

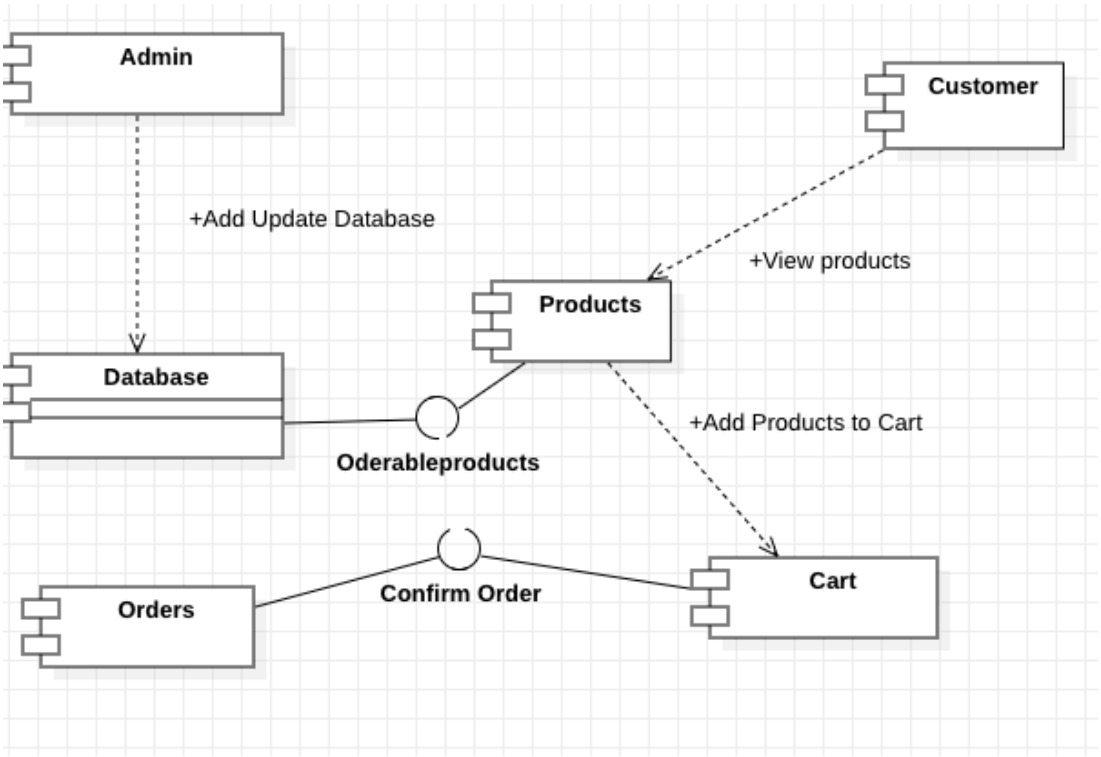


Figure 11: Component Diagram

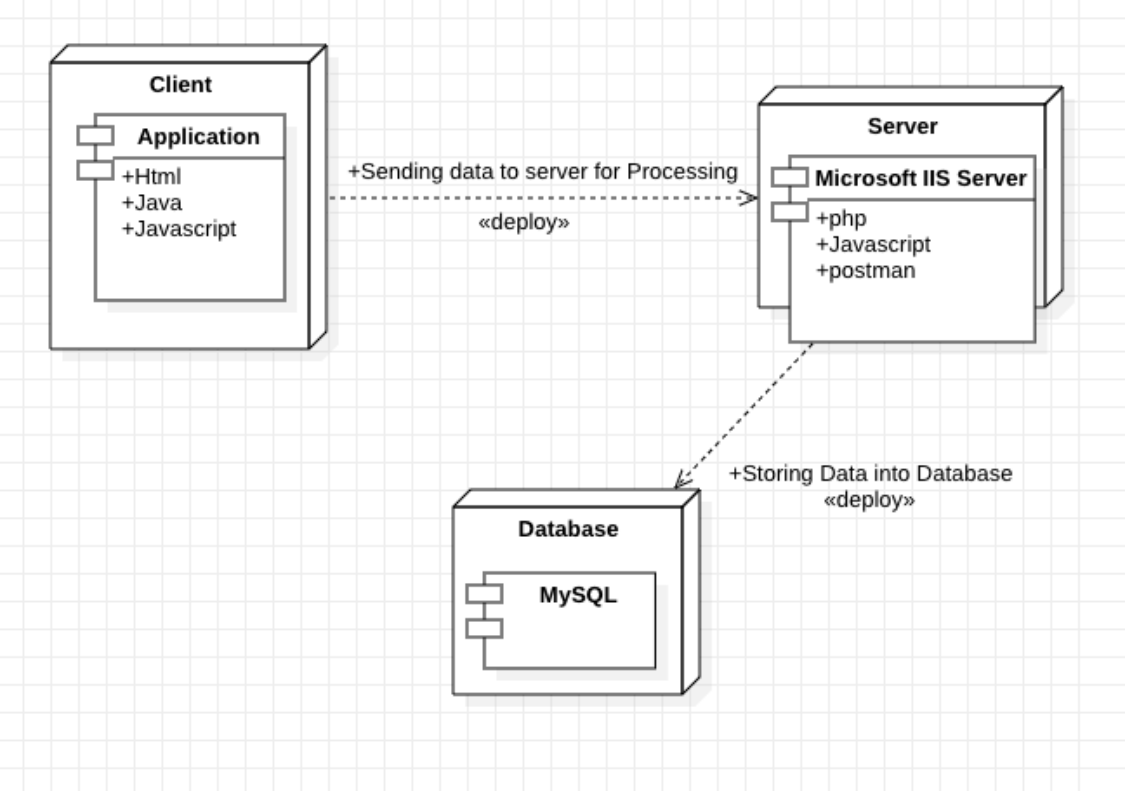


Figure 12: Deployment Diagram

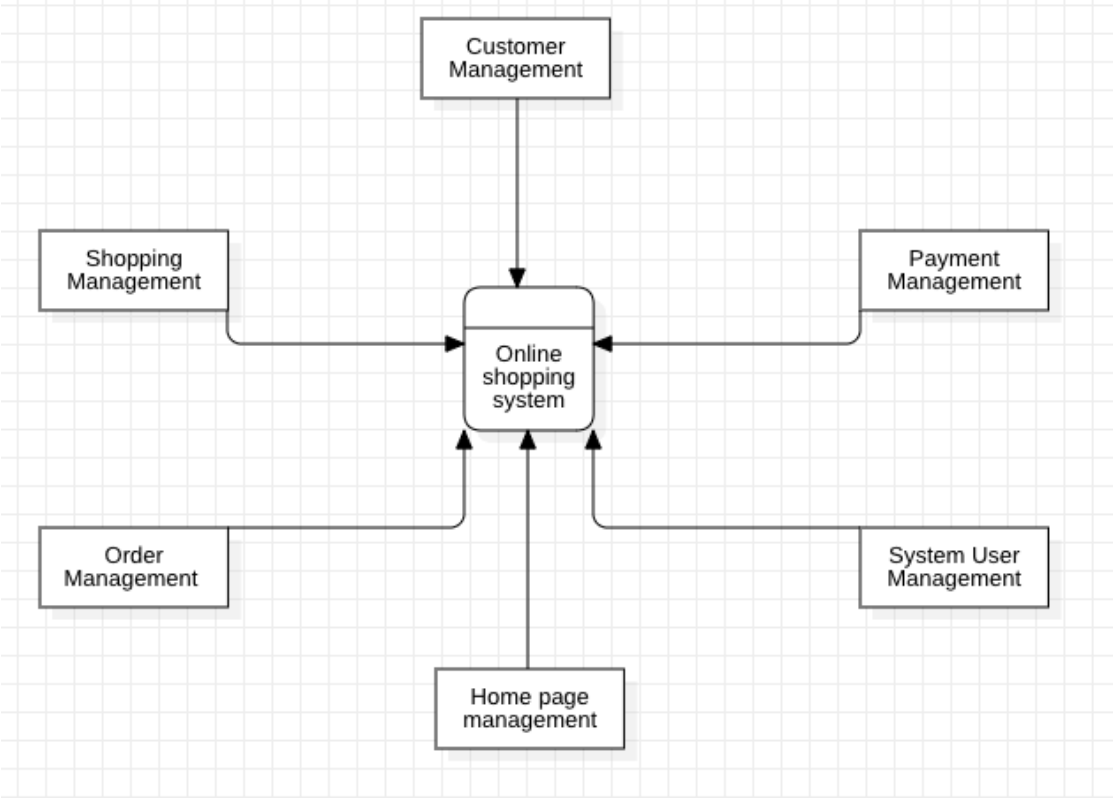


Figure 13: Data Flow Diagram (Context Level)

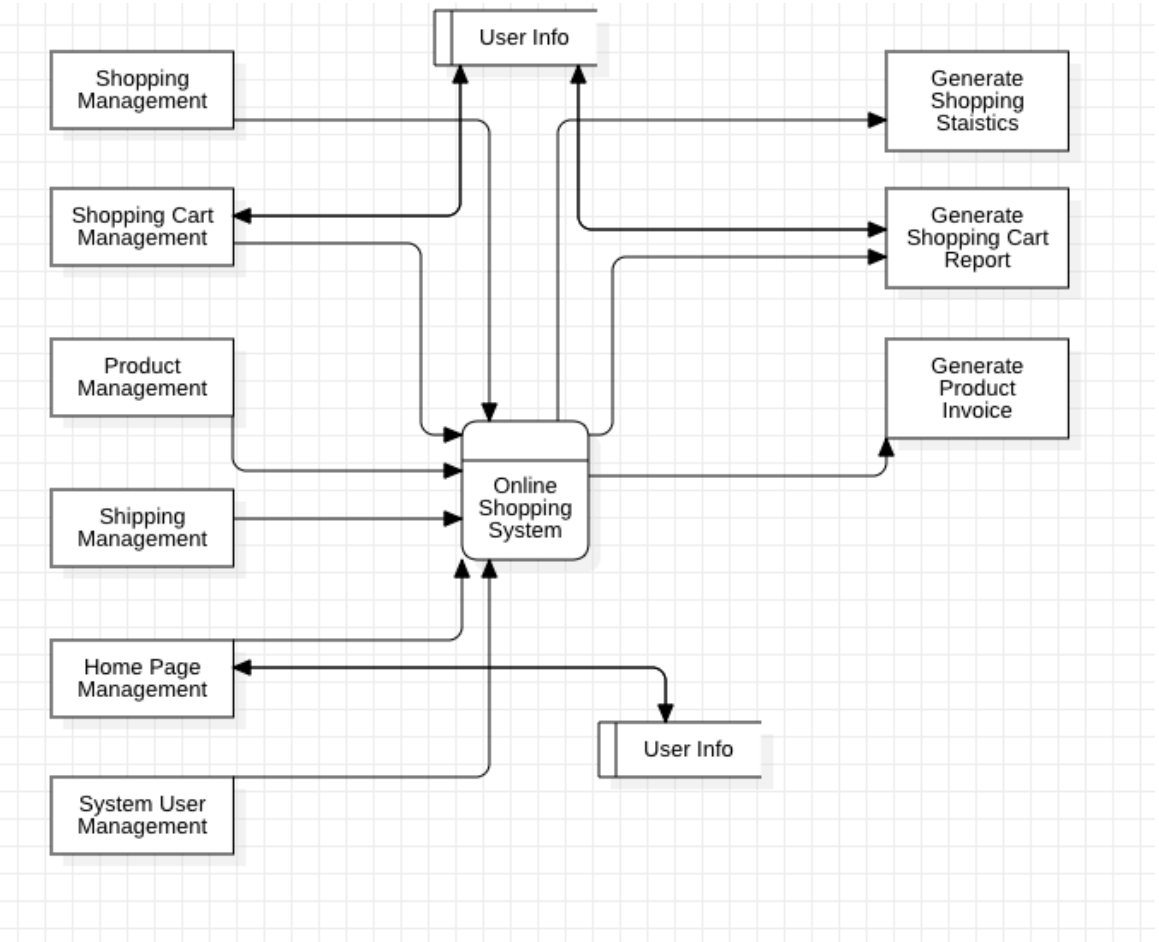


Figure 14: Data Flow Diagram (Level 1)

