Capstone Project final Report On

MyBookShop

An android app to rent or buy used/new book

Submitted by

Sanjay Kumar	**11500754**
Niraj Kumar Sahni	**11805773**
Mahesh Kumar	**11808635**
Asif Ahmad	**11703900**
S. Tujliman	**11703969**

Section	**:**	**KC373**
Course code	**:**	**CSE445**

Under the guidance of

Gurbakash Phonsa(15483)

Assistant professor

School of Computer Science and Engineering

Declaration

We hereby declare that the project work entitled “MyBookShop” is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara under the guidance of Ms. Manpreet Dhindsa during January to April 2021. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Name of Student 1 : Sanjay Kumar

Registration Number: 11500754

Name of Student 2 : Niraj Kumar Sahni

Registration Number: 11805773

Name of Student 3 : Mahesh Kumar

Registration Number: 11808635

Name of Student 4 : Asif Ahmad

Registration Number: 11703900

Name of Student 5 : S. Tujliman

Registration Number: 11703969

TOPIC APPROVAL PERFORMA

School of Computer Science and Engineering (SCSE)

Program : P132::B.Tech. (Computer Science & Engineering)

COURSE CODE : CSE445 **REGULAR/BACKLOG :** Regular **GROUP NUMBER :** CSERGC0373

Supervisor Name : Manpreet Dhindsa **UID :** 23551 **Designation :** Assistant Professor

Qualification : **M.tech** **Research Experience :**3 years

SR.NO.	**NAME OF STUDENT**	**Prov. Regd. No.**	**BATCH**	**SECTION**	**CONTACT NUMBER**
1	Mahesh Kumar	11808635	2018	K17GN	7009672284
2	Niraj Kumar Sahni	11805773	2018	K17CM	9877785317
3	S Tujliman	11703969	2017	K17MD	7317506584
4	Asif Ahmad	11703900	2017	K17MD	9935327516
5	Sanjay Kumar	11500754	2015	K17BN	7979008489

SPECIALIZATION AREA : Software Engineering **Supervisor Signature:** ____________________

PROPOSED TOPIC : My Bookshop

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	**Parameter**	**Rating (out of 10)**
1	Project Novelty: Potential of the project to create new knowledge	6.77
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	6.77
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.00
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	7.45
5	Social Applicability: Project work intends to solve a practical problem.	6.45
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	6.45

PAC Committee Members		
PAC Member (HOD/Chairperson) Name: Dalwinder Singh	UID: 11265	Recommended (Y/N): Yes
PAC Member (Allied) Name: Asha Rani	UID: 11332	Recommended (Y/N): Yes
PAC Member 3 Name: Dr. Parampreet Kaur	UID: 18758	Recommended (Y/N): Yes

Final Topic Approved by PAC: **My Bookshop**

Overall Remarks: Approved

PAC CHAIRPERSON Name:14307::Raj Karan Singh **Approval Date:** 12 Mar 2021

Certificate

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfilment of the conditions for the award of B. Tech degree in Computer Science Engineering from Lovely Professional University, Phagwara

Signature and Name of the Mentor: Manpreet Dhindsa

Designation: Assistant Professor

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date:

Acknowledgements

We humbly take this opportunity to present our votes of thanks to all those guideposts who really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

We are really grateful to our mentor Manpreet Dhindsa for providing us with an opportunity to undertake this project in this university and providing us with all the bright and innovative ideas for making our project a really worthwhile of running in an organization. We are highly thankful for her active support, valuable time and advice, whole-hearted guidance, sincere cooperation and pains-taking involvement during the study and in completing the capstone project within the time stipulated.

We are thankful to all those, particularly our friends, who have been instrumental in creating proper, healthy and conductive environment and including new and fresh innovative ideas for us during the project, without their help, it would have been extremely difficult for us to prepare the project in a time bound framework.

Niraj Kumar sahni

Sanjay Kumar

Mahesh Kumar

Asif Ahmad

S Tujliman

Abstract

Nowadays, the network plays an import role in person's life. In the process of the improvement of the person's living standard, person's demands of the life's quality and efficiency is higher, the traditional bookstore's inconvenience gradually emerge, and the My Bookshop has gradually been used in public.

The My Bookshop is a revolution of book industry. The traditional bookstore operation time, address and space is limited, so the types of books and books to find received a degree of restriction. But the My Bookshop broke the management mode of traditional bookstore, as long as you have a computer, you can buy the book anywhere, saving time and effort, shortening the time of book selection link effectively. The My Bookshop system based on the principle of provides convenience and service to people.

TABLE OF CONTENTS

1. Introduction

1.1 Purpose and Motivation

With the My Bookshop system, consumers do not need to blindly go to various places to find their own books, but only in a computer connected to the Internet log on My Bookshop system, in the search box, type you want to find of the book information retrieval, you can efficiently know whether the site has its own books, if you can online direct purchase, if not, you can change the home bookshop to continue to search or provide advice to the seller in order to supply, This greatly facilitates every consumer, saving time and labour. The online bookshop system can not only reduce costs, save time, space, to bring convenience to everyone, but also to promote the development of the logistics industry, serve three purposes, mutual benefit. More importantly, in today's world, the increasingly close ties between countries, more frequent exchanges, the economy tends to globalization, which promote the future development of My Bookshop system has some practical significance.

The motivation to create this project has many sources

• Interest to develop a good user-friendly android App with many online transactions using a database.

• To increase our knowledge horizon in technologies like Android studio, Java, Xml and firebase

• To gain good experience in Android before joining in a full-time job.

2. Project Overview

2.1 Research Background

The Internet has been favoured by more and more people for its high efficiency and richness, and android application has emerged. The My Bookshop is a form of application and book sales industry in one form, it has many advantages, such as: Bookshop size is relatively small, cost savings, transaction activities can be anytime, anywhere, improve service efficiency. The information is complete, more convenient retrieval, the new book information on the new, consumers can see in a timely manner, trading activities can be launched immediately, so My Bookshop in today's era of development is extremely rapid. My Bookshop system is the main function of the trading platform for the site, consumers can connect to the Internet through the computer into the online bookshop and then check the book information, if you need to purchase should be registered landing, select their own books, submit orders and pay operation to complete the entire book ordering process, to achieve online transactions.

This project has the following functionalities:

1) **A Home page**

   When a user is successfully logged in to the application they will be navigated to thes Page. Here all the book categories will be displayed and a search keyword option will be provided to search for the book user requires. It will include some special sections like recommended titles, weekly special books.

2) **Searching**

   A search Textbox is provided to user where user can enter the book title and by clicking on the search icon the user will see the recommended books according to the searched title of the book.

3) **Description of book**

   A short and clear description of the book will be written below the title of the book along with the picture and the name of the author of the book.

4) **Shopping Cart**

   All the books user has selected will be there in the shopping cart. Shopping cart can be edited, deleted or updated by the user. A final shopping cart description will be displayed which will include all the books the user has chosen and the final total cost.

5)Manage user accounts

Every user shall have an user account to access all the features of application. User can logged in using login page and logged out using the logout page. All the members sessions will be stored in the database.

6)Administrator

The Administrator will be provided with some special features like

- Adding or deleting a book category
- Adding or deleting a user
- Managing user orders.
- Adding or deleting a Card type.

7)Seller

Like users sellers are also required to register in the Application that he/she can add item with description and image which will be than reflected to users after administrator has approved the item.

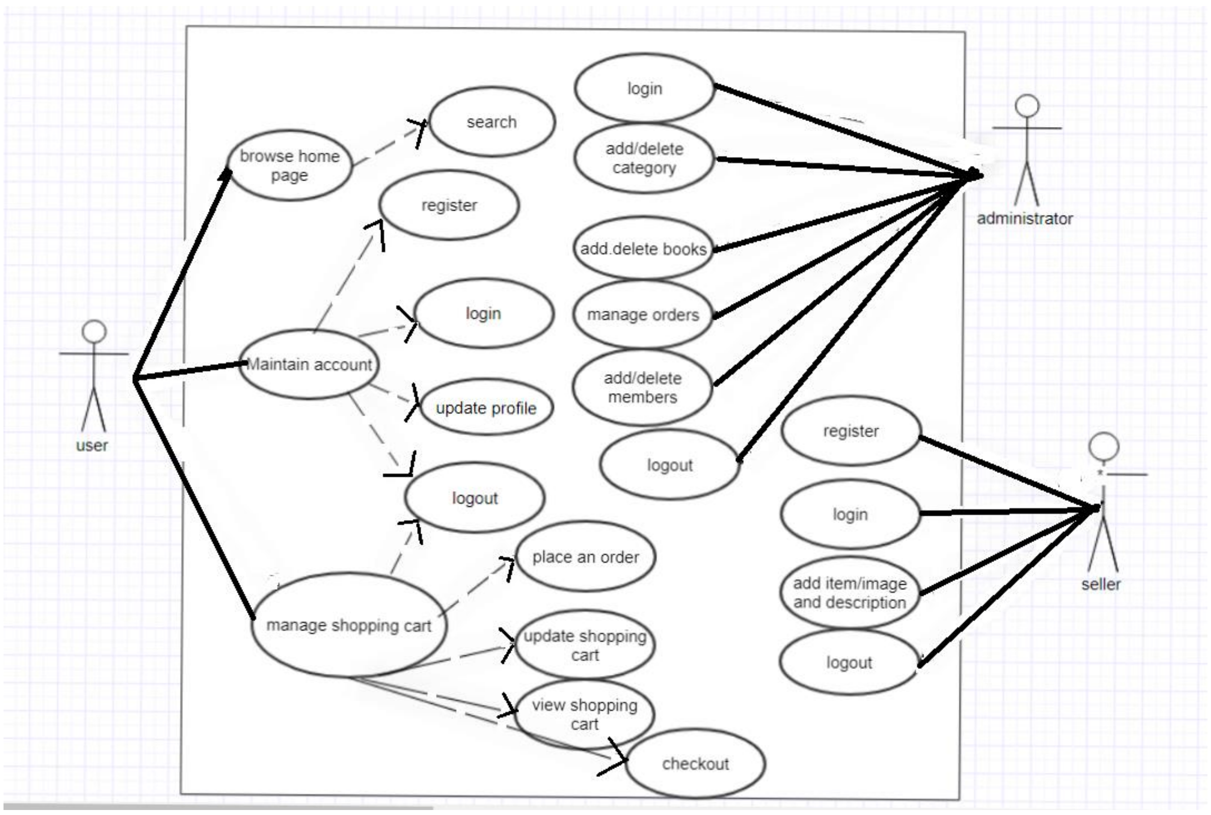
3.<u>Requirement specifications</u>

3.1 Main requirement

The Main Requirements include Android studio and Java development kit to develop the Android application, firebase to design the database and Emulator as a main browser to run the Application