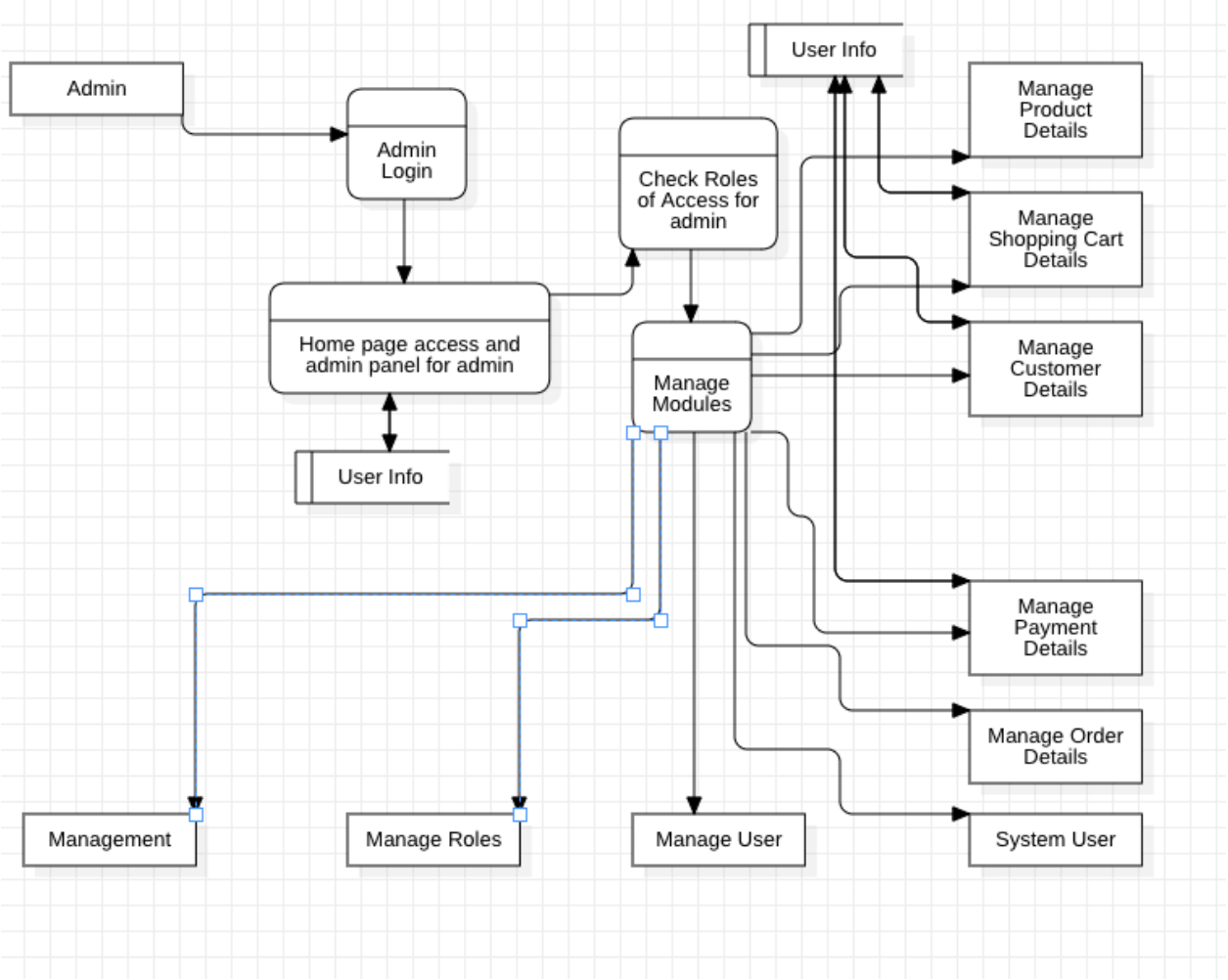


Figure 15: Data Flow Diagram (Level 2)

4)1.Basic Modules

- **Homepage:**

  The home page of the app will contain a navigation bar to search for products and sliders displaying the best-selling products. The user can select one of the 8 sports categories displayed on the home page to view the product list.

- **Product Page:**

  On selecting a particular sports category, the user will be directed to a page where they can view the products list. The products list will contain the details of all the products available under that sports category. By clicking on a particular product, the user can view the product description and add it to the cart.

- **Shopping Cart:**

  After adding a product to the cart, the user can view the product list by clicking on the cart icon. In the cart, the user can change the product quantity or remove the product from the cart. After reviewing the products in the cart, the user can proceed to checkout.

- **Checkout:**

  The checkout page will ask the user to fill in the personal information such as name, email ID, phone number, shipping address, and shipping date. The user can also add comments about the order. After filling in the personal information, the user can see the order summary with the total cost of the order. Clicking on the process checkout button will redirect the user to the payment gateway.

- **Payment:**

  The payment gateway will provide options to pay via Razor Pay, or bank transfer. The user can select one of the payment methods and proceed to make the payment. After successful payment, a dialog box will appear displaying the order ID and payment details.

4)2.Data Design

4)2.1.Schema Design

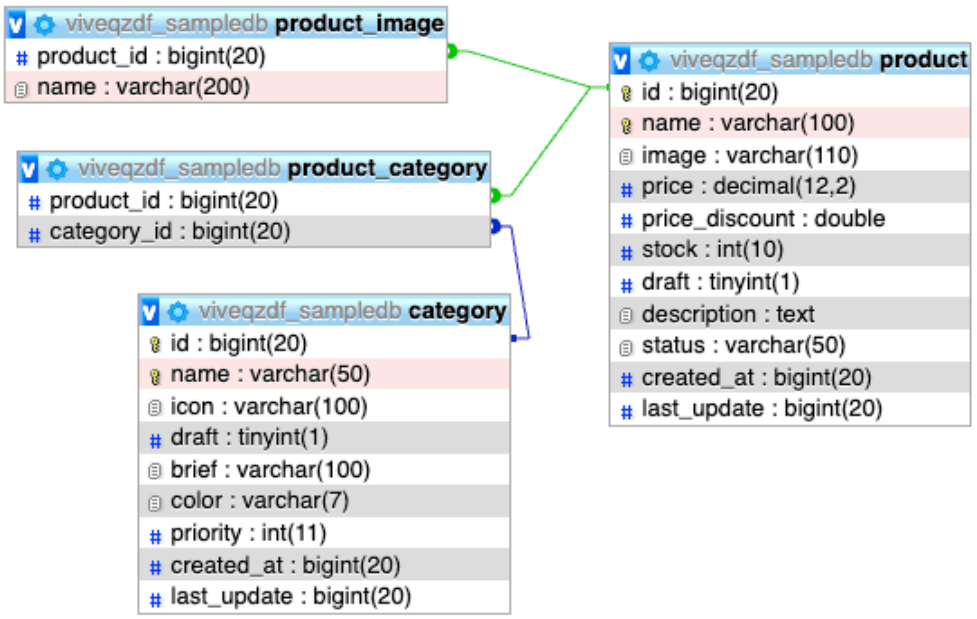


Figure 16: Database and Schema Design

id	name	username	email	password
1	Jersey	admin	chaurasiavivek0304@gmail.com	e6e061838856bf

id	title	brief_content	full_content	image	draft	status	created_at	last_update
12	Old version but vintage shot	Vintage	Vintage One	1680915328181.png	0	FEATURED	1680915328181	1680915328181
16	feature it	feature it	feature it	1680929890324.png	0	FEATURED	1680929890324	1680929890324
17	cucum	cucum	cucum	1680930021459.jpg	0	FEATURED	1680930021459	1680930021459
18	kikakika	kikakika	kikakika	1680930057437.png	0	FEATURED	1680930057437	1680930057437

id	name	image	price	price_discount	stock	draft	description	status	created_at	last_update
142	MI Jersey	1680930616360.jpg	1299.50	10	30	0	<h2 style="border: 0px solid rgb(217, 217, 227); b...	READY STOCK	1680930616360	1680930742225
143	RCB Jersey	1680930777965.jpg	1499.50	10	27	0	<h2 style="border: 0px solid rgb(217, 217, 227); b...	READY STOCK	1680930777965	1680930777965

id	name	icon	draft	brief	color	priority	created_at	last_update
16	Cricket	1680930120425.png	0	Cricket	#ffffff	1	1680930120425	1680930120425
17	Football	1680930162369.png	0	Football	#ffffff	2	1680930162369	1680930162369
18	Baseball	1680930215502.png	0	Baseball	#ffffff	3	1680930215502	1680930215502
19	Basketball	1680930249514.png	0	Basketball	#ffffff	4	1680930249514	1680930249514
20	Hockey	1680930292656.png	0	Hockey	#ffffff	5	1680930292656	1680930292656
21	Rugby	1680930336587.png	0	Rugby	#ffffff	6	1680930336587	1680930336587
22	Volleyball	1680930407633.png	0	Volleyball	#ffffff	7	1680930407633	1680930407633
23	Kabaddi	1680930455131.png	0	Kabaddi	#ffffff	8	1680930455131	1680930561967

4)2.2.Data Integrity and Constraints

- **Auto Increment Constraint:**

  Automatically increments the value of a column for each new record.

- **Primary Key Constraint:**

  Uniquely identifies each row/record in a database table.

- **Foreign Key Constraint:**

  Creates a relationship between tables based on a primary key in one table and a corresponding foreign key in another table.

- **Check Constraint:**

  Ensures that values in a column satisfy certain conditions.