3.2 Critical Use Case Requirements

User, Administrator and seller are the three actors included in the My BookShop. Fig. shows the use case diagram for this Application



MyBookShope use case DIAGRAM

Use Cases:

<u>Browse Catalog</u>

1) Searching for a Book

- **Purpose:** A user searched for a article or a book of their desire by tapping on the category and book title. In the database a select query is used to fetch data from the database and displays the fetched information.
- **Actor:** User
- **Input:** The user have to choose a category or enter book title in a text box provided.
- **Output:** The system will fetch the information and it will display the books which matches the selected search criteria. A dataset will be created as a result of select query and Later the dataset is binded to the data repeater to display the selected data.

<u>Maintaining Account</u>

1)Register

- **Purpose:** If the user is not registered hence do not have an account then he/she will be asked to register to the application.
- **Actor:** User
- **Input:** The user has to enter the asked details, user can skip the optional details if they like to, in the registration form in the required fields accordingly. The fields include **1.** User_name **2.** Password **3.** confirm password 4. first name **5.** last name **6.** email **7.** Address **8.** Phone
- **Output:** After successful registration the user will be navigated to the main home page.

2)Login

- **Purpose:** to get access to all the features and functionalities of My BookShop user should login using their username and password.
- **Actor:** User
- **Input:** The user has to provide their username and password.
- **Output:** after the successful login the user will be navigated to the main home page. Or if the user enters wrong information, A pop up message will appear asking to check the provided information.

3) Update Profile

- **Purpose:** If the user chooses to edit any personal account information then they can update their selected fields and all the edited data will be updated in the database through an update query.
- **Actor:** User
- **Input:** The user has to edit his account information.
- **Output:** The system will update the edited information in the database using an SQL update query.

5)Logout

- **Purpose:** When the user is done with the shopping and wants to terminate his session and log out of the application then he can use the logout option.
- **Actor:** User

- **Input:** The user has to click on the logout/sign out button.
- **Output:** The user's account session will end as soon as they click on logout button, they have to login again to gain access to the functionalities of the application.

Manage Shopping Cart

1)Place an order

- **Purpose:** If the user gets their desire book and want to purchase or rent the book then he can place an order by clicking the add to cart button and entering the quantity required under the book description.
- **Actor:** User
- **Input:** The user will have to enter the quantity he/she wants and tap on the add to cart button.
- **Output:** The items will be appeared to the user's shopping cart.

2)Update Shopping Cart

- **Purpose:** If the user wants to increase or decrease the quantity of a book or the user wants to change the book then they can update their shopping cart.
- **Actor:** User
- **Input:** The user has to select the details button in the shopping cart summary to edit and update their order details.
- **Output:** Soon updated order details will be reflected in the shopping cart summary.

3)View Shopping Cart

- **Purpose:** If the user wants to view the books they put to the shopping cart then they can click on the shopping cart button.
- **Actor:** User
- **Input:** The user has to click on the shopping cart icon button.
- **Output:** The user's shopping cart information will be displayed to the user in the form of a tabular structure with all the books a user has added and the quantity of the. A total of all the books will also be displayed at the bottom.

Administrator

1)Login

- **Purpose:** If the Administrator wants to get access to all the functionalities of My BookShop he should login using his username and password.
- **Actor:** Administrator
- **Input:** The Administrator has to provide his username and password.
- **Output:** after the successful login the user will be navigated to the menu page. Or if the admin enters wrong information, A pop up message will appear asking to check the provided information.

2)Add or Delete Category

- **Purpose:** A administrator can add or delete the book category. As soon as the administrator has done any changes , the database will be updated and hence it will be reflected in the application to the user and seller as well.
- **Actor:** Administrator
- **Input:** If the Administrator wanted to add a book category the feature of adding and deleting the book category is provided by clicking on it he can perform the operation .
- **Output:** Soon the updated categories list will be displayed in the main home page.

3) Add or Delete Book

- **Purpose:** If the Administrator wants to add or delete a book then he has been given the rights to add or delete the book, the database will be updated and hence it will be reflected in the application to the user.
- **Actor:** Administrator
- **Input:** If the administrator wanted to add book, functionality is provided to the administrator, by providing book details like book title, description and a picture he can add the book.
- **Output:** Soon the updated book list will be displayed in the main home page.

3)Manage Orders

- **Purpose:** Administrator can manage orders like he can add or delete an order using his administration rights.
- **Actor:** Administrator
- **Input:** If the administrator wanted to add or delete order functionality is provided to the administrator.
- **Output:** The updated orders list will be processed to the users.

4)Add or delete member

- **Purpose:** If the Administrator wants to add or delete a book then he has been given the rights to add or delete the book, the database will be updated.
- **Actor:** Administrator
- **Input:**. If the Administrator wanted to add or remove a member, the feature of adding and deleting the member is provided by clicking on it he can perform the operation .
- **Output:** The database will be updated.

5)Logout

- **Purpose:** When the administrator is done with the management and wants to terminate his session and log out of the application then he can use the logout option.
- **Actor:** Administrator
- **Input:** The Administrator has to click on the logout button.
- **Output:** The Administrator's account session will end as soon as they click on logout button, they have to login again to gain access to the functionalities of the application.

<u>Seller</u>

1)Register

- **Purpose:** If the seller is not registered hence do not have an account then he/she will be asked to register to the application.
- **Actor:** seller
- **Input:** The seller has to enter the asked details, user can skip the optional details if they like to, in the registration form in the required fields accordingly. The fields include **1.** User_name **2.** Password **3.** confirm password 4. first name **5.** last name **6.** email **7.** Address **8.** Phone
- **Output:** After successful registration the seller will be navigated to the add product page.

2)Login

- **Purpose:** to get access to all the features and functionalities of My BookShop seller should login using their username and password.
- **Actor:** seller
- **Input:** The seller will provide his username and password.
- **Output:** after the successful login the seller will be navigated to the add product page. Or if the seller enters wrong information, A pop up message will appear asking to check the provided information.

3)Add item

- **Purpose:** here seller has to upload item with title, description and a clear picture of the item.
- **Actor:** seller
- **Input:** the seller will upload a item picture, description and title in the given textbox
- **Output:** after adding item it will be verified by administrator after being approved it will be reflected to the user

4)Logout

- **Purpose:** When the seller is done with adding new products and wants to terminate his session and log out of the application then he can use the logout option.
- **Actor:** Seller
- **Input:** The seller will have to click on the logout button.

- **Output:** The seller's account session will end as soon as they click on logout button, they have to login again to gain access to the functionalities of the application.

Environment

• The My Bookshop is be developed in Android studio environment.

• Java and xml are used as the programming language.

Task Breakdown

1) **The inception phase** In this phase we as a group discussed thoroughly and defined all the project's requirements. This phase we made a vision document, discussed about the Project Plan, a Software Quality Assurance Plan, and a Demonstration. Vision Document, it include all the project's requirements and overview, project purpose, project goals and the risks associated with it, constraints, and direction. It includes a listing of the main requirements and their respective Use case models to illustrate the functionality.

2) **The elaboration phase** in this phase we work upon the project's architecture. In this phase we concluded that the project should be based on client-server architecture i.e. 3- tier architecture. This phase also include the production of revisions to the Project Plan, an Architecture Design Plan, a formal specification, Testing Plan.

3) **The production phase** In this phase we defined about the project implementation and testing. This phase also includes the user manual of the application.

   User Manual includes an explanations of common usage of the application at the user point of view. We have provided the use case diagram/ sequence diagrams of the user, administrator and seller as well for the better understanding of the functionalities. Also we have attached the actual screenshot of the application following with the further explanation.

Architecture design

1.Introduction

The purpose of architecture design to provide an understanding of the 3-tier architecture of the My BookShop application. It shows the presentation tier, the middle tier consisting of classes, sequence diagrams, and the data tier.

2.Architecture

Three-tier architecture is a well-established software application architecture that organizes applications into three logical and physical computing tiers: the presentation tier, or user interface; the application tier, where data is processed; and the data tier, where the data associated with the application is stored and managed.

The Architecture of My BookShop is based on three-tier architecture. The three logical tiers are

• Presentation tier -Android studio, Mater Pages, Images.

• Middle tier – Java classes.

• Data tier- Database Figure below shows the model of 3-tier architecture.

The main reason for considering three-tier architecture for the My Bookshop store is as follows:

Flexibility:

• Management of data is independent from the physical storage support,

• Maintenance of the business logic is easier,

• Migration to new graphical environments is faster.

• If there is a minor change in the business logic, we don’t have to install the entire system in individual user’s device.

Reusability:

• Reusability of business logic is greater for the presentation layer. As this component is developed and tested, we can use it in any other project and would be helpful for future use.