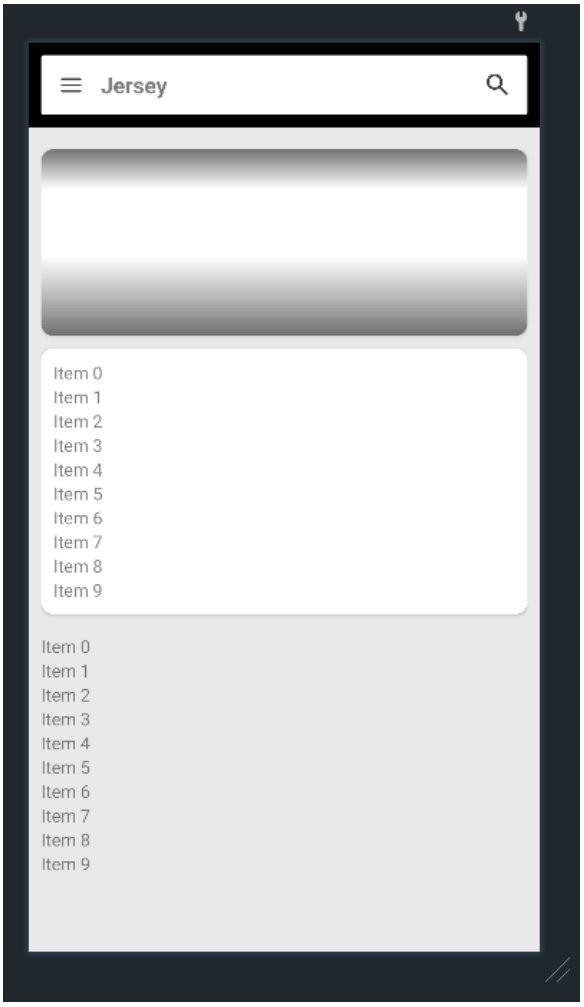
- **Not Null Constraint:**

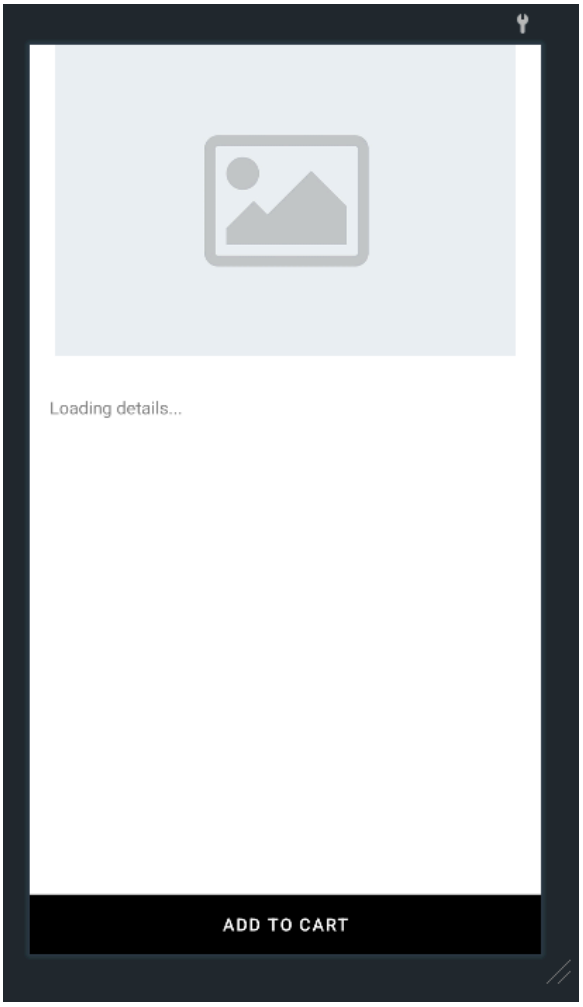
  Specifies that a column cannot contain NULL values.