Security:

- More secured architecture since the client cannot access the database directly.

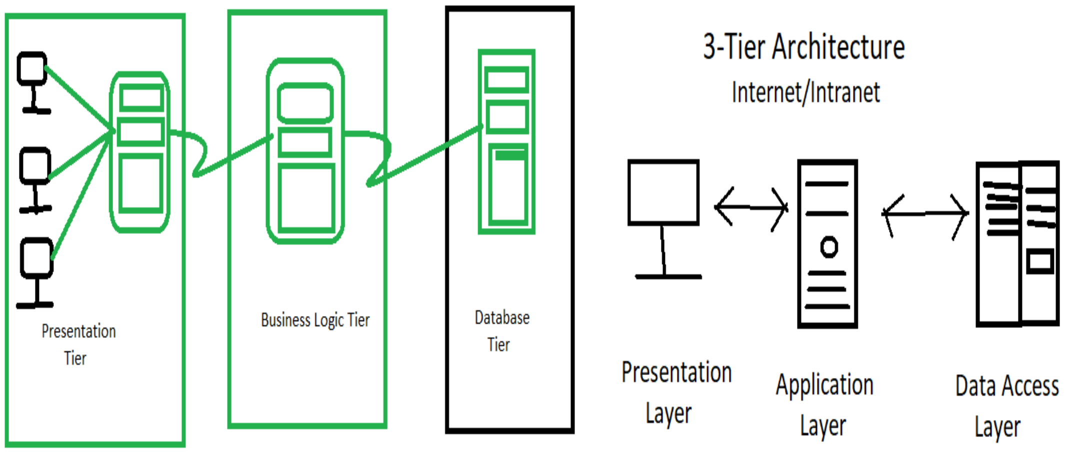
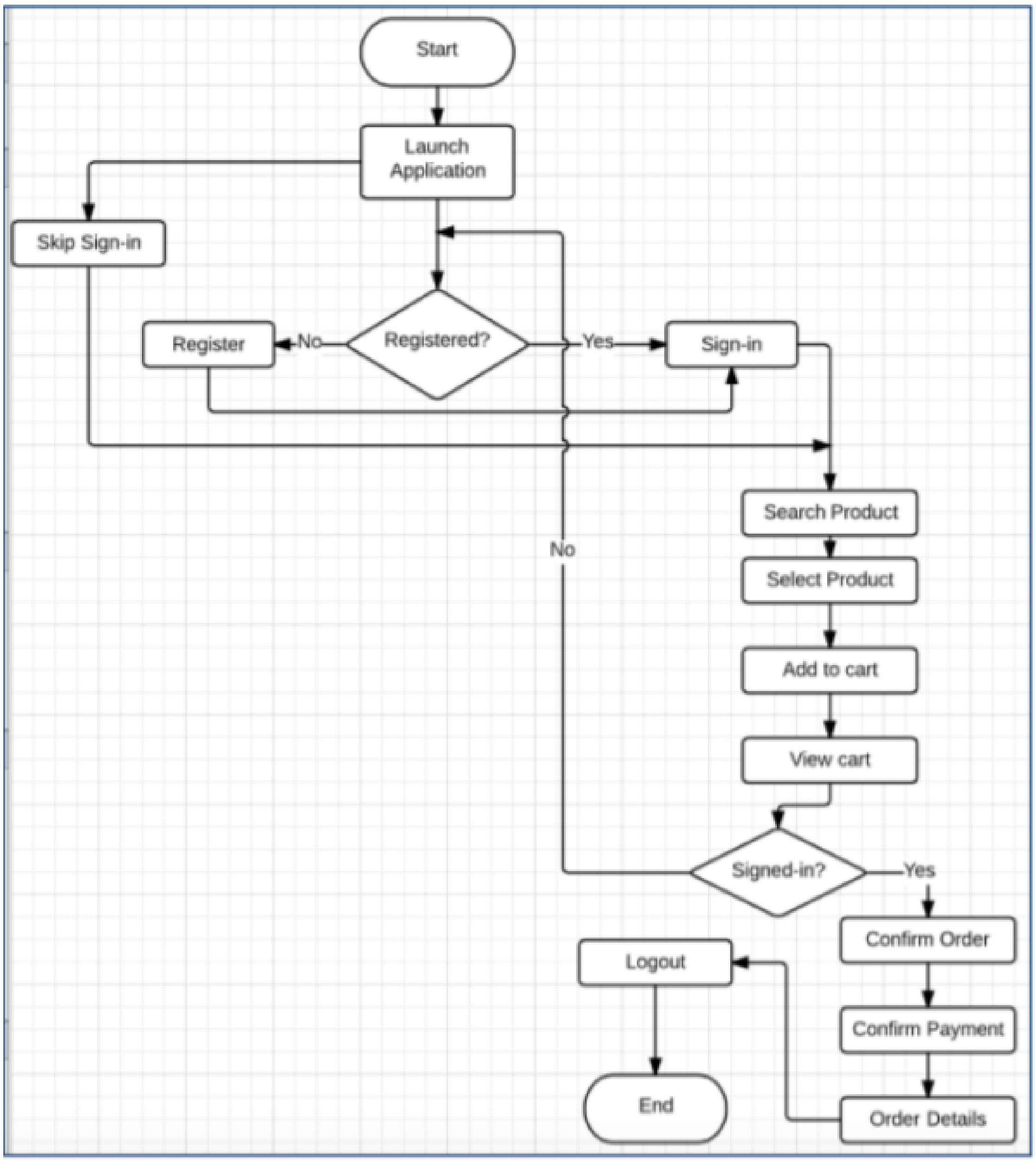


Figure: 3-tier Architecture

The diagram below captures the page flow for user in the My BookShop Application

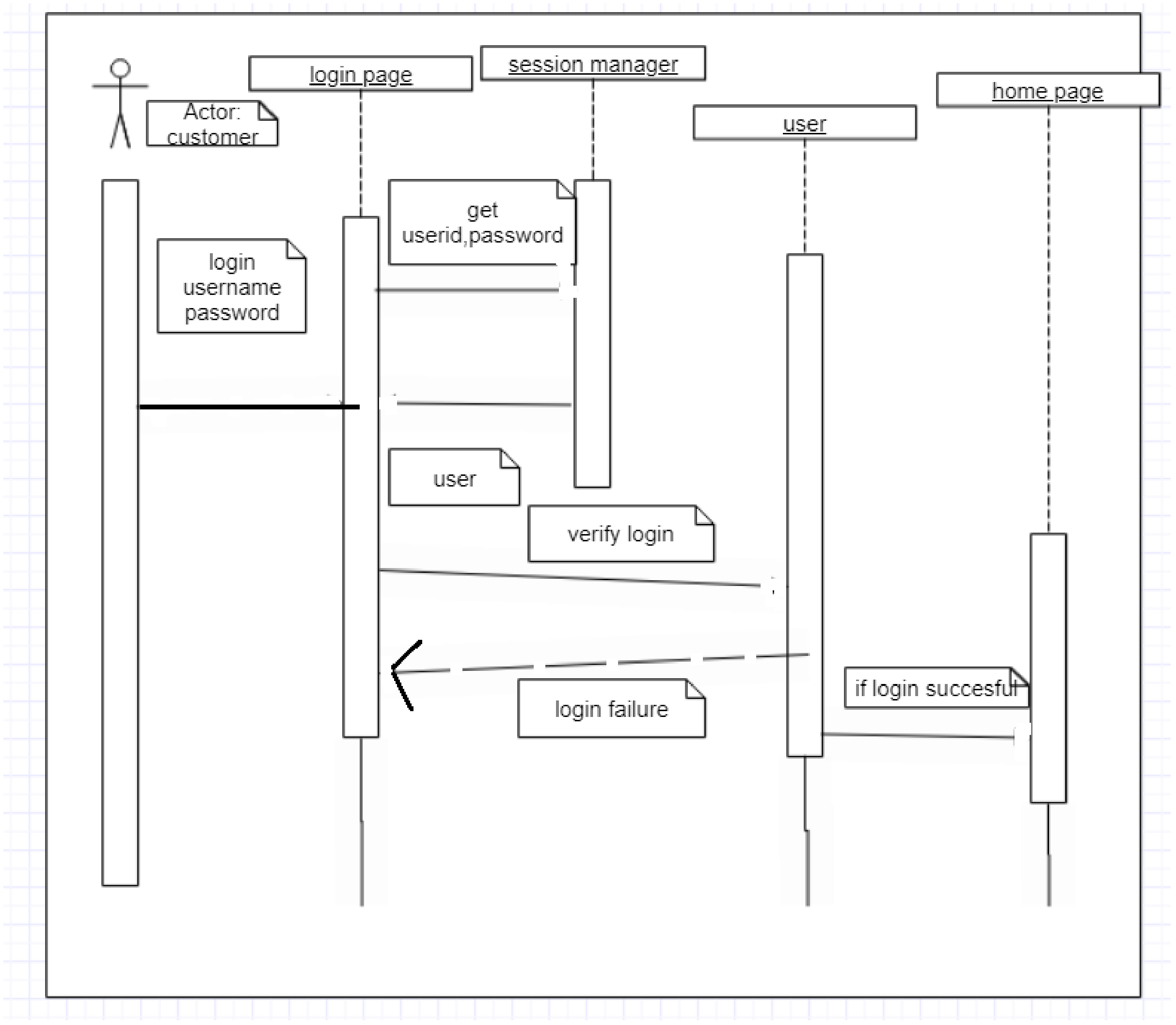


Buyer in My Bookshop

The diagram below captures the page flow for seller in the My BookShop Application

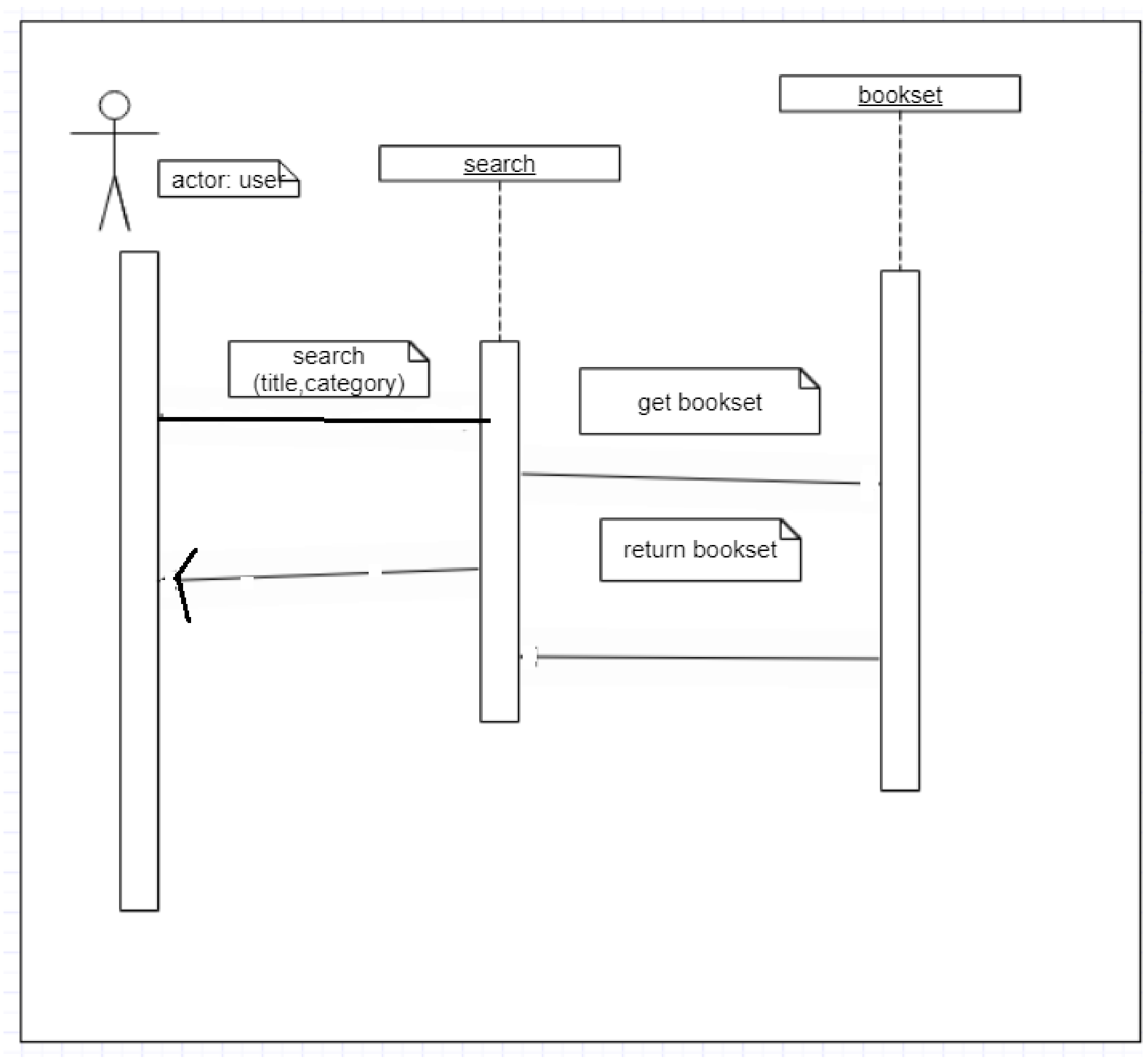
Sequence Diagrams

1) Login



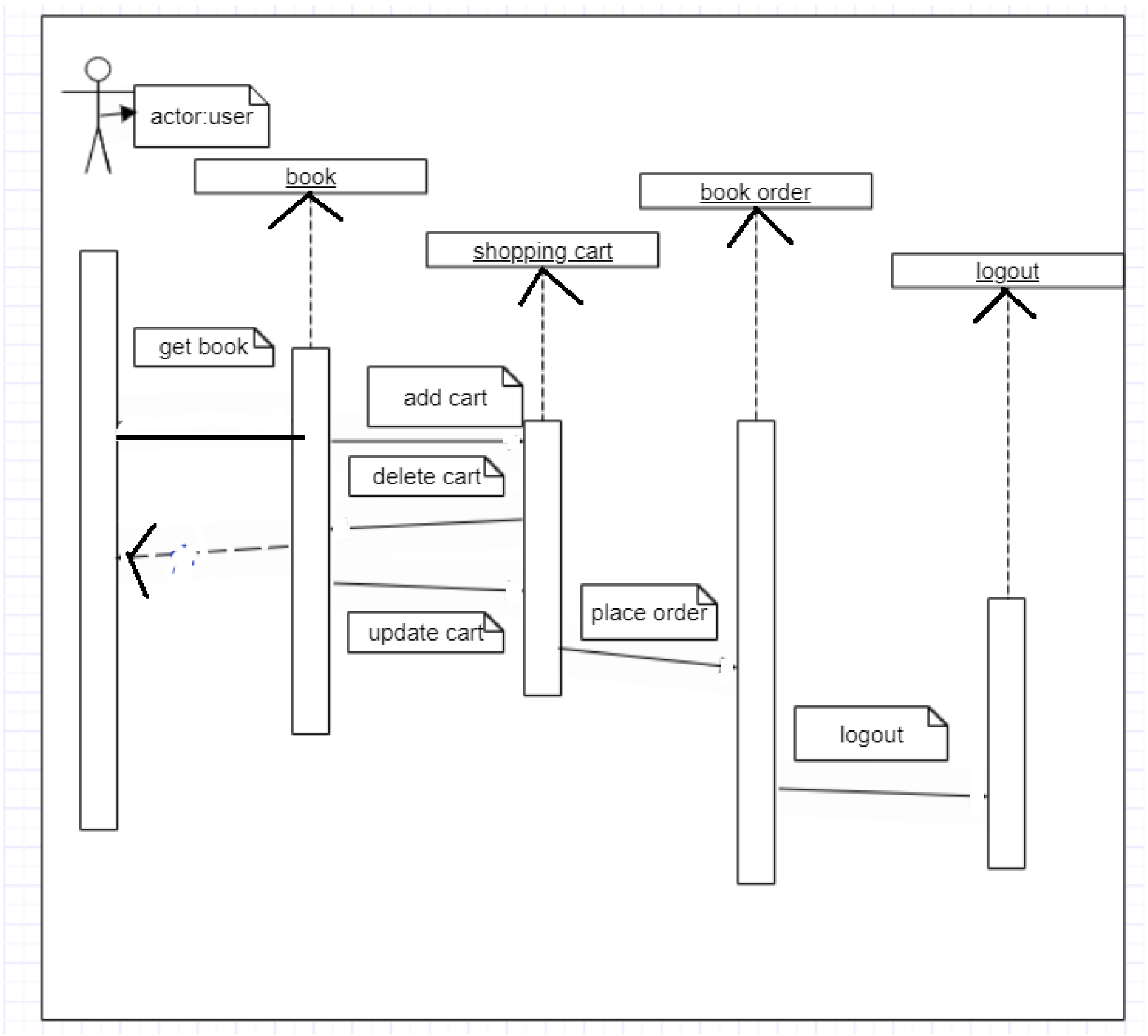
User-login sequence diagram

2) Book search



User book search sequence diagram

3)Add to shopping cart



add to shopping cart sequence diagram

Testing approach

1)Unit testing

Unit testing is a method of testing, testing is done in isolation of an individual unit of source code in to check if the individual unit is working properly, if there is any bugs we can resolve it quickly . The Android emulator is a testing tool used for unit testing**.**

2)System testing

System testing is a approach to test whole system at once. Once the system is completely developed system testing is carried on to test whether the system is working properly or there is any bugs in the system. In system testing resolving bugs is little complex and time consuming.

3)Manual testing

Manual Testing is done to ensure the correctness of various parts of the code using test cases generated by the group.

Test case functionalities for manual testing include:

USER:

• Registration

• Login

• Add To Cart

• Edit Cart

Test cases

Registration

Id	Test case	User Input	Result	Pass Criteria
U-REG-1	Registration of user	If User has selected already existing user name	Pass	Pop-up message to provide different user name
U-REG-2	Registration of user	If the User provides different password in password confirm field	Pass	Pop-up message that Password and Confirm Password fields do not match
U-REG-3	Registration of user	If the User do not enter a particular mandatory fields	Pass	Pop-up message That value in field is required
U-REG-4	Registration of user	If the User provides all the details correctly	Pass	User account will be created

Login

Id	Test case	User Input	Result	Pass Criteria
U-log-1	Login of user	If the User provides incorrect username	Pass	Pop-up message that Login or Password is incorrect.
U-log-2	Login of user	If the User provides incorrect password	Pass	Pop-up message that Login or Password is incorrect.
U-log-3	Login of user	If the User provides both correct username and password	Pass	User logs in successfully