4)3.User Interface Designs

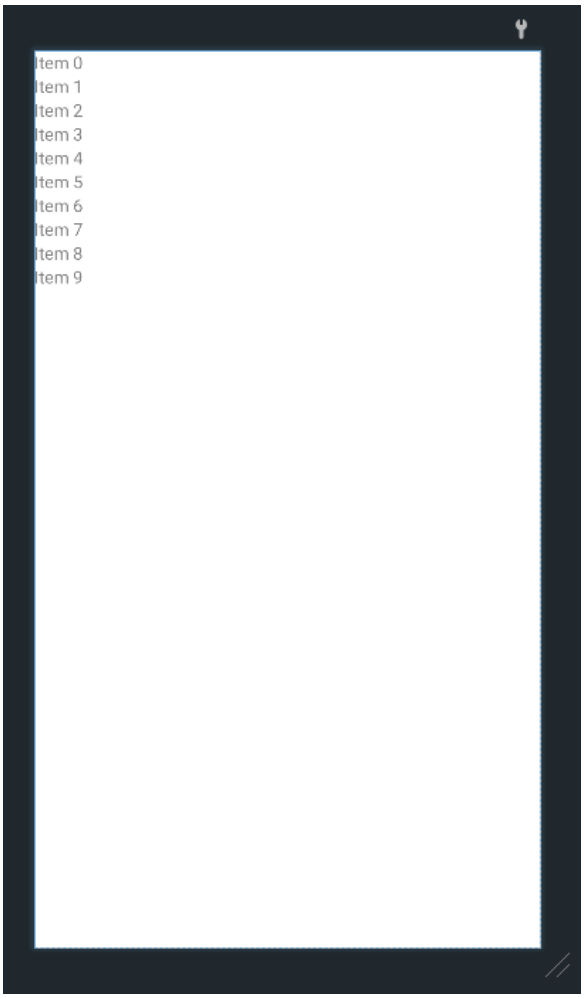
Figure 17: UI Designs

main(home page) **product detail**

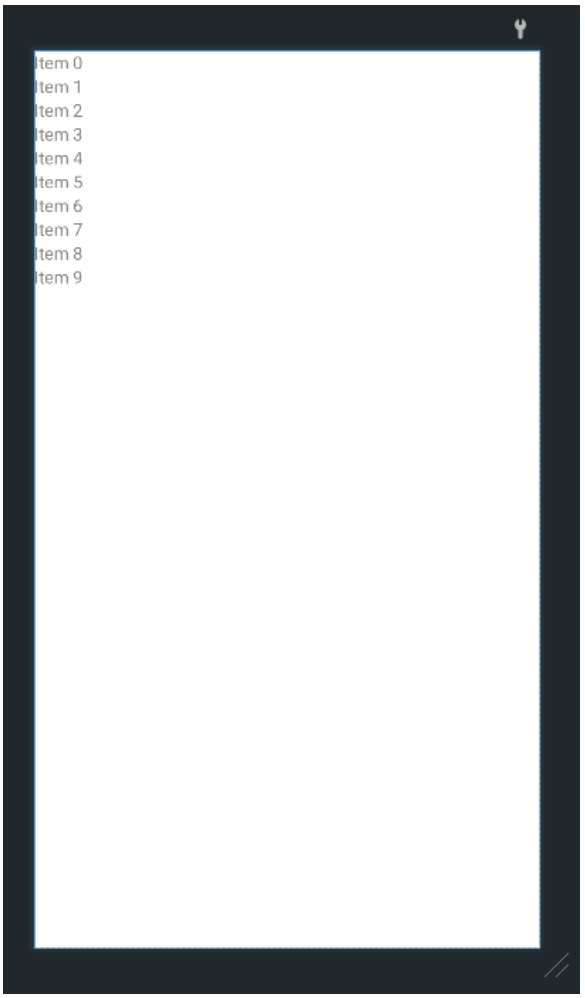




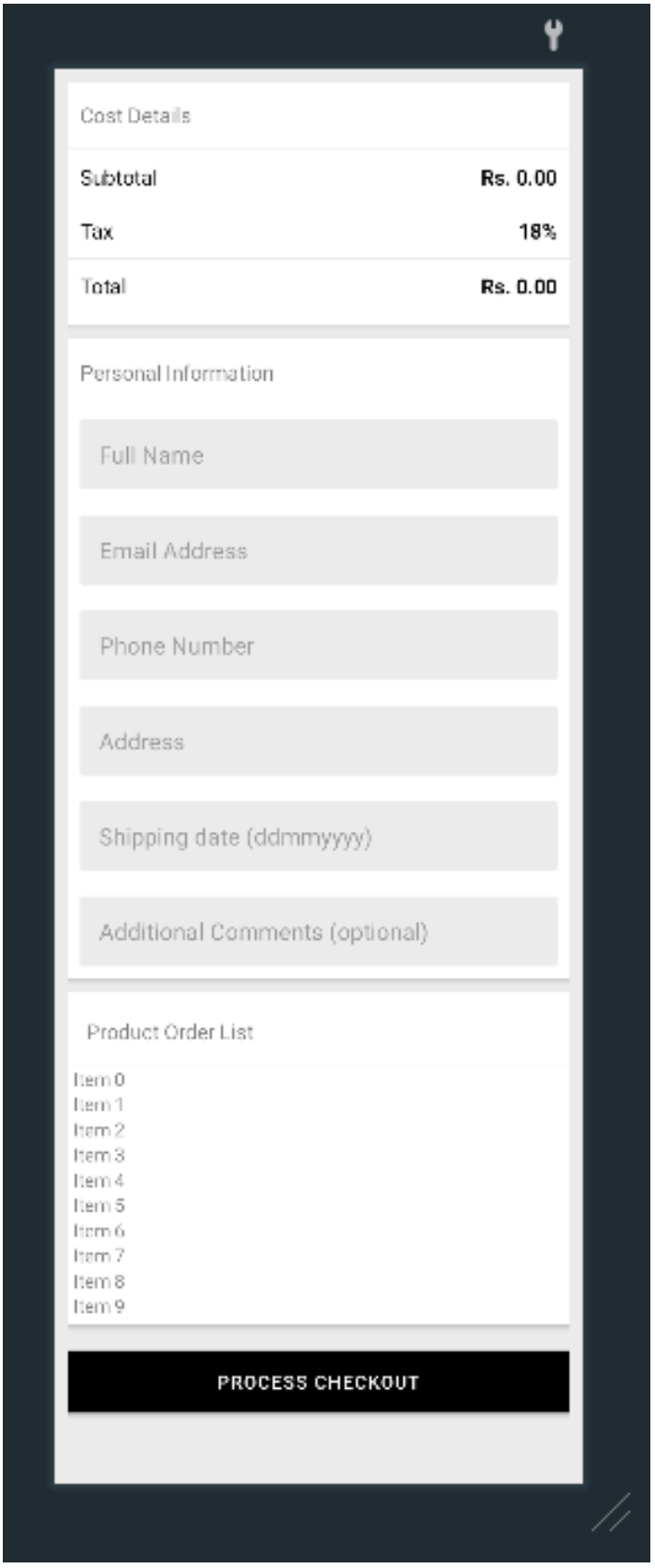
Category



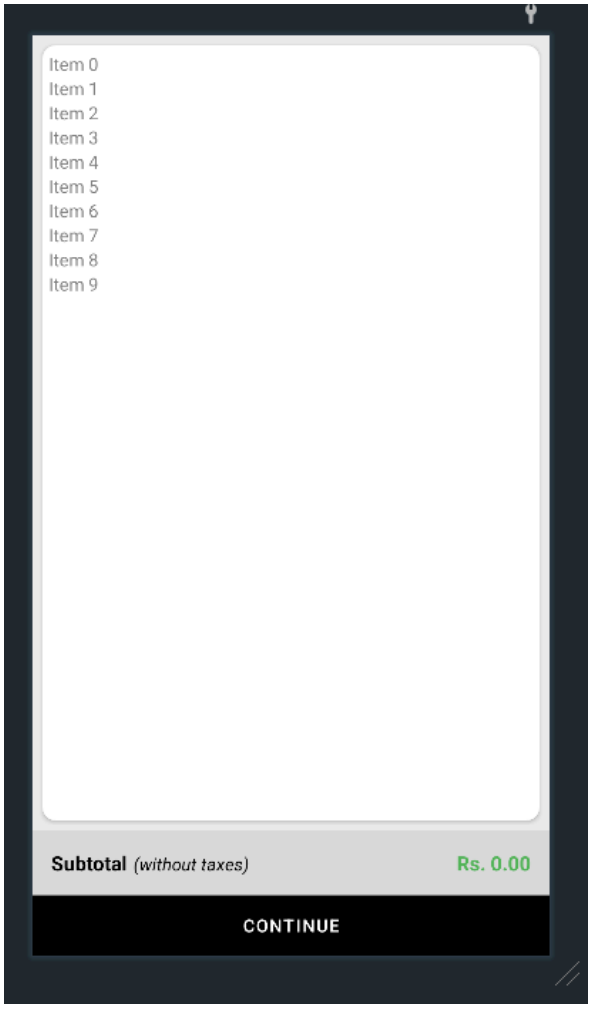
Search



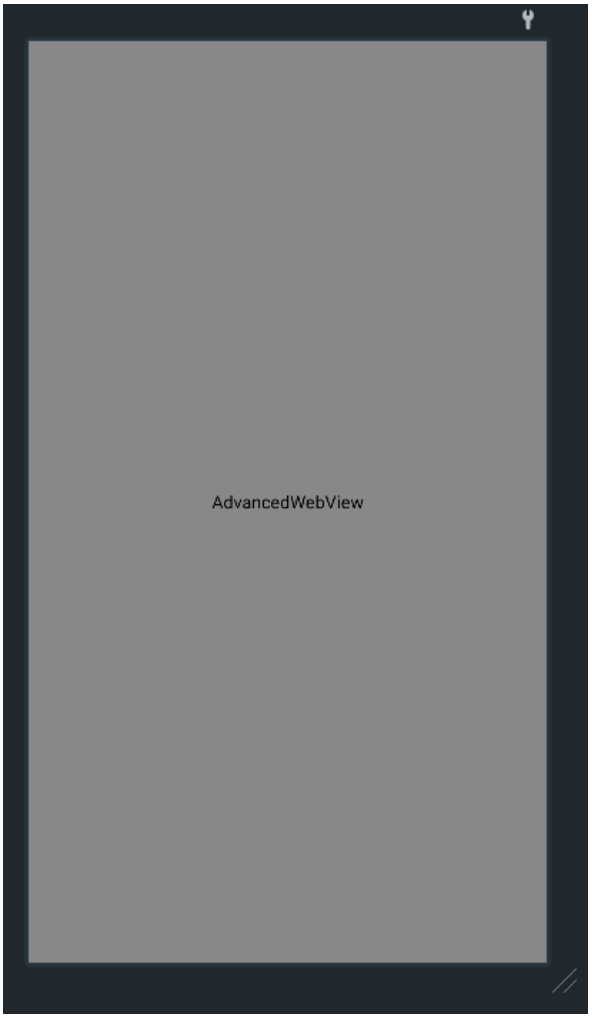
Checkout


Cost Details
Subtotal
Rs. 0.00
Tax
18%
Total
Rs. 0.00
Personal Information
Full Name
Email Address
Phone Number
Address
Shipping date (ddmmyyyy)
Additional Comments (optional)
Product Order List
Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9
PROCESS CHECKOUT

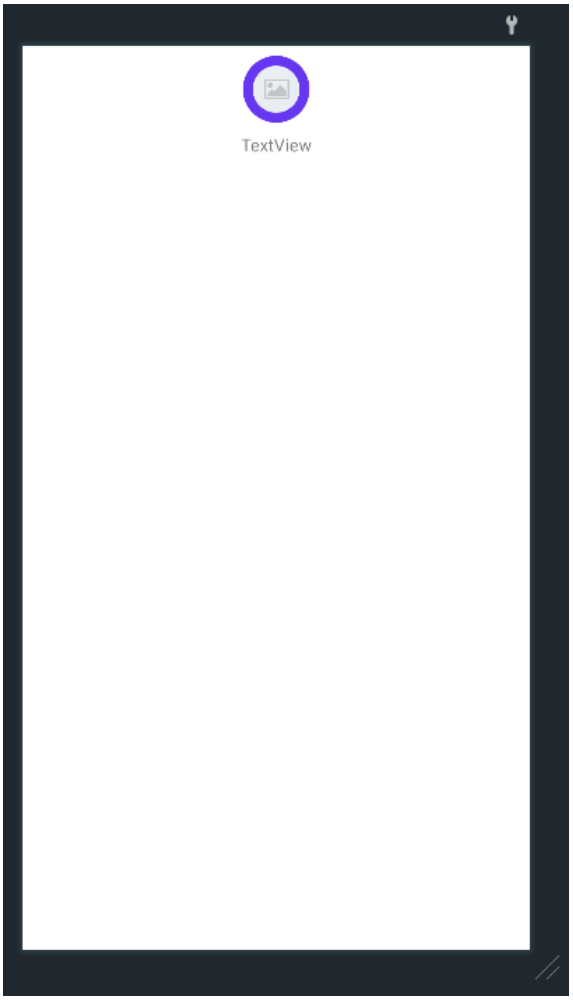
Cart



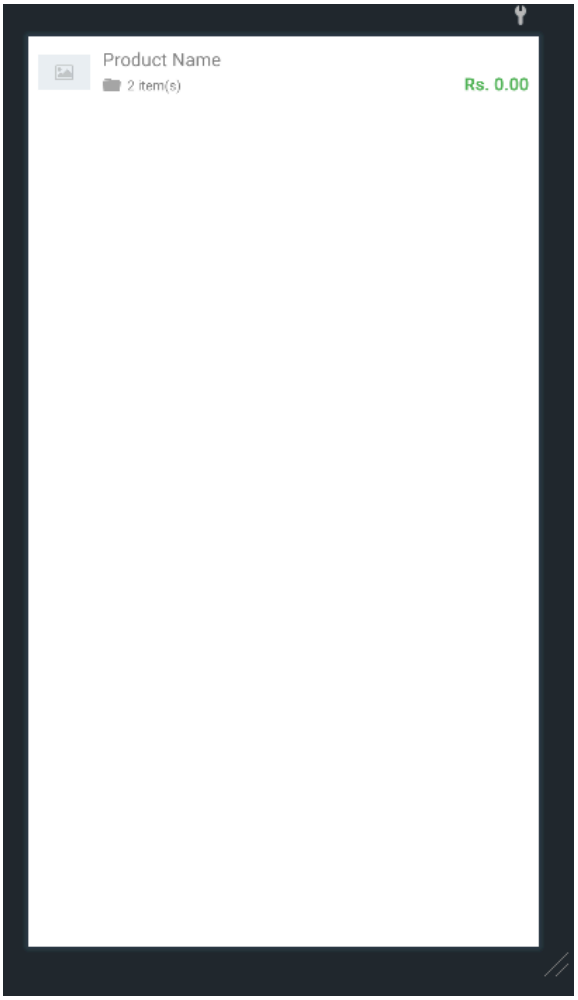
Payment



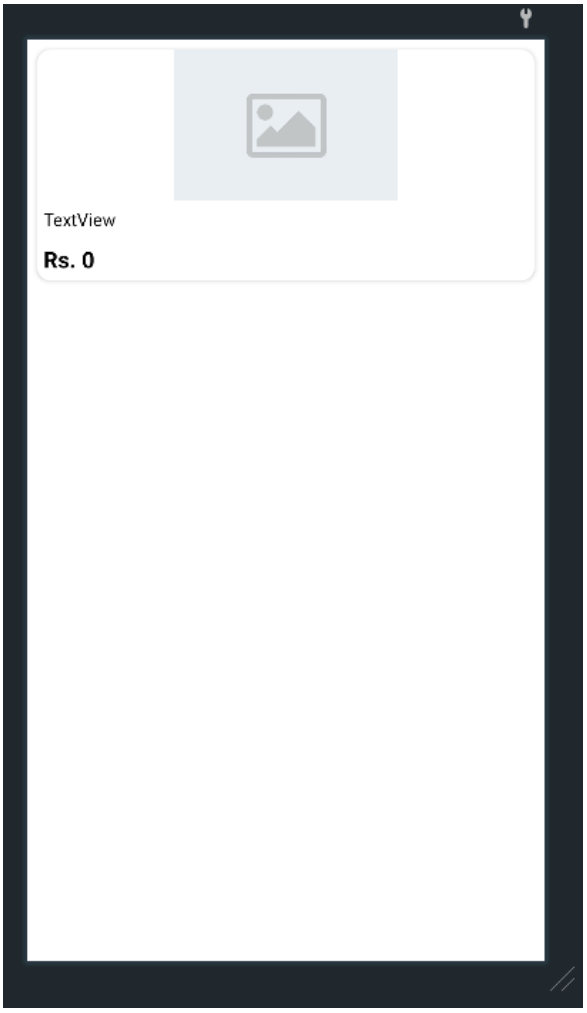
Item-categories



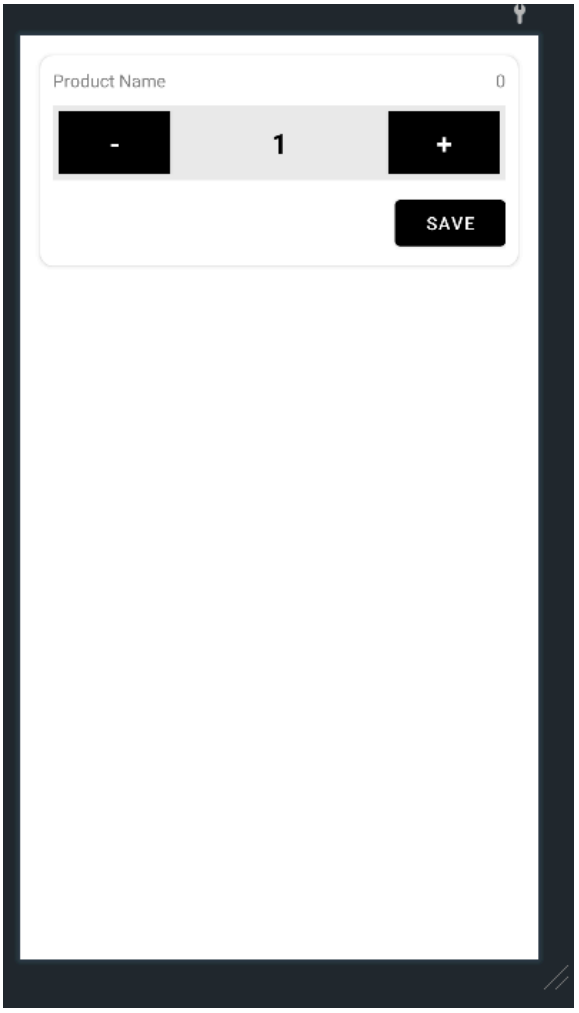
Item-cart



Item-product



Quantity-dialog



4)4. Security Issues

- **Authorization:** The system must authorize users based on their roles to prevent unauthorized access to the app's data. For example, only authenticated users with specific roles such as customer, admin, or vendor should be able to access certain parts of the application.

- **Insecure data storage:** If sensitive user data such as passwords, payment details, and personal information are not properly encrypted or stored securely, they can be vulnerable to theft or hacking.

- **Lack of secure communication:** If the app does not use secure communication protocols such as HTTPS, the user's sensitive data can be intercepted by attackers during transmission.

- **Weak authentication:** If the app uses weak authentication methods, such as simple passwords or no password at all, it can make it easier for attackers to gain access to user accounts.

- **Malware:** If the app is not properly secured, it can be vulnerable to malware attacks that can compromise user data and security.

- **Insecure third-party libraries:** If the app uses insecure third-party libraries, it can open up vulnerabilities and expose user data to attackers.

- **Insufficient app permissions:** If the app has access to more permissions than necessary, it can be easier for attackers to gain access to user data.

4)5.Test Case Design

Test Case ID	Test Case Description	Expected Result	Pass/ Fail
T001	Verify that the app launches without any errors.	The home page of the app should load without any errors.	Pass
T002	Verify that the navigation bar with search option to select particular product.	The navigation bar to search products and view that particular product should appear with select.	Pass
T003	Verify that all 8 sports options are clickable and lead to the correct page.	Clicking on any of the sports options should lead to a new page with the correct products related to that sport.	Pass
T004	Verify that the Add to Cart button adds the selected product to the cart.	Clicking on the Add to Cart button should add the selected product to the cart without any errors.	Pass
T005	Verify that the Cart page shows all the selected products.	The Cart page should display all the selected products with the correct details and quantity.	Pass
T006	Verify that the Continue button on the Cart page leads to the Checkout page.	Clicking on the Continue button should lead to the Checkout page without any errors.	Pass
T007	Verify that all the required fields on the Checkout page are compulsory.	All the required fields (name, email, phone number, address, shipping date) should be compulsory to fill out.	Pass
T008	Verify that the process checkout button leads to a payment page.	Clicking on the process checkout button should lead to a payment page without any errors.	Pass
T009	Verify that the payment page shows correct order details.	The payment page should show correct order details (order ID, name, payment options) without any errors.	Pass
T010	Verify that the admin panel has options to update the products and categories.	The admin panel should have options to add, edit and delete products and categories.	Pass

5. IMPLEMENTATION AND TESTING

5.1.Implementation Approach

• Introduction

The Android app starts with the home page where the user can see the navigation bar to search and below that, there are sliders displaying the best selling products. Below that, there are 8 options to select sports. After selecting any sport, a new page appears where the user can see the list of products and select whichever they want. Then a page appears with the product description and an "Add to Cart" button. By clicking on the "Add to Cart" button, the user can add the product to their cart. At the top left corner, the user can see an option to view their cart, where they can see all the selected products.

After selecting the products, the user can click on the "Continue" button and see the total cost details. The user needs to fill in the compulsory personal information such as name, email ID, phone number, address, shipping date, and optional comments. After filling in the details, the user can click on the "Process Checkout" button.

A dialog box will appear after clicking on the "Process Checkout" button for a successful order with a "Pay Now" option. Clicking on the "Pay Now" option will take the user to a payment page where they can see their order ID/code, their name, and options to make payment using Razor Pay, or bank transfer.
Note: For testing, we have used razor pay test case api we can replace it with the actual API keys in future.

• Input and Output Design Implementation

For this Android app project, the implementation approach involved creating different modules for specific functionalities and designing a database to store the necessary information. The user interface includes a navigation bar for searching and options to select from eight different sports. Once a sport is selected, the user can view the list of products available and add them to their cart. At the checkout page, the user is required to input personal information such as name, email, phone number, address, shipping date, and optional comments.

The app integrates with PHPMyAdmin and shared web hosting at the backend on NameCheap, allowing for easy updates to product and category options, news info for sliders, and payment options.

• Code Module

This Android app was created by using Java for the front-end and SQLite for the back-end. The app is linked with phpMyAdmin and web host at the backend to update the options.

• System Implementation

The Android app is designed in such a way that the user can easily navigate through the different sections. The app has 8 sports options on the home page, and after selecting any sports option, the user can see the list of products related to that sport. By clicking on the "Add to Cart" button, the user can add the product to their cart, and they can see the selected products in the cart.

The user can go back to the home page by clicking on the top right corner and selecting any other sports option or by adding more products. The user can increase or decrease the quantity of any individual product and see the available stock as well.

The admin can update all the content from the back end which is built on PhpMyAdmin and the database is on mysql and these are hosted on NameCheap hosting. First we have bought a domain and then connected it to our NameCheap server and then access the CPanel where we have all our files of PhpMyAdmin and created database on MySql. In the "Product" option, the admin can add a product and select which of the 8 categories they want to keep and can add more. In the "Category" option, the admin can edit the category for the 8 categories of sports options and can add more categories. In the "News Info" option, the admin can update the sliders, and in the "Payment Option" option, the admin can keep the payment options. The app also has an "Order List" option, where the admin can see the list of orders made on the app.

• Project Summary

The Android Jersey app is specifically designed to facilitate users in purchasing sports products through their mobile devices. As soon as the app is launched, users can utilize the navigation bar to search for specific products and get redirected to the desired product page. Additionally, there are eight sports options to choose from below the sliders. After selecting a sport, users will be taken to a page where they can view products related to that sport and select the items they want to purchase. After selecting a product, a page with its description will appear, and users can add it to their cart by clicking on the "Add to Cart" button.

By clicking on the cart icon, users can view their selected products and proceed to checkout. The checkout page requires users to fill in personal information such as name, email address, phone number, shipping address, and comments (optional). Users can also adjust the quantity of items and see the total cost. After filling in the information, users can click on "Process Checkout" and choose a payment option such as PayPal, Razor Pay, or bank transfer.

The app also includes options to update sports teams and products, view a list of orders made and get confirmation via email within next 24 hours. It is linked via namecheap's shared hosting which includes phpmyadmin & mysql for database, allowing for easy backend management. Overall, the app provides a convenient and user-friendly way for sports enthusiasts to purchase products on their mobile devices.

5.2.Code Details and Code Efficiency

Code:

Please find below the list of the main Java files for this project. For the complete set of project files and their contents, please visit the provided link and download the project's ZIP file:

https://github.com/Vivek-chaurasia-03/jersey.git

MainActivity.java

```
package com.example.aexpress.activities;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.aexpress.R;
import com.example.aexpress.adapters.CategoryAdapter;
import com.example.aexpress.adapters.ProductAdapter;
import com.example.aexpress.databinding.ActivityMainBinding;
import com.example.aexpress.model.Category;
import com.example.aexpress.model.Product;
import com.example.aexpress.utils.Constants;
import com.mancj.materialsearchbar.MaterialSearchBar;

import org.imaginativeworld.whynotimagecarousel.model.CarouselItem;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    ActivityMainBinding binding;
    CategoryAdapter categoryAdapter;
    ArrayList<Category> categories;

    ProductAdapter productAdapter;
    ArrayList<Product> products;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());


        binding.searchBar.setOnSearchActionListener(new MaterialSearchBar.OnSearchActionListener() {
            @Override
            public void onSearchStateChanged(boolean enabled) {

            }

            @Override
            public void onSearchConfirmed(CharSequence text) {
                Intent intent = new Intent(MainActivity.this, SearchActivity.class);
                intent.putExtra("query", text.toString());
                startActivity(intent);
            }

            @Override
            public void onButtonClicked(int buttonCode) {

            }
```

```
    });

    initCategories();
    initProducts();
    initSlider();
}

private void initSlider() {
    getRecentOffers();
}

void initCategories() {
    categories = new ArrayList<>();
    categoryAdapter = new CategoryAdapter(this, categories);

    getCategories();

    GridLayoutManager layoutManager = new GridLayoutManager(this, 4);
    binding.categoriesList.setLayoutManager(layoutManager);
    binding.categoriesList.setAdapter(categoryAdapter);
}

void getCategories() {
    RequestQueue queue = Volley.newRequestQueue(this);

    StringRequest request = new StringRequest(Request.Method.GET, Constants.GET_CATEGORIES_URL, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                Log.e("err", response);
                JSONObject mainObj = new JSONObject(response);
                if(mainObj.getString("status").equals("success")) {
                    JSONArray categoriesArray = mainObj.getJSONArray("categories");
                    for(int i =0; i< categoriesArray.length(); i++) {
                        JSONObject object = categoriesArray.getJSONObject(i);
                        Category category = new Category(
                                object.getString("name"),
                                Constants.CATEGORIES_IMAGE_URL + object.getString("icon"),
                                object.getString("color"),
                                object.getString("brief"),
                                object.getInt("id")
                        );
                        categories.add(category);
                    }
                    categoryAdapter.notifyDataSetChanged();
                } else {
                    // DO nothing
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {

        }
    });

    queue.add(request);
}

void getRecentProducts() {
    RequestQueue queue = Volley.newRequestQueue(this);

    String url = Constants.GET_PRODUCTS_URL + "?count=8";
    StringRequest request = new StringRequest(Request.Method.GET, url, response -> {
        try {
            JSONObject object = new JSONObject(response);
            if(object.getString("status").equals("success")){
                JSONArray productsArray = object.getJSONArray("products");
                for(int i =0; i< productsArray.length(); i++) {
                    JSONObject childObj = productsArray.getJSONObject(i);
                    Product product = new Product(
                            childObj.getString("name"),
                            Constants.PRODUCTS_IMAGE_URL + childObj.getString("image"),
                            childObj.getString("status"),
                            childObj.getDouble("price"),
                            childObj.getDouble("price_discount"),
                            childObj.getInt("stock"),
                            childObj.getInt("id")

                    );
                    products.add(product);
```

```
                }
                productAdapter.notifyDataSetChanged();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }, error -> { });

    queue.add(request);
}

void getRecentOffers() {
    RequestQueue queue = Volley.newRequestQueue(this);

    StringRequest request = new StringRequest(Request.Method.GET, Constants.GET_OFFERS_URL, response -> {
        try {
            JSONObject object = new JSONObject(response);
            if(object.getString("status").equals("success")) {
                JSONArray offerArray = object.getJSONArray("news_infos");
                for(int i =0; i < offerArray.length(); i++) {
                    JSONObject childObj =  offerArray.getJSONObject(i);
                    binding.carousel.addData(
                            new CarouselItem(
                                    Constants.NEWS_IMAGE_URL + childObj.getString("image"),
                                    childObj.getString("title")
                            )
                    );
                }
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }, error -> {});
    queue.add(request);
}

void initProducts() {
    products = new ArrayList<>();
    productAdapter = new ProductAdapter(this, products);

    getRecentProducts();

    GridLayoutManager layoutManager = new GridLayoutManager(this, 2);
    binding.productList.setLayoutManager(layoutManager);
    binding.productList.setAdapter(productAdapter);
}

}
```

ProductDetailActivity.java

```
package com.example.aexpress.activities;

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

import android.content.Intent;
import android.os.Bundle;
import android.text.Html;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Toast;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.bumptech.glide.Glide;
import com.example.aexpress.R;
import com.example.aexpress.databinding.ActivityProductDetailBinding;
import com.example.aexpress.model.Product;
import com.example.aexpress.utils.Constants;
import com.hishd.tinycart.model.Cart;
import com.hishd.tinycart.util.TinyCartHelper;

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

public class ProductDetailActivity extends AppCompatActivity {
```

```
ActivityProductDetailBinding binding;
Product currentProduct;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityProductDetailBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    String name = getIntent().getStringExtra("name");
    String image = getIntent().getStringExtra("image");
    int id = getIntent().getIntExtra("id",0);
    double price = getIntent().getDoubleExtra("price",0);

    Glide.with(this)
            .load(image)
            .into(binding.productImage);

    getProductDetails(id);

    getSupportActionBar().setTitle(name);

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

    Cart cart = TinyCartHelper.getCart();


    binding.addToCartBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            cart.addItem(currentProduct,1);
            binding.addToCartBtn.setEnabled(false);
            binding.addToCartBtn.setText("Added in cart");
        }
    });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.cart, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if(item.getItemId() == R.id.cart) {
        startActivity(new Intent(this, CartActivity.class));
    }
    return super.onOptionsItemSelected(item);
}

void getProductDetails(int id) {
    RequestQueue queue = Volley.newRequestQueue(this);

    String url = Constants.GET_PRODUCT_DETAILS_URL + id;

    StringRequest request = new StringRequest(Request.Method.GET, url, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject object = new JSONObject(response);
                if(object.getString("status").equals("success")) {
                    JSONObject product = object.getJSONObject("product");
                    String description = product.getString("description");
                    binding.productDescription.setText(
                            Html.fromHtml(description)
                    );

                    currentProduct = new Product(
                            product.getString("name"),
                            Constants.PRODUCTS_IMAGE_URL + product.getString("image"),
                            product.getString("status"),
                            product.getDouble("price"),
                            product.getDouble("price_discount"),
                            product.getInt("stock"),
                            product.getInt("id")
                    );
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
```