Add to cart

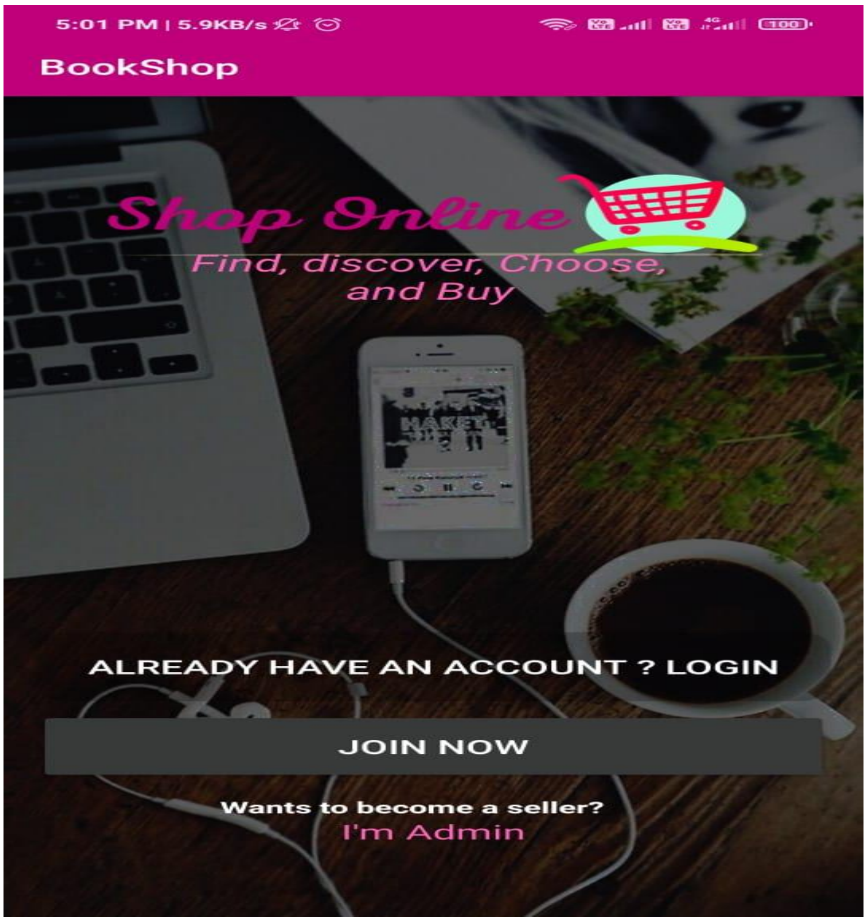
Id	Test case	User Input	Result	Pass Criteria
U-Ac-1	Adding to cart	User selects a book and clicks add to cart button	Pass	Book is added to the shopping cart
UAc-2	Adding to cart	Guest selects a book and clicks add to cart button	Pass	User should create an account

Edit Cart

Id	Test case	User Input	Result	Pass Criteria
U-EC-1	Editing of Cart	User changes the Quantity	Pass	Quantity and total cost of Cart should be updated
U-EC-2	Editing of Cart	User deletes a book from shopping Cart	Pass	Books and total cost of Cart should be updated
U-EC-3	Editing of Cart	User selects a new book to shopping Cart	Pass	Books and total cost of Cart should be updated

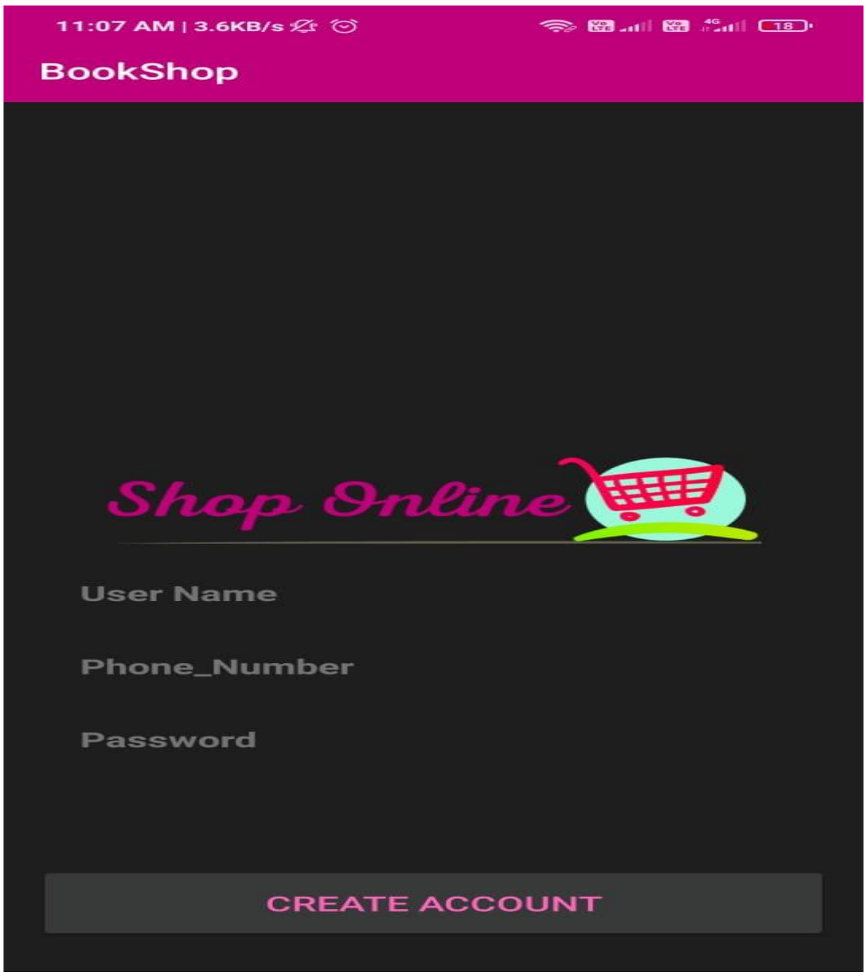
User Manual

When we open the My BookShop Application first login/register page is displayed. The Login/register page will appear as below.


5:01 PM | 5.9KB/s
BookShop
Shop Online
Find, discover, Choose, and Buy
ALREADY HAVE AN ACCOUNT ? LOGIN
JOIN NOW
Wants to become a seller?
I'm Admin

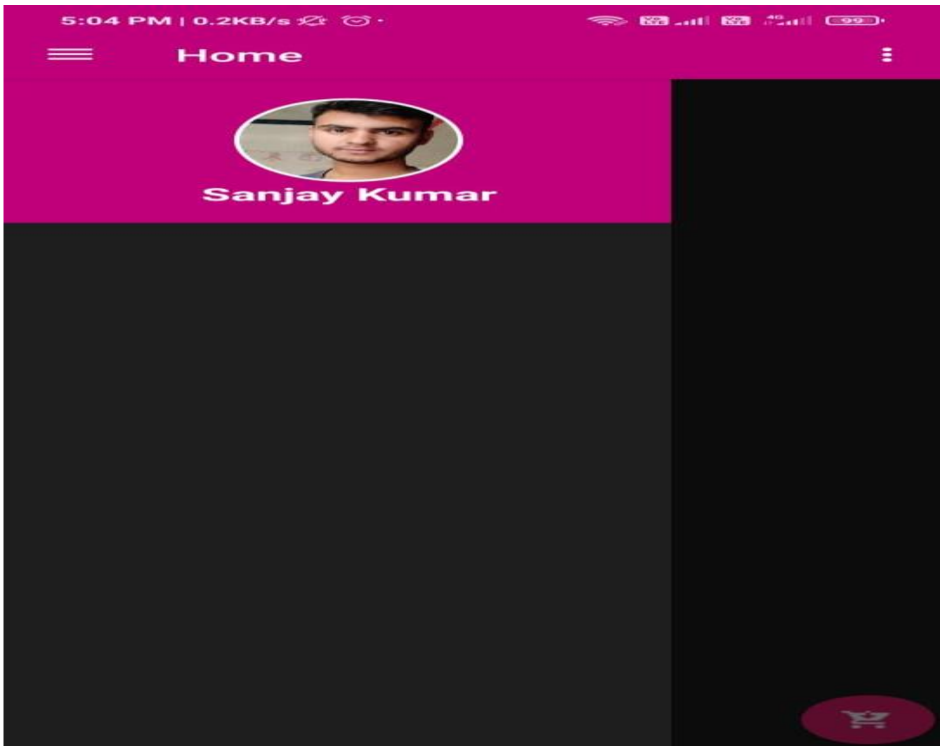
If you are a new user you can register using the Join now Button or if you are already a user you can login to purchase book and pay using the checkout.

User Registration page is shown Below



Once you register and login using username and password

Home page will appear where you can search for the book by the title of the book. Picture below is the Picture of the Home page w here Name and picture of the user is shown



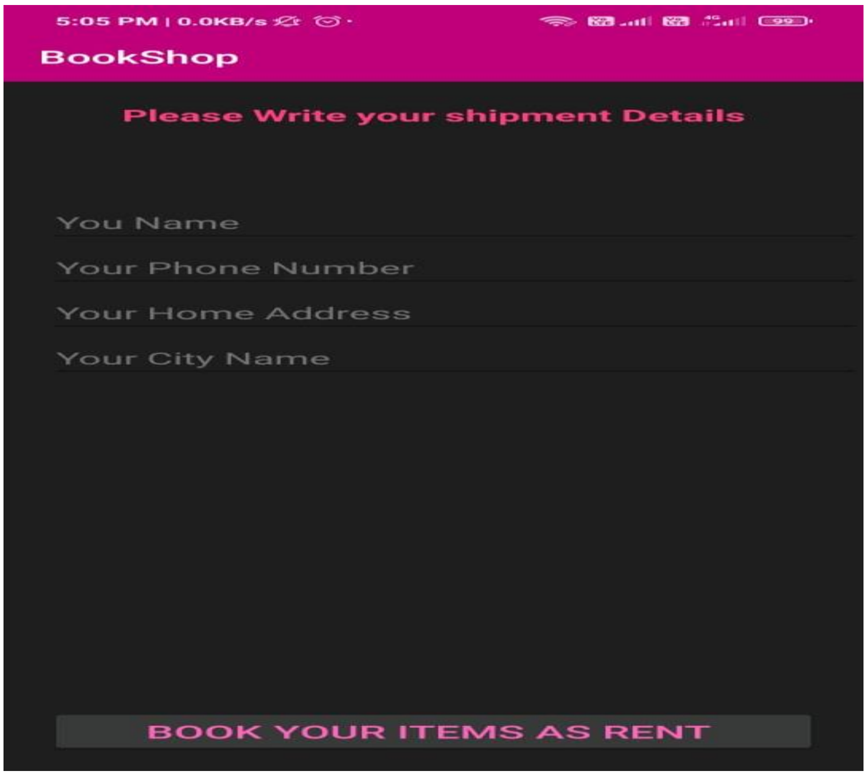
At the right top corner there is menu button by clicking on it, you will get the search button After searching for a book you will get the Description of the book written below the tiltle and price, Tap on the Book and Tap on Add to cart button to get the item in the cart,