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

            }
        });

        queue.add(request);
    }

    @Override
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }
}
```

CategoryActivity.java

```
package com.example.aexpress.activities;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;

import android.os.Bundle;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.aexpress.adapters.ProductAdapter;
import com.example.aexpress.databinding.ActivityCategoryBinding;
import com.example.aexpress.model.Product;
import com.example.aexpress.utils.Constants;

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

import java.util.ArrayList;

public class CategoryActivity extends AppCompatActivity {

    ActivityCategoryBinding binding;
    ProductAdapter productAdapter;
    ArrayList<Product> products;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityCategoryBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        products = new ArrayList<>();
        productAdapter = new ProductAdapter(this, products);


        int catId = getIntent().getIntExtra("catId", 0);
        String categoryName = getIntent().getStringExtra("categoryName");

        getSupportActionBar().setTitle(categoryName);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        getProducts(catId);

        GridLayoutManager layoutManager = new GridLayoutManager(this, 2);
        binding.productList.setLayoutManager(layoutManager);
        binding.productList.setAdapter(productAdapter);
    }

    @Override
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }

    void getProducts(int catId) {
        RequestQueue queue = Volley.newRequestQueue(this);

        String url = Constants.GET_PRODUCTS_URL + "?category_id=" + catId;
        StringRequest request = new StringRequest(Request.Method.GET, url, response -> {
```

```
        try {
            JSONObject object = new JSONObject(response);
            if(object.getString("status").equals("success")){
                JSONArray productsArray = object.getJSONArray("products");
                for(int i =0; i< productsArray.length(); i++) {
                    JSONObject childObj = productsArray.getJSONObject(i);
                    Product product = new Product(
                            childObj.getString("name"),
                            Constants.PRODUCTS_IMAGE_URL + childObj.getString("image"),
                            childObj.getString("status"),
                            childObj.getDouble("price"),
                            childObj.getDouble("price_discount"),
                            childObj.getInt("stock"),
                            childObj.getInt("id")

                    );
                    products.add(product);
                }
                productAdapter.notifyDataSetChanged();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }, error -> { });

    queue.add(request);
  }
}
```

CartActivity.java

```
package com.example.aexpress.activities;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DividerItemDecoration;
import androidx.recyclerview.widget.LinearLayoutManager;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.example.aexpress.R;
import com.example.aexpress.adapters.CartAdapter;
import com.example.aexpress.databinding.ActivityCartBinding;
import com.example.aexpress.model.Product;
import com.hishd.tinycart.model.Cart;
import com.hishd.tinycart.model.Item;
import com.hishd.tinycart.util.TinyCartHelper;

import java.util.ArrayList;
import java.util.Map;

public class CartActivity extends AppCompatActivity {

  ActivityCartBinding binding;
  CartAdapter adapter;
  ArrayList<Product> products;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityCartBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());

    products = new ArrayList<>();

    Cart cart = TinyCartHelper.getCart();

    for(Map.Entry<Item, Integer> item : cart.getAllItemsWithQty().entrySet()) {
      Product product = (Product) item.getKey();
      int quantity = item.getValue();
      product.setQuantity(quantity);

      products.add(product);
    }

    adapter = new CartAdapter(this, products, new CartAdapter.CartListener() {
      @Override
      public void onQuantityChanged() {
        binding.subtotal.setText(String.format("Rs. %.2f",cart.getTotalPrice()));
      }
    });
```

```
        LinearLayoutManager layoutManager = new LinearLayoutManager(this);
        DividerItemDecoration itemDecoration = new DividerItemDecoration(this, layoutManager.getOrientation());
        binding.cartList.setLayoutManager(layoutManager);
        binding.cartList.addItemDecoration(itemDecoration);
        binding.cartList.setAdapter(adapter);


        binding.subtotal.setText(String.format("Rs. %.2f",cart.getTotalPrice()));


        binding.continueBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startActivity(new Intent(CartActivity.this, CheckoutActivity.class));
            }
        });

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }
}
```

SearchActivity.java

```
package com.example.aexpress.activities;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.GridLayoutManager;

import android.os.Bundle;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.aexpress.adapters.ProductAdapter;
import com.example.aexpress.databinding.ActivitySearchBinding;
import com.example.aexpress.model.Product;
import com.example.aexpress.utils.Constants;

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

import java.util.ArrayList;

public class SearchActivity extends AppCompatActivity {


    ActivitySearchBinding binding;
    ProductAdapter productAdapter;
    ArrayList<Product> products;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySearchBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        products = new ArrayList<>();
        productAdapter = new ProductAdapter(this, products);


        String query = getIntent().getStringExtra("query");

        getSupportActionBar().setTitle(query);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        getProducts(query);

        GridLayoutManager layoutManager = new GridLayoutManager(this, 2);
        binding.productList.setLayoutManager(layoutManager);
        binding.productList.setAdapter(productAdapter);
    }
```



```
    @Override
    public boolean onSupportNavigateUp() {
        finish();
```

```
        return super.onSupportNavigateUp();
    }

    void getProducts(String query) {
        RequestQueue queue = Volley.newRequestQueue(this);

        String url = Constants.GET_PRODUCTS_URL + "?q=" + query;
        StringRequest request = new StringRequest(Request.Method.GET, url, response -> {
            try {
                JSONObject object = new JSONObject(response);
                if(object.getString("status").equals("success")){
                    JSONArray productsArray = object.getJSONArray("products");
                    for(int i =0; i< productsArray.length(); i++) {
                        JSONObject childObj = productsArray.getJSONObject(i);
                        Product product = new Product(
                                childObj.getString("name"),
                                Constants.PRODUCTS_IMAGE_URL + childObj.getString("image"),
                                childObj.getString("status"),
                                childObj.getDouble("price"),
                                childObj.getDouble("price_discount"),
                                childObj.getInt("stock"),
                                childObj.getInt("id")

                        );
                        products.add(product);
                    }
                    productAdapter.notifyDataSetChanged();
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }, error -> { });

        queue.add(request);
    }
}
```

CheckoutActivity.java

```
package com.example.aexpress.activities;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DividerItemDecoration;
import androidx.recyclerview.widget.LinearLayoutManager;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.util.Patterns;
import android.view.View;
import android.widget.Toast;

import com.android.volley.AuthFailureError;
import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import com.example.aexpress.R;
import com.example.aexpress.adapters.CartAdapter;
import com.example.aexpress.databinding.ActivityCheckoutBinding;
import com.example.aexpress.model.Product;
import com.example.aexpress.utils.Constants;
import com.hishd.tinycart.model.Cart;
import com.hishd.tinycart.model.Item;
import com.hishd.tinycart.util.TinyCartHelper;

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

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

public class CheckoutActivity extends AppCompatActivity {
```

```
ActivityCheckoutBinding binding;
CartAdapter adapter;
ArrayList<Product> products;
double totalPrice = 0;
final int tax = 18;
ProgressDialog progressDialog;
Cart cart;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    binding = ActivityCheckoutBinding.inflate(getLayoutInflater());
    setContentView(binding.getRoot());


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

    products = new ArrayList<>();

    cart = TinyCartHelper.getCart();

    for(Map.Entry<Item, Integer> item : cart.getAllItemsWithQty().entrySet()) {
        Product product = (Product) item.getKey();
        int quantity = item.getValue();
        product.setQuantity(quantity);

        products.add(product);
    }

    adapter = new CartAdapter(this, products, new CartAdapter.CartListener() {
        @Override
        public void onQuantityChanged() {
            binding.subtotal.setText(String.format("Rs. %.2f",cart.getTotalPrice()));
        }
    });

    LinearLayoutManager layoutManager = new LinearLayoutManager(this);
    DividerItemDecoration itemDecoration = new DividerItemDecoration(this, layoutManager.getOrientation());
    binding.cartList.setLayoutManager(layoutManager);
    binding.cartList.addItemDecoration(itemDecoration);
    binding.cartList.setAdapter(adapter);

    binding.subtotal.setText(String.format("Rs. %.2f",cart.getTotalPrice()));

    totalPrice = (cart.getTotalPrice().doubleValue() * tax / 100) + cart.getTotalPrice().doubleValue();
    binding.total.setText("Rs. " + totalPrice);

    binding.checkoutBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            processOrder();
        }
    });

    getSupportActionBar().setDisplayHomeAsUpEnabled(true);

}

void processOrder() {
    if (!validateFields()) {
        return;
    }
    progressDialog.show();
    RequestQueue queue = Volley.newRequestQueue(this);

    JSONObject productOrder = new JSONObject();
    JSONObject dataObject = new JSONObject();
    try {

        productOrder.put("address",binding.addressBox.getText().toString());
        productOrder.put("buyer",binding.nameBox.getText().toString());
        productOrder.put("comment", binding.commentBox.getText().toString());
        productOrder.put("created_at", Calendar.getInstance().getTimeInMillis());
        productOrder.put("last_update", Calendar.getInstance().getTimeInMillis());
        productOrder.put("date_ship", Calendar.getInstance().getTimeInMillis());
        productOrder.put("email", binding.emailBox.getText().toString());
        productOrder.put("phone", binding.phoneBox.getText().toString());
        productOrder.put("serial", "cab8c1a4e4421a3b");
        productOrder.put("shipping", "");
        productOrder.put("shipping_location", "");
        productOrder.put("shipping_rate", "0.0");
```

```
            productOrder.put("status", "WAITING");
            productOrder.put("tax", tax);
            productOrder.put("total_fees", totalPrice);

            JSONArray product_order_detail = new JSONArray();
            for(Map.Entry<Item, Integer> item : cart.getAllItemsWithQty().entrySet()) {
                Product product = (Product) item.getKey();
                int quantity = item.getValue();
                product.setQuantity(quantity);

                JSONObject productObj = new JSONObject();
                productObj.put("amount", quantity);
                productObj.put("price_item", product.getPrice());
                productObj.put("product_id", product.getId());
                productObj.put("product_name", product.getName());
                product_order_detail.put(productObj);
            }

            dataObject.put("product_order",productOrder);
            dataObject.put("product_order_detail",product_order_detail);

            Log.e("err", dataObject.toString());

        } catch (JSONException e) {}

        JsonObjectRequest request = new JsonObjectRequest(Request.Method.POST, Constants.POST_ORDER_URL, dataObject, new
Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    if (response.getString("status").equals("success")) {
                        Toast.makeText(CheckoutActivity.this, "Success order.", Toast.LENGTH_SHORT).show();
                        String orderNumber = response.getJSONObject("data").getString("code");
                        new AlertDialog.Builder(CheckoutActivity.this)
                                .setTitle("Order Successful")
                                .setCancelable(false)
                                .setMessage("Your order number is: " + orderNumber)
                        .setPositiveButton("Pay Now", new DialogInterface.OnClickListener() {
                            @Override
                            public void onClick(DialogInterface dialogInterface, int i) {
                                Intent intent = new Intent(CheckoutActivity.this, PaymentActivity.class);
                                intent.putExtra("orderCode", orderNumber);
                                startActivity(intent);
                            }
                        }).show();
                    } else {
                        new AlertDialog.Builder(CheckoutActivity.this)
                                .setTitle("Order Failed")
                                .setMessage("Something went wrong, please try again.")
                                .setCancelable(false)
                                .setPositiveButton("Close", new DialogInterface.OnClickListener() {
                                    @Override
                                    public void onClick(DialogInterface dialogInterface, int i) {

                                    }
                                }).show();
                        Toast.makeText(CheckoutActivity.this, "Failed order.", Toast.LENGTH_SHORT).show();
                    }
                    progressDialog.dismiss();
                    Log.e("res", response.toString());
                } catch (Exception e) {}
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {

            }
        }) {
            @Override
            public Map<String, String> getHeaders() throws AuthFailureError {
                Map<String, String> headers = new HashMap<>();
                headers.put("Security","secure_code");
                return headers;
            }
        } ;

        queue.add(request);
    }

    @Override
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }
```

```
    private boolean validateFields() {
        String name = binding.nameBox.getText().toString();
        String email = binding.emailBox.getText().toString();
        String phone = binding.phoneBox.getText().toString();
        String address = binding.addressBox.getText().toString();
        String shippingDate = binding.dateBox.getText().toString();

        if (TextUtils.isEmpty(name)) {
            binding.nameBox.setError("Full name is required.");
            binding.nameBox.requestFocus();
            return false;
        }

        if (TextUtils.isEmpty(email)) {
            binding.emailBox.setError("Email address is required.");
            binding.emailBox.requestFocus();
            return false;
        } else if (!Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
            binding.emailBox.setError("Invalid email address.");
            binding.emailBox.requestFocus();
            return false;
        }

        if (TextUtils.isEmpty(phone)) {
            binding.phoneBox.setError("Phone number is required.");
            binding.phoneBox.requestFocus();
            return false;
        } else if (phone.length() != 10) {
            binding.phoneBox.setError("Phone number should have exactly 10 digits.");
            binding.phoneBox.requestFocus();
            return false;
        }

        if (TextUtils.isEmpty(address)) {
            binding.addressBox.setError("Address is required.");
            binding.addressBox.requestFocus();
            return false;
        }

        if (TextUtils.isEmpty(shippingDate)) {
            binding.dateBox.setError("Shipping date is required.");
            binding.dateBox.requestFocus();
            return false;
        } else if (!shippingDate.matches("\\d{2}\\d{2}\\d{4}")) {
            binding.dateBox.setError("Shipping date should be in the format 'ddmmyyyy'.");
            binding.dateBox.requestFocus();
            return false;
        }

        return true;
    }
}
```

PaymentActivity.java

```
package com.example.aexpress.activities;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import com.example.aexpress.databinding.ActivityPaymentBinding;
import com.example.aexpress.utils.Constants;

public class PaymentActivity extends AppCompatActivity {

    ActivityPaymentBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityPaymentBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        String orderCode = getIntent().getStringExtra("orderCode");

        binding.webview.setMixedContentAllowed(true);
        binding.webview.loadUrl(Constants.PAYMENT_URL + orderCode);

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
```

```
    @Override
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }
}
```

CartAdapter.java

```
package com.example.aexpress.adapters;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.drawable.ColorDrawable;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Toast;

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

import com.bumptech.glide.Glide;
import com.example.aexpress.R;
import com.example.aexpress.activities.CartActivity;
import com.example.aexpress.databinding.ItemCartBinding;
import com.example.aexpress.databinding.ItemCategoriesBinding;
import com.example.aexpress.databinding.QuantityDialogBinding;
import com.example.aexpress.model.Product;
import com.hishd.tinycart.model.Cart;
import com.hishd.tinycart.util.TinyCartHelper;

import java.util.ArrayList;

public class CartAdapter extends RecyclerView.Adapter<CartAdapter.CartViewHolder> {

    Context context;
    ArrayList<Product> products;
    CartListener cartListener;
    Cart cart;

    public interface CartListener {
        public void onQuantityChanged();
    }


    public CartAdapter(Context context, ArrayList<Product> products, CartListener cartListener) {
        this.context = context;
        this.products = products;
        this.cartListener = cartListener;
        cart = TinyCartHelper.getCart();
    }

    @NonNull
    @Override
    public CartViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new CartViewHolder(LayoutInflater.from(context).inflate(R.layout.item_cart, parent,false));
    }

    @Override
    public void onBindViewHolder(@NonNull CartViewHolder holder, int position) {
        Product product = products.get(position);
        Glide.with(context)
                .load(product.getImage())
                .into(holder.binding.image);

        holder.binding.name.setText(product.getName());
        holder.binding.price.setText("Rs. " + product.getPrice());
        holder.binding.quantity.setText(product.getQuantity() + " item(s)");

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @SuppressLint("ResourceAsColor")
            @Override
            public void onClick(View view) {
                QuantityDialogBinding quantityDialogBinding = QuantityDialogBinding.inflate(LayoutInflater.from(context));

                AlertDialog dialog = new AlertDialog.Builder(context)
                        .setView(quantityDialogBinding.getRoot())
                        .create();
```

```
            dialog.getWindow().setBackgroundDrawable(new ColorDrawable(android.R.color.transparent));

            quantityDialogBinding.productName.setText(product.getName());
            quantityDialogBinding.productStock.setText("Stock: " + product.getStock());
            quantityDialogBinding.quantity.setText(String.valueOf(product.getQuantity()));
            int stock = product.getStock();


            quantityDialogBinding.plusBtn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    int quantity = product.getQuantity();
                    quantity++;

                    if(quantity>product.getStock()) {
                        Toast.makeText(context, "Max stock available: "+ product.getStock(), Toast.LENGTH_SHORT).show();
                        return;
                    } else {
                        product.setQuantity(quantity);
                        quantityDialogBinding.quantity.setText(String.valueOf(quantity));
                    }

                    notifyDataSetChanged();
                    cart.updateItem(product, product.getQuantity());
                    cartListener.onQuantityChanged();
                }
            });

            quantityDialogBinding.minusBtn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    int quantity = product.getQuantity();
                    if(quantity > 1)
                        quantity--;
                    product.setQuantity(quantity);
                    quantityDialogBinding.quantity.setText(String.valueOf(quantity));

                    notifyDataSetChanged();
                    cart.updateItem(product, product.getQuantity());
                    cartListener.onQuantityChanged();
                }
            });

            quantityDialogBinding.saveBtn.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    dialog.dismiss();
//                    notifyDataSetChanged();
//                    cart.updateItem(product, product.getQuantity());
//                    cartListener.onQuantityChanged();
                }
            });

            dialog.show();
        }
    });
}

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

public class CartViewHolder extends RecyclerView.ViewHolder {

    ItemCartBinding binding;
    public CartViewHolder(@NonNull View itemView) {
        super(itemView);
        binding = ItemCartBinding.bind(itemView);
    }
}
}
```

CategoryAdapter.java

```
package com.example.aexpress.adapters;

import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.text.Html;
import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.aexpress.R;
import com.example.aexpress.activities.CategoryActivity;
import com.example.aexpress.databinding.ItemCategoriesBinding;
import com.example.aexpress.model.Category;

import java.util.ArrayList;

public class CategoryAdapter extends RecyclerView.Adapter<CategoryAdapter.CategoryViewHolder> {

    Context context;
    ArrayList<Category> categories;

    public CategoryAdapter(Context context, ArrayList<Category> categories) {
        this.context = context;
        this.categories = categories;
    }

    @NonNull
    @Override
    public CategoryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new CategoryViewHolder(LayoutInflater.from(context).inflate(R.layout.item_categories, parent, false));
    }

    @Override
    public void onBindViewHolder(@NonNull CategoryViewHolder holder, int position) {
        Category category = categories.get(position);
        holder.binding.label.setText(Html.fromHtml(category.getName()));
        Glide.with(context)
                .load(category.getIcon())
                .into(holder.binding.image);

        holder.binding.image.setBackgroundColor(Color.parseColor(category.getColor()));

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context, CategoryActivity.class);
                intent.putExtra("catId", category.getId());
                intent.putExtra("categoryName", category.getName());
                context.startActivity(intent);
            }
        });
    }

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

    public class CategoryViewHolder extends RecyclerView.ViewHolder {
        ItemCategoriesBinding binding;

        public CategoryViewHolder(@NonNull View itemView) {
            super(itemView);
            binding = ItemCategoriesBinding.bind(itemView);
        }
    }
}
```

ProductAdapter.java

```
package com.example.aexpress.adapters;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.example.aexpress.R;
import com.example.aexpress.activities.ProductDetailActivity;
import com.example.aexpress.databinding.ItemProductBinding;
import com.example.aexpress.model.Product;
```

```
import java.lang.reflect.Array;
import java.util.ArrayList;

public class ProductAdapter extends RecyclerView.Adapter<ProductAdapter.ProductViewHolder> {

    Context context;
    ArrayList<Product> products;

    public ProductAdapter(Context context, ArrayList<Product> products) {
        this.context = context;
        this.products = products;
    }

    @NonNull
    @Override
    public ProductViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        return new ProductViewHolder(LayoutInflater.from(context).inflate(R.layout.item_product, parent, false));
    }

    @Override
    public void onBindViewHolder(@NonNull ProductViewHolder holder, int position) {
        Product product = products.get(position);
        Glide.with(context)
                .load(product.getImage())
                .into(holder.binding.image);
        holder.binding.label.setText(product.getName());
        holder.binding.price.setText("Rs. " + product.getPrice());

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(context, ProductDetailActivity.class);
                intent.putExtra("name", product.getName());
                intent.putExtra("image", product.getImage());
                intent.putExtra("id", product.getId());
                intent.putExtra("price", product.getPrice());
                context.startActivity(intent);
            }
        });
    }

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

    public class ProductViewHolder extends RecyclerView.ViewHolder {

        ItemProductBinding binding;

        public ProductViewHolder(@NonNull View itemView) {
            super(itemView);
            binding = ItemProductBinding.bind(itemView);
        }
    }
}
```

Category.java

```
package com.example.aexpress.model;

public class Category {
    private String name, icon, color, brief;
    private int id;

    public Category(String name, String icon, String color, String brief, int id) {
        this.name = name;
        this.icon = icon;
        this.color = color;
        this.brief = brief;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getIcon() {
```

```
        return icon;
    }

    public void setIcon(String icon) {
        this.icon = icon;
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public String getBrief() {
        return brief;
    }

    public void setBrief(String brief) {
        this.brief = brief;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```

Product.java

```
package com.example.aexpress.model;
import com.hishd.tinycart.model.Item;

import java.io.Serializable;
import java.math.BigDecimal;

public class Product implements Item, Serializable {

    private String name, image, status;
    private double price, discount;
    private int stock, id;
    private int quantity;

    public Product(String name, String image, String status, double price, double discount, int stock, int id) {
        this.name = name;
        this.image = image;
        this.status = status;
        this.price = price;
        this.discount = discount;
        this.stock = stock;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getImage() {
        return image;
    }

    public void setImage(String image) {
        this.image = image;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public double getPrice() {
```

```
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public double getDiscount() {
        return discount;
    }

    public void setDiscount(double discount) {
        this.discount = discount;
    }

    public int getStock() {
        return stock;
    }

    public void setStock(int stock) {
        this.stock = stock;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Override
    public BigDecimal getItemPrice() {
        return new BigDecimal(price);
    }

    @Override
    public String getItemName() {
        return name;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }
}
```