Once you added the item in the cart then total Price will be shown

To place order Tap on “Place Order” button

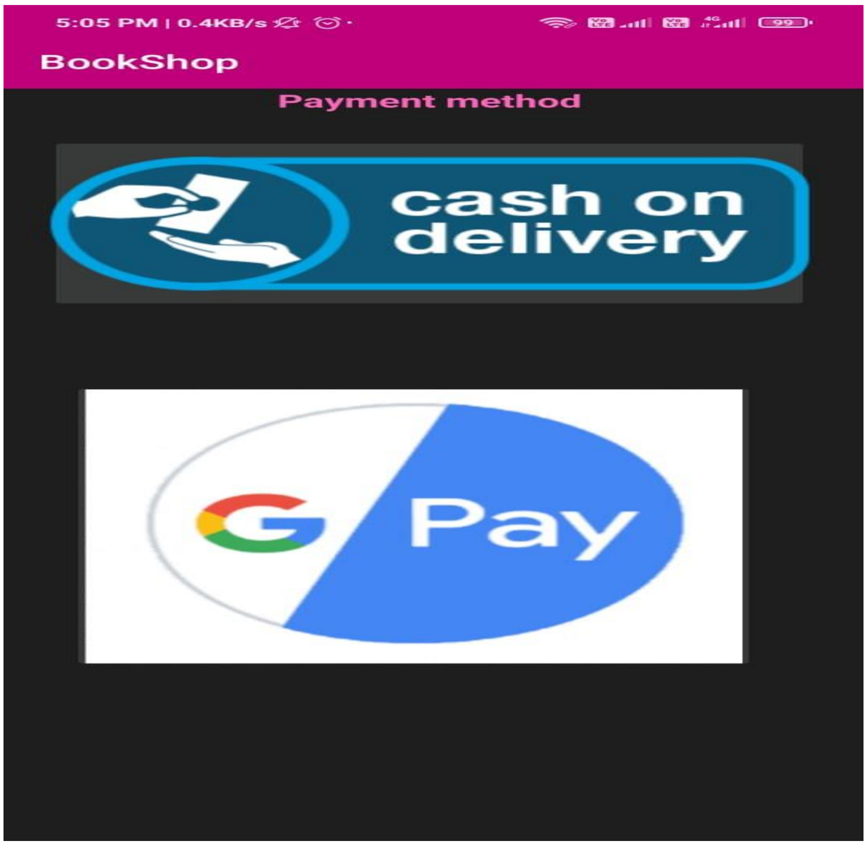
A page will be appeared in fornt of you where you will fill the address to get the item delivered to your address.

Below the picture of the address page



After filling the Adress you get the option to rent the book as well as to purchase the book.

The Payment page will appear after that where you can pay using UPI or cash on delivery



After payment you will get the confirmation of your purchase.

Lesson learned

Programming

The My Bookshop application helped us to improve our confidence level in Android studio, Java Programming, XML and Firebase. Though we have made many mistakes during the initial phase we have learnt how to use Android studio and firebase functionalities.

Time Management

Since MSE Project is done as an group we have learnt how to manage time during the Software Life Cycle Process. we have also learned how to face tense situations and meet the deadlines .This would add as a good experience for us for our future job prospective.

Software Lifecycle

As software student though we have good knowledge in UML and Software LIFE cycle we never had any good practical experience regarding them. Through this project we have learnt how to develop a project following the various stages in Software Life Cycle.

Documentation

we always had a feeling that we are not good at documentation .But through this project and suggestions from my committee members we believe that we have improved our Documentation skills.

Source code

Main Activity code

```
package com.example.bookshop.Buyers;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.example.bookshop.Admin.AdminLoginActivity;

import com.example.bookshop.Model.Users;

import com.example.bookshop.Prevalent.Prevalent;

import com.example.bookshop.R;

import com.example.bookshop.Sellers.SellerRegistrationActivity;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import io.paperdb.Paper;
```

```
public class MainActivity extends AppCompatActivity {
    private Button joinButton,loginButton;
    private ProgressDialog loadingBar;
    private TextView sellerBegin,Adminlayout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        joinButton=findViewById(R.id.main_join_now_btn);
        loginButton=findViewById(R.id.main_login_btn);
        loadingBar=new ProgressDialog(this);
        sellerBegin=findViewById(R.id.seller_begin);
        Adminlayout=findViewById(R.id.admin_panel_link);



        Paper.init(this);


        Adminlayout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this, AdminLoginActivity.class);
                startActivity(intent);
            }
        });
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {


                Intent intent=new Intent(MainActivity.this, LoginActivity.class);
                startActivity(intent);
            }
        });

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


                    Intent intent=new Intent(MainActivity.this, SellerRegistrationActivity.class);
                    startActivity(intent);
              }
        });

        joinButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(MainActivity.this, RegisterActivity.class);
                startActivity(intent);
            }
        });
        String UserPhoneKey=Paper.book().read(Prevalent.UserPhoneKey);
        String UserPasswordKey=Paper.book().read(Prevalent.UserPasswordKey);
```

```
    if (UserPhoneKey!=null && UserPasswordKey!=null)
    {
        if (!TextUtils.isEmpty(UserPhoneKey) && !TextUtils.isEmpty(UserPasswordKey))
        {
            AllowAccess(UserPhoneKey,UserPasswordKey);


            loadingBar.setTitle("Already Logged in");
            loadingBar.setMessage("Please wait...");
            loadingBar.setCanceledOnTouchOutside(false);
            loadingBar.show();
        }
    }
}

private void AllowAccess(final String phone, final String password) {


    final DatabaseReference RootRef;
    RootRef= FirebaseDatabase.getInstance().getReference();
    RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {


            if (snapshot.child("Users").child(phone).exists())
            {


                Users usersData=snapshot.child("Users").child(phone).getValue(Users.class);
```

```
          if (usersData.getPhone().equals(phone))
          {
             if (usersData.getPassword().equals(password))
             {
                Toast.makeText(MainActivity.this,"logged in
successfully",Toast.LENGTH_LONG).show();
                loadingBar.dismiss();
                Intent intent=new Intent(MainActivity.this, HomeActivity.class);
                Prevalent.currentOnlineUser=usersData;
                startActivity(intent);


             }
             else
             {
                loadingBar.dismiss();
                Toast.makeText(MainActivity.this,"Password is
incorrect",Toast.LENGTH_LONG).show();
             }
          }
       }
       else
       {
          Toast.makeText(MainActivity.this,"Account with this"+phone+"do not
exits..",Toast.LENGTH_LONG).show();


          loadingBar.dismiss();
       }
```

```
            }


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


            }
        });
    }
}
```

Home Activity

```
package com.example.bookshop.Buyers;
import android.content.Intent;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.GravityCompat;
import androidx.drawerlayout.widget.DrawerLayout;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.example.bookshop.Admin.AdminMaintainProductActivity;
import com.example.bookshop.Model.Products;
import com.example.bookshop.Prevalent.Prevalent;
import com.example.bookshop.R;
import com.example.bookshop.ViewHolder.ProductViewHolder;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.navigation.NavigationView;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;
import de.hdodenhof.circleimageview.CircleImageView;
import io.paperdb.Paper;

public class HomeActivity extends AppCompatActivity
implements NavigationView.OnNavigationItemSelectedListener{
    private TextView textView;



    private AppBarConfiguration mAppBarConfiguration;
    private DatabaseReference ProductRef;
    private RecyclerView recyclerView;
```

```
RecyclerView.LayoutManager layoutManager;
private String type=" ";

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

    Intent intent=getIntent();
    Bundle bundle=intent.getExtras();
    if (bundle!=null)
    {
        type=getIntent().getExtras().get("Admin").toString();
    }

    Paper.init(this);
    ProductRef= FirebaseDatabase.getInstance().getReference().child("Products");
    recyclerView=findViewById(R.id.recycle_menu);
    recyclerView.setHasFixedSize(true);
    layoutManager=new LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);

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



            }
        });
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);

        View headerView=navigationView.getHeaderView(0);
        textView=headerView.findViewById(R.id.user_profile_name);
        CircleImageView
circleImageView=headerView.findViewById(R.id.user_profile_image);
        if (!type.equals("Admin"))
        {

            textView.setText(Prevalent.currentOnlineUser.getName());

Picasso.get().load(Prevalent.currentOnlineUser.getImage()).placeholder(R.drawable.profile).i
nto(circleImageView);
        }

        mAppBarConfiguration = new AppBarConfiguration.Builder(
                R.id.nav_home, R.id.nav_gallery, R.id.nav_slideshow)
                .setDrawerLayout(drawer)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
    }


    @Override
    protected void onStart() {
        super.onStart();
        FirebaseRecyclerOptions<Products> options=
                new FirebaseRecyclerOptions.Builder<Products>()
                        .setQuery(ProductRef.orderByChild("ProductState").equalTo("Approved"),
Products.class)
                        .build();
        FirebaseRecyclerAdapter<Products, ProductViewHolder> adapter=
                new FirebaseRecyclerAdapter<Products, ProductViewHolder>(options) {
                    @Override
                    protected void onBindViewHolder(@NonNull ProductViewHolder
productViewHolder, int i, @NonNull Products products) {


                        productViewHolder.txtProductName.setText(products.getPname());
                        productViewHolder.txtProductDescrition.setText(products.getDescription());
                        productViewHolder.getTxtProductprice.setText("Price
"+products.getPrice()+"$");
                        Picasso.get().load(products.getImage()).into(productViewHolder.imageView);
                productViewHolder.itemView.setOnClickListener(new
View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        Intent intent;
                        if (type.equals("Admin"))
                        {

                            intent = new Intent(HomeActivity.this,
AdminMaintainProductActivity.class);

                        }
                        else
                        {

                            intent = new Intent(HomeActivity.this, ProductDetailsActivity.class);

                        }
                        intent.putExtra("pid",products.getPid());
                        startActivity(intent);

                    }
                });

            }

            @NonNull
```