Constants.java

```
package com.example.aexpress.utils;

public class Constants {
    public static String API_BASE_URL = "http://vivekchaurasia.me/backend";
    public static String GET_CATEGORIES_URL = API_BASE_URL + "/services/listCategory";
    public static String GET_PRODUCTS_URL = API_BASE_URL + "/services/listProduct";
    public static String GET_OFFERS_URL = API_BASE_URL + "/services/listFeaturedNews";
    public static String GET_PRODUCT_DETAILS_URL = API_BASE_URL + "/services/getProductDetails?id=";
    public static String POST_ORDER_URL = API_BASE_URL + "/services/submitProductOrder";
    public static String PAYMENT_URL = API_BASE_URL + "/services/paymentPage?code=";

    public static String NEWS_IMAGE_URL = API_BASE_URL + "/uploads/news/";
    public static String CATEGORIES_IMAGE_URL = API_BASE_URL + "/uploads/category/";
    public static String PRODUCTS_IMAGE_URL = API_BASE_URL + "/uploads/product/";
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.aexpress">

    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@drawable/ic_launcher"
        android:supportsRtl="true"
```

```
        android:theme="@style/Theme.AExpress"
        android:usesCleartextTraffic="true">
        <activity
            android:name=".activities.SearchActivity"
            android:exported="false"
            android:theme="@style/Theme.AExpress.Actionbar"/>
        <activity
            android:name=".activities.CategoryActivity"
            android:exported="false"
            android:theme="@style/Theme.AExpress.Actionbar" />
        <activity
            android:name=".activities.PaymentActivity"
            android:exported="false"
            android:label="Payment"
            android:theme="@style/Theme.AExpress.Actionbar" />
        <activity
            android:name=".activities.CheckoutActivity"
            android:exported="false"
            android:label="Checkout"
            android:theme="@style/Theme.AExpress.Actionbar" />
        <activity
            android:name=".activities.CartActivity"
            android:exported="false"
            android:label="Shopping Cart"
            android:theme="@style/Theme.AExpress.Actionbar" />
        <activity
            android:name=".activities.ProductDetailActivity"
            android:exported="false"
            android:theme="@style/Theme.AExpress.Actionbar" />
        <activity
            android:name=".activities.MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <meta-data
            android:name="preloaded_fonts"
            android:resource="@array/preloaded_fonts" />
    </application>

</manifest>
```

build.gradle

```
plugins {
    id 'com.android.application'
}

android {
    compileSdk 32

    buildFeatures {
        viewBinding true
    }

    defaultConfig {
        applicationId "com.example.aexpress"
        minSdk 23
        targetSdk 32
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
```

```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.4.1'
    implementation 'com.google.android.material:material:1.6.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.3'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'

    /* For Loading images from internet - Start */
    implementation 'com.github.bumptech.glide:glide:4.13.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.13.0'
    /* For Slider - End */

    /* For Rounded Image View */
    implementation 'com.makeramen:roundedimageview:2.3.0'
    /* For Material Search Bar */
    implementation 'com.github.mancj:MaterialSearchBar:0.8.5'

    /* For Slider - Start */
    // Material Components for Android. Replace the version with the latest version of Material Components library.
    implementation 'com.google.android.material:material:1.5.0'
    // Circle Indicator (To fix the xml preview "Missing classes" error)
    implementation 'me.relex:circleindicator:2.1.6'
    implementation 'org.imaginativeworld.whynotimagecarousel:whynotimagecarousel:2.1.0'
    /* For Slider - End */

    implementation 'com.android.volley:volley:1.2.1'

    implementation 'com.github.hishd:TinyCart:1.0.1'
    implementation 'com.github.delight-im:Android-AdvancedWebView:v3.2.1'
}
```

5.2.1.Code Efficiency

This Android app project uses Java for efficient app development and integrates well with the backend database through PHPMyAdmin, Mysql database and the web hosting via NameCheap. The optimized UI design includes navigation bars to search products with multiple pages for displaying product information and order details. The use of PHP programming language allows for easy updates to product details, categories, and news information. This project's efficient programming languages, UI design, and database management make it a user-friendly solution for online shopping. Overall, this Jersey app project is a reliable and efficient solution for online shopping needs.

5.3.Testing Approach

Following testing approaches has been performed on the system: Unit Testing, Integration Testing, System Testing

5.3.1.Unit Testing

Unit Testing was performed by testing each module as a separate function. Each module in the project was tested separately to ensure that it is functioning as expected. For example, the Cart module was tested separately to verify that the details are added to the database.

5.3.2.Integration Testing

Integration Testing was performed by testing multiple modules at a time to ensure that they work together correctly. For example, the Payment module was tested with the My Orders module to ensure that the payment made for an order is displayed correctly in the My Orders module.

5.3.3.System Testing

System Testing was performed on the completely integrated system to assess whether it meets the requirements and passes the defined test cases. The Android app has successfully met all the criteria and has passed all the tests as separate units and as a completely integrated system.

Note: Acceptance testing was done by my friends and the review was they were really liking this app because of no login credentials as they don't need to enter their login credentials. But the drawback was they can't see their transaction history.

5.4.Modification and Improvements

After thoroughly testing and debugging the Android app, all the encountered errors were fixed by identifying and troubleshooting them. Exceptions were properly handled by incorporating try-catch blocks in relevant sections of the code, minimizing redundant code in index files. While it is impossible to completely eliminate all bugs, the majority of them have been resolved, with the remaining ones being addressed in ongoing development efforts.

To further enhance the functionality of the app, one possible improvement could be adding a product comparison feature to facilitate decision-making for users. Additionally, as there is no login system, direct bill generation cannot be incorporated into the app. Instead, a workaround has been provided to users with the option to email their order details, and receive a bill through email or for any queries on jerseysupport@gmail.com.

Furthermore, since a chatbot is not included in the app, an alternative option could be to include an email-based query system where users can send their queries through email and receive a response from the support team. By making these modifications and improvements, the overall user experience of the app can be enhanced, making it more efficient and user-friendly.

5.5 Deployment and Maintenance

Since the testing phase has been completely done with acceptance testing done by my friends, our Android Jersey app has been made available for download through a shared Google Drive link. The link has been made accessible to the target audience, and they can download the app by clicking on the link. The app will be in the form of an APK file that can be installed on any Android device.

As for maintenance, the app will require ongoing support to ensure that it continues to function correctly and remains relevant to the users. This support will involve fixing any issues that arise, adding new features, updating the app to keep up with changes in the Android operating system, and responding to user feedback. The maintenance activities will be carried out by the development team and will involve regular updates to the APK file available on Google Drive.

6.RESULTS AND DISCUSSIONS

1. Test Report

Test Condition	Input	Expected Result	Actual Result	Pass/ Fail
Search Product	Search for valid product name in search bar	Product list with matching products is displayed	Product list with matching products is displayed	Pass
	Search for invalid product name in search bar	No matching products found message is displayed	No matching products found message is displayed	Pass
Select Sport	Click on a sport category button	Product list of selected sport category is displayed	Product list of selected sport category is displayed	Pass
View Product	Click on a product in the product list	Product details page is displayed	Product details page is displayed	Pass
Add to Cart	Click on "Add to Cart" button in product details page	Product is added to cart and success message is displayed	Product is added to cart and success message is displayed	Pass
	Increase product quantity using "+" button	Quantity is increased and price is updated	Quantity is increased and price is updated	Pass
	Decrease product quantity using "-" button	Quantity is decreased and price is updated	Quantity is decreased and price is updated	Pass
Remove Product	Click on "Remove" button in the cart	Product is removed from cart and cart total is updated	Product is removed from cart and cart total is updated	Pass
Checkout	Fill all required fields in checkout form	User is taken to payment options page	User is taken to payment options page	Pass
	Leave a required field blank in checkout form	Error message is displayed and user cannot proceed to payment	Error message is displayed and user cannot proceed to payment	Pass

2. User Documentation

Welcome to **Jersey** App, an easy and hassle-free way to purchase sports products! This user documentation will guide you through the app's features and functionalities to help you buy your favourite sports products.

Let's get started!

Download the app:
First, download the app from the link below and install it on your device. Once installed, open the app and you'll see the home screen.

Link: https://drive.google.com/drive/folders/1ppqpaxC5DZJH86_6jtt84vKuVqHT3Eex?usp=sharing

Home Screen:

- Open the App named - “Jersey”.

- The home screen of the app displays 8 different sports team options, the best-selling product lists and with a navigation bar to search products. Choose any sports team option to see the products for that particular sport.

Product Selection:
After clicking on any sports team, a new page will appear where you can see the list of products related to that sport. Select the product that you want to buy by clicking on it. A new page with the product's description and an "Add to Cart" button will appear.

Adding Products to Cart:
Click on the "Add to Cart" button to add the product to your cart. You can click on any individual product to increase or decrease the quantity. You can also see the available stock and affected price.

View Cart:
To view your cart, click on the cart icon at the top right corner of the screen. A new page will appear where you can see all the products you've selected and a "Continue" button below.

Checkout:
After clicking on the "Continue" button, you will see the total cost details. You need to add the personal info which is compulsory such as name, email Id, phone number, address, shipping date and comments which is optional and you can see below the product order list.

Payment:

After filling all the details at the checkout page, click on "Process Checkout". A dialog box will appear after clicking on "Process Checkout" of a successful order with a "Pay Now" option. Click on it, and a payment page will appear. You can see your order id/ code, your name, and options to make payment either via Razorpay or bank transfer.

Order Notification (manual):

After placing an order, if the payment is successful, you will receive a confirmation email with the order details in 24 hours from our team. If the payment is unsuccessful, you will receive an email with the reason for the failure.

Congratulations, you have successfully purchased a sports product using Jersey App! If you face any issues, you can reach out to the support team by sending an email to jerseysupport@gmail.com.

7.CONCLUSION

1. Conclusion

The Android app project "Jersey" has been developed successfully with features like product browsing, selection, and checkout. However, it has some limitations like no login system, transaction history, or chatbot for query resolution. In the future, the project can be enhanced with features like a login system, transaction history, chatbot, and improved user interface.

2. Limitations

- No transaction history to view past purchases or orders
- No chatbot to solve queries or provide customer support
- Limited payment options available for users
- Limited product categories and sports teams available

3. Future scope of the project

- Provide a transaction history feature for users to view past purchases or orders
- Integrate a chatbot for customer support and query solving
- Increase the payment options available for users
- Expand the product categories and sports teams available for selection
- Implement a recommendation system to suggest products based on user preferences
- Add a feature for users to rate and review products
- Integrate social media sharing options for users to share their purchases or orders
- Implement a loyalty program to incentivize repeat purchases

4. Feasibility Study

After conducting a comprehensive feasibility study, I have concluded that the Jersey Project is technically, operationally, and economically feasible. I have completed the project, and the technologies and tools required have been thoroughly researched. The project has also been designed keeping in mind the operational requirements of the start up in future if any, ensuring that it can be easily integrated into the operations of the start up without disruption.

Furthermore, the cost of hosting the server on NameCheap was **Rs. 385.61**which is economically feasible. However, if the start up continues further then we need to check the other economic feasibility such as for marketing, storage of goods, etc.

As a result, I am confident that this Jersey project can be the future for the change in sports brand.

References

- Android Studio: https://developer.android.com/studio
- PHP: https://www.php.net/manual/en/intro-whatis.php
- MySQL: https://www.mysql.com/what-is-mysql/
- PayPal: https://developer.paypal.com/docs/api-basics/
- RazorPay: https://razorpay.com/docs/payment-gateway/
- Google's Android developer portal: https://developer.android.com/docs
- Volley: https://google.github.io/volley/
- RoundedView: https://github.com/vinc3m1/RoundedImageView
- Carousel: https://github.com/ImaginativeShohag/Why-Not-Image-Carousel
- Postman: https://www.postman.com/
- NameCheap: https://namecheap.com/
- Geeks for Geeks: https://www.geeksforgeeks.org/