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


                }
            };
        recyclerView.setAdapter(adapter);
        adapter.startListening();


    }


    @Override
    public void onBackPressed() {
        super.onBackPressed();
        DrawerLayout drawer=findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START))
        {
            drawer.closeDrawer(GravityCompat.START);
        }
        else
        {
            super.onBackPressed();
        }
```

```
}


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


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


    if (id== R.id.action_settings)
    {
        Intent intent=new Intent(HomeActivity.this, SettingsActivity.class);
        startActivity(intent);


    }
    if (id== R.id.nav_cart)
    {
        if (!type.equals("Admin"))
        {
            Intent intent=new Intent(HomeActivity.this,CartActivity.class);
            startActivity(intent);
        }
        }


        else if (id== R.id.nav_search)
        {
           if (!type.equals("Admin"))
           {

               Intent intent=new Intent(HomeActivity.this, SearchProductActivity.class);
               startActivity(intent);
           }

        }


        else if (id== R.id.nav_logout)
        {
           if (!type.equals("Admin"))
           {

               Paper.book().destroy();
               Intent intent=new Intent(HomeActivity.this, MainActivity.class);

intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR
_TASK);
               startActivity(intent);
               finish();
           }
        }
        return super.onOptionsItemSelected(item);
    }

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

    @Override
    public boolean onNavigationItemSelected(MenuItem item)
    {
        int id=item.getItemId();
        return true;
    }
    @Override
    public void onPointerCaptureChanged(boolean hasCapture) {

    }
}
```

User Login

```
package com.example.bookshop.Buyers;

import android.app.ProgressDialog;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.example.bookshop.Model.Users;

import com.example.bookshop.Prevalent.Prevalent;

import com.example.bookshop.R;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import com.rey.material.widget.CheckBox;

import io.paperdb.Paper;


public class LoginActivity extends AppCompatActivity {
```

```
private EditText InputPhoneNumber,InputPassword;
private Button LoginButton;
private ProgressDialog loadingBar;
private String parentDbName="Users";
private CheckBox chkBoxRememberMe;
private TextView ForgotPasswordLink;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    LoginButton=findViewById(R.id.login_btn);
    InputPhoneNumber=findViewById(R.id.login_phone_number_input);
    InputPassword=findViewById(R.id.login_password_input);
    loadingBar=new ProgressDialog(this);
    chkBoxRememberMe=findViewById(R.id.remember_me_chkb);
    ForgotPasswordLink=findViewById(R.id.forgot_password_link);

    Paper.init(this);
    LoginButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            LoginUser();
        }
    });
```

```
        ForgotPasswordLink.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent=new Intent(LoginActivity.this, ResetPasswordActivity.class);
                intent.putExtra("check","login");
                startActivity(intent);
            }
        });
    }
    private void LoginUser() {
        String phone=InputPhoneNumber.getText().toString();
        String password=InputPassword.getText().toString();

        if (TextUtils.isEmpty(phone))
        {
            Toast.makeText(this,"Please write your phone
Number...",Toast.LENGTH_LONG).show();
        }
        else if (TextUtils.isEmpty(password))
        {
            Toast.makeText(this,"Please write your
Password...",Toast.LENGTH_LONG).show();
        }
        else
        {
            loadingBar.setTitle("Login Account");
            loadingBar.setMessage("Please wait, while we are checking the credentials");
            loadingBar.setCanceledOnTouchOutside(false);
            loadingBar.show();
            AllowAccessToAccount(phone,password);
        }
    }
    private void AllowAccessToAccount(String phone, String password)
    {
        if (chkBoxRememberMe.isChecked())
        {
            Paper.book().write(Prevalent.UserPhoneKey,phone);
            Paper.book().write(Prevalent.UserPasswordKey,password);
        }
        final DatabaseReference RootRef;
        RootRef= FirebaseDatabase.getInstance().getReference();
        RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.child(parentDbName).child(phone).exists())
                {
                    Users
usersData=snapshot.child(parentDbName).child(phone).getValue(Users.class);
                    if (usersData.getPhone().equals(phone))
                    {
                        if (usersData.getPassword().equals(password))
                        {
                            Toast.makeText(LoginActivity.this,"logged in
successfully",Toast.LENGTH_LONG).show();
```

```
                        loadingBar.dismiss();
                        Intent intent=new Intent(LoginActivity.this, HomeActivity.class);
                         Prevalent.currentOnlineUser=usersData;
                        startActivity(intent);
                    }
                    else
                    {
                        loadingBar.dismiss();
                        Toast.makeText(LoginActivity.this,"Password is
incorrect",Toast.LENGTH_LONG).show();

                    }
                }
            }
            else
            {
                Toast.makeText(LoginActivity.this,"Account with this"+phone+"do not
exits..",Toast.LENGTH_LONG).show();
                loadingBar.dismiss();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
  }
}
```

User Register

```
package com.example.bookshop.Buyers;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.example.bookshop.R;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.HashMap;

public class RegisterActivity extends AppCompatActivity {
    private Button CreateAccountButtom;
    private EditText InputName,InputPhoneNumber,InputPassword;
private ProgressDialog loadingBar;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_register);
    CreateAccountButtom=findViewById(R.id.register_btn);
    InputName=findViewById(R.id.register_name_input);
    InputPhoneNumber=findViewById(R.id.register_phone_number_input);
    InputPassword=findViewById(R.id.register_password_input);
    loadingBar=new ProgressDialog(this);
    CreateAccountButtom.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            CreateAccount();
        }
    });
}
private void CreateAccount() {
    String name=InputName.getText().toString();
    String phone=InputPhoneNumber.getText().toString();
    String password=InputPassword.getText().toString();

    if (TextUtils.isEmpty(name))
    {
        Toast.makeText(this,"Please write your name...",Toast.LENGTH_LONG).show();
    }
```

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

            validatephoneNumber(name,phone,password);
        }
    }
    private void validatephoneNumber(String name,String phone,String password)
    {

        final DatabaseReference RootRef;
        RootRef= FirebaseDatabase.getInstance().getReference();

        RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
```

```
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if(!(snapshot.child("Users").child(phone).exists()))
            {

                HashMap<String, Object> userdataMap=new HashMap<>();
                userdataMap.put("phone",phone);
                userdataMap.put("password",password);
                userdataMap.put("name",name);


RootRef.child("Users").child(phone).updateChildren(userdataMap).addOnCompleteListener(
new OnCompleteListener<Void>() {
                    @Override
                    public void onComplete(@NonNull Task<Void> task) {
                        if (task.isSuccessful())
                        {
                            Toast.makeText(RegisterActivity.this,"Congratulations, your account
created ",Toast.LENGTH_LONG).show();
                            loadingBar.dismiss();

                            Intent intent=new Intent(RegisterActivity.this, LoginActivity.class);
                            startActivity(intent);

                        }
                        else
                        {
                            loadingBar.dismiss();
                            Toast.makeText(RegisterActivity.this,"Network
issues???",Toast.LENGTH_LONG).show();


                        }
                    }
                });
            }
            else
            {
                Toast.makeText(RegisterActivity.this,"This"+phone+ "is already
Exits",Toast.LENGTH_LONG).show();
                loadingBar.dismiss();
                Toast.makeText(RegisterActivity.this,"Please try another phone
number",Toast.LENGTH_LONG).show();


                Intent intent=new Intent(RegisterActivity.this,LoginActivity.class);
                startActivity(intent);


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


        }
    });
}}
```

AdminLogin:

```
package com.example.bookshop.Admin;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.example.bookshop.Model.Users;
import com.example.bookshop.R;
import com.example.bookshop.Sellers.SellerProductCategoryActivity;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import io.paperdb.Paper;

public class AdminLoginActivity extends AppCompatActivity {
    private EditText InputPhoneNumber,InputPassword;
    private Button LoginButton;
    private ProgressDialog loadingBar;
    private String parentDbName="Admin";
    private TextView AdminRegisterText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_login);
        LoginButton=findViewById(R.id.Admin_login_btn);
        InputPhoneNumber=findViewById(R.id.Admin_login_phone_number_input);
        InputPassword=findViewById(R.id.Admin_login_password_input);
        loadingBar=new ProgressDialog(this);
        AdminRegisterText=findViewById(R.id.Admin_register_Activity);
        AdminRegisterText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(AdminLoginActivity.this, AdminRegisterActivity.class);
                startActivity(intent);
            }
        });

        Paper.init(this);
        LoginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                LoginUser();
            }
        });
    }

    private void LoginUser() {
        String phone=InputPhoneNumber.getText().toString();
        String password=InputPassword.getText().toString();
        if (TextUtils.isEmpty(phone))
        {
            Toast.makeText(this,"Please write your phone
Number...",Toast.LENGTH_LONG).show();
        }
        else if (TextUtils.isEmpty(password))
        {
            Toast.makeText(this,"Please write your
Password...",Toast.LENGTH_LONG).show();
        }
        else
        {
            loadingBar.setTitle("Login Account");
            loadingBar.setMessage("Please wait, while we are checking the credentials");
            loadingBar.setCanceledOnTouchOutside(false);
            loadingBar.show();

            AllowAccessToAccount(phone,password);
        }
    }
    private void AllowAccessToAccount(String phone, String password) {
        final DatabaseReference RootRef;
        RootRef= FirebaseDatabase.getInstance().getReference();
        RootRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.child(parentDbName).child(phone).exists())
                {
                    Users
usersData=snapshot.child(parentDbName).child(phone).getValue(Users.class);

                    if (usersData.getPhone().equals(phone))
                    {
                        if (usersData.getPassword().equals(password))
                        {
                            Toast.makeText(AdminLoginActivity.this,"logged in
successfully",Toast.LENGTH_LONG).show();
                            loadingBar.dismiss();
                            Intent intent=new Intent(AdminLoginActivity.this,
AdminHomeActivity.class);
                            startActivity(intent);
                        }
                        else
                        {
                            loadingBar.dismiss();
                            Toast.makeText(AdminLoginActivity.this,"Password is
incorrect",Toast.LENGTH_LONG).show();


                        }
                    }
                }
                else
                {
                    Toast.makeText(AdminLoginActivity.this,"Account with this"+phone+"do not
exits..",Toast.LENGTH_LONG).show();
                    loadingBar.dismiss();
                }
            }

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

SellerAddNewProduct

```
package com.example.bookshop.Sellers;

import android.app.ProgressDialog;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import com.example.bookshop.R;

import com.google.android.gms.tasks.Continuation;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.HashMap;


public class  SellerAddNewProductActivity extends AppCompatActivity {

    private String CategoryName,
Description,Price,Pname,saveCurrentDate,saveCurrentTime;

    private Button AddNewProductButton;

    private ImageView InputProductImage;

    private Uri ImageUri;

    private  static final int GalleryPick=1;

    private String productRandomKey,downloadImageUrl;

    private StorageReference ProductImageRef;

    private ProgressDialog loadingBar;

    private DatabaseReference ProductsRef,sellersRef;

    private String sName,sAddress,sPhone,sEmail,sID;

    private EditText InputProductName,InputProductDescription,InputProductPrice;


    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_seller_add_new_product);
        CategoryName = getIntent().getExtras().get("category").toString();
        loadingBar=new ProgressDialog(this);
        ProductsRef= FirebaseDatabase.getInstance().getReference().child("Products");
        sellersRef= FirebaseDatabase.getInstance().getReference().child("Seller");
        ProductImageRef= FirebaseStorage.getInstance().getReference().child("Product
Images");
        AddNewProductButton=findViewById(R.id.add_new_product);
        InputProductImage=findViewById(R.id.select_product_image);
        InputProductName=findViewById(R.id.Product_name);
        InputProductDescription=findViewById(R.id.Product_description);
        InputProductPrice=findViewById(R.id.Product_price);
        InputProductImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                OpenGallery();
            }
        });
        AddNewProductButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ValidateProductData();
            }
        });
        sellersRef.child(FirebaseAuth.getInstance().getCurrentUser().getUid())
.addValueEventListener(new ValueEventListener() {
```

```
                @Override
                public void onDataChange(@NonNull DataSnapshot snapshot) {
                    if (snapshot.exists())
                    {
                        sName=snapshot.child("name").getValue().toString();
                        sAddress=snapshot.child("address").getValue().toString();
                        sPhone=snapshot.child("phone").getValue().toString();
                        sID=snapshot.child("sid").getValue().toString();
                        sEmail=snapshot.child("email").getValue().toString();
                    }
                }
                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                }
            });
    }
    private void ValidateProductData()
    {
        Description = InputProductDescription.getText().toString();
        Price=InputProductPrice.getText().toString();
        Pname=InputProductName.getText().toString();
        if (ImageUri==null)
        {
            Toast.makeText(this,"Product Image is
mandatory...",Toast.LENGTH_LONG).show();
        }
        else if (TextUtils.isEmpty(Description))
        {
           Toast.makeText(this,"Please write product
description...",Toast.LENGTH_LONG).show();
        }
        else if (TextUtils.isEmpty(Price))
        {
           Toast.makeText(this,"Please write product Price...",Toast.LENGTH_LONG).show();
        }
        else if (TextUtils.isEmpty(Pname))
        {
           Toast.makeText(this,"Please write product Name...",Toast.LENGTH_LONG).show();
        }
        else
        {
           StoreProductInformation();
        }
    }
    private void StoreProductInformation()
    {
        loadingBar.setTitle("Adding New Product...");
        loadingBar.setMessage("Dear Seller, Please wait, while we are checking the
credentials");
        loadingBar.setCanceledOnTouchOutside(false);
        loadingBar.show();
        Calendar calendar=Calendar.getInstance();
        SimpleDateFormat currentDate=new SimpleDateFormat("MMM dd, yyyy");
        saveCurrentDate=currentDate.format(calendar.getTime());
        SimpleDateFormat currentTime=new SimpleDateFormat("HH:mm:ss a");
        saveCurrentTime=currentTime.format(calendar.getTime());
        productRandomKey=saveCurrentDate+saveCurrentTime;
        StorageReference
filePath=ProductImageRef.child(ImageUri.getLastPathSegment()+productRandomKey
+".jpg" );
        final UploadTask uploadTask=filePath.putFile(ImageUri);
        uploadTask.addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                String message=e.toString();
                Toast.makeText(SellerAddNewProductActivity.this,"Error :
"+message,Toast.LENGTH_SHORT).show();
                loadingBar.dismiss();
            }
        }).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
            @Override
            public void onSuccess(UploadTask.TaskSnapshot taskSnapshot)
            {
                Toast.makeText(SellerAddNewProductActivity.this,"Image Uploaded
Successfully...",Toast.LENGTH_SHORT).show();

                Task<Uri> uriTask=uploadTask.continueWithTask(new
Continuation<UploadTask.TaskSnapshot, Task<Uri>>() {
                    @Override
                    public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task)
throws Exception {
                    if (!task.isSuccessful())
                    {
                        throw task.getException();
                    }
                    downloadImageUrl=filePath.getDownloadUrl().toString();
                    return filePath.getDownloadUrl();
                    }
                }).addOnCompleteListener(new OnCompleteListener<Uri>() {
                    @Override
                    public void onComplete(@NonNull Task<Uri> task) {
                    if (task.isSuccessful())
                    {
                        downloadImageUrl=task.getResult().toString();
                        Toast.makeText(SellerAddNewProductActivity.this, "got Product Image
save to DataBase Successfully...", Toast.LENGTH_SHORT).show();
                        SaveProductInfoToDatabase();
                    }
                    }
                });
            }
        });
}

private void SaveProductInfoToDatabase()
{
        HashMap<String, Object> productMap= new HashMap<>();
        productMap.put("pid",productRandomKey);
        productMap.put("date",saveCurrentDate);
        productMap.put("time",saveCurrentTime);
        productMap.put("description",Description);
        productMap.put("image",downloadImageUrl);
        productMap.put("category",CategoryName);
        productMap.put("price",Price);
        productMap.put("pname",Pname);
        productMap.put("sellerName",sName);
        productMap.put("sellerAddress",sAddress);
        productMap.put("sellerPhone",sPhone);
        productMap.put("sellerEmail",sEmail);
        productMap.put("sid",sID);
        productMap.put("ProductState","Not Approved" );
ProductsRef.child(productRandomKey).updateChildren(productMap).addOnCompleteListen
er(new OnCompleteListener<Void>() {
            @Override
            public void onComplete(@NonNull Task<Void> task) {
                if (task.isSuccessful())
                {
                    Intent intent=new Intent(SellerAddNewProductActivity.this,
SellerHomeActivity.class);
                    startActivity(intent);
                    loadingBar.dismiss();
                    Toast.makeText(SellerAddNewProductActivity.this,"Product is added
successfully",Toast.LENGTH_SHORT).show();
```

```
            }
            else
            {
                loadingBar.dismiss();
                String message=task.getException().toString();
                Toast.makeText(SellerAddNewProductActivity.this,"Error
"+message,Toast.LENGTH_SHORT).show();
            }
        }
    });
}
private void OpenGallery() {
    Intent galleryIntent=new Intent();
    galleryIntent.setAction(Intent.ACTION_GET_CONTENT);
    galleryIntent.setType("image/*");
    startActivityForResult(galleryIntent,GalleryPick);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode==GalleryPick && resultCode==RESULT_OK && data!=null)
    {
        ImageUri =data.getData();
        InputProductImage.setImageURI(ImageUri);
    }
}
```

REFERENCES

Jie Liu (2015). Design and Implementation of Online Bookstore Based on JSP and JavaBean Technology. Modern Information.

Gupta, A. (2014, January). E-Commerce: Role Of ECommerce In Today's Business. International Journal of Computing and Corporate Research

1. https://en.wikipedia.org/wiki/Android_Studio
2. https://firebase.google.com/docs/database/android/read-and-write
3. https://android-developers.googleblog.com/2020/10/android-studio-41.html
4. https://developer.android.com/studio/run/emulator
5. https://developer.android.com/studio/releases/gradle-plugin
6. https://developer.android.com/studio/releases/platform-tools
7. https://developer.android.com/
8. https://firebase.google.com/
9. https://firebase.flutter.dev/docs/auth/usage/
10. https://www.geeksforgeeks.org/how-to-populate-recyclerview-with-firebase-data-using-firebaseui-in-android-studio/