VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

BACHUPALLY(SO), NIZAMPET,HYDERABAD-5000090.

LABORATORY RECORD

TAMASOMA JYOTIRGAMAYA

SUBJECT: SOFTWARE ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

NAME:	**N. Akhil**	**YEAR/Sem :**	**II/II**
ROLL NUMBER:	**19071A12C1**	**SECTION:**	**IT-3**

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

BACHUPALLY(SO), NIZAMPET,HYDERABAD-5000090.

CERTIFICATE

Certified that is the bonafide record of the practical work done during the Academic Year **2020-2021** by the Student Name N.AKHIL Hall Ticket No 19071A12C1 Section 3 in the Laboratory **of SOFTWARE ENGINEERING** Department of **INFORMATION TECHNOLOGY .**

Signature of the Head of the Dept. Signature of the Staff Member

Date of Examination :

Signature of the Examiners:

Internal Examiner External Examine

COURSE OBJECTIVES:

• To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

List of Sample Projects :

1. Passport automation System,
2. Book Bank ,
3. Online Exam Registration,
4. Stock Maintenance System,
5. Online course reservation system,
6. E-ticketing ,
7. Software Personnel Management System,
8. Credit Card Processing ,
9. E-book management System,
10. Recruitment system .

EXNO : 1a E-BOOK MANAGEMENT SYSTEM

Aim :

To develop the E-Book Management System using rational software .

Project Description :

This project provides an online interface to the customers purchasing the necessary books .It was implemented in order to reduce the workload of customers searching out at the book stores daily and finally not getting it .

Initially it requests for login details which include customers id and his/her password , then the customer can look into the availability of book by filtering with genres , book name /author name . Here this project is developed such that not only experienced users can handle but also new and naïve users as well .

After searching for that required book there would be two cases :

Case 1 : If the book is available ,there's an option to fix the quantity ,include any other book written by the same author also the option which redirects to the payment section .

Case 2 : If the book us not available there's an option to getting notified by giving our email address hence the system/site notifies when the book is available and further case 1 is followed then .

The system also takes the feedback in case of any damaged book is received/feeling giving complaints in order to improve , such that the administrator will get idea on what areas to be improved .

Problem Statement :

1. Enter your valid id and password in order to get access into the system .
2. Select the author name /book genres in which you want to make a purchase .
3. If found have a sample review and purchase it .

SOFTWARE REQUIREMENTS SPECIFICATION

SNO	SOFTWARE REQUIREMENTS SPECIFICATION
1.0	Introduction
1.1	Purpose
1.2	Scope
1.3	Definition, Acronyms and Abbreviations
1.4	Reference
1.5	Technology to be used
1.6	Tools to be used
1.7	Overview
1.8	Overall description
1.9	Productive description
2.0	Software interface
2.1	Hardware interface
2.2	System function
2.3	User Characteristic
2.4	Constraints
2.5	Assumption and Dependences

1.0 INTRODUCTION :

E – Book Management System is an interface between the existing database of information and customers purchasing (accessing) . It's aim to improve the efficiency if issuing a required book and also to reduce the complexities involved in it to maximum possible event .

1.1 PURPOSE :

Manually the customer has to roam about number of book shops in order to find his/her required book which could take several amount of time and in worst case he/she may not find the book . Considering the fact that the Automated System are essentially in demand now-a-days . So, this system uses several programming and database technique in order to carry out the process.

1.2 SCOPE :

The Software provides customer or the administrator a easy and efficient way to buy and manage books online . Not only facilitates easy shopping , provides e-mail facility for future improvements also provides backup facility (as like recoverable in databases) .

1.3 DEFINITIONS , ACRONYMS AND THE ABBREVIATIONS :

1. Administrator – refers to the super user who manages the entire system .
2. Registered User – one who wishes to buy a book through his validated Registered id .
3. EBMS – refers to E-Book Management System .

1.4 REFERENCES :

IEEE Software requirement specification format .

1.5 TECHNOLOGIES TO BE USED :

HTML , TCP/IP, JSP , Java .

1.6 TOOLS TO BE USED :

Oracle Web Server , javac ,Visual Studio Code , Rationale Rose Tool .

1.7 OVERVIEW :

SRS includes two sections overall description and specific requirements . Overall description will describe major role of the system components and interconnections. Specific requirements will describe roles and functions of the actors .

1.8 OVERALL DESCRIPTION :

1.9 PRODUCT PERSPECTIVE :

The EBMS acts as an interface between the ‘customer’ and the ‘administrator’ . This system tries to make an interface as simple as possible at the same time not risking the data security / data redundancy stored in . This minimizes the time duration in which the customer could purchase a book online.

2.0 SOFTWARE INTERFACE :

1. Front End Client - the customer and administrator online is rational rose tool .

2.Back End - Oracle .

2.1 HARDWARE INTERFACES :

The server is directly connected to the client system . The client system have access to the database in the server .

2.2 SYSTEM FUNCTIONS :

1. Secure registration of information by the customers and also storing the info already registered .

2. Message box for Books purchased and to be notified Status Display by the administrator .

3. Administrator can generate reports from the information and can add other eligible person to give users the database in case of any emergency .

2.3 USER CHARACTERISTICS :

1. Customer - The people who desires to purchase a book online and get registered to any future purchases .

2. Administrator – Person having certain privileges to add the customer's status and to approve this issue of purchased book . He may contain a group of persons under him to take in customer's suggestions , verifying the new stock and giving suggestions to whether or not payment has done .

2.4 CONSTRAINTS :

1. The customers required an electronic device connected to internet to submit their information .

2. The user has to be careful while submitting the information .

3. Although the security is given high importance in order to have safe transactions as there's a chance of intrusion while money is transacted.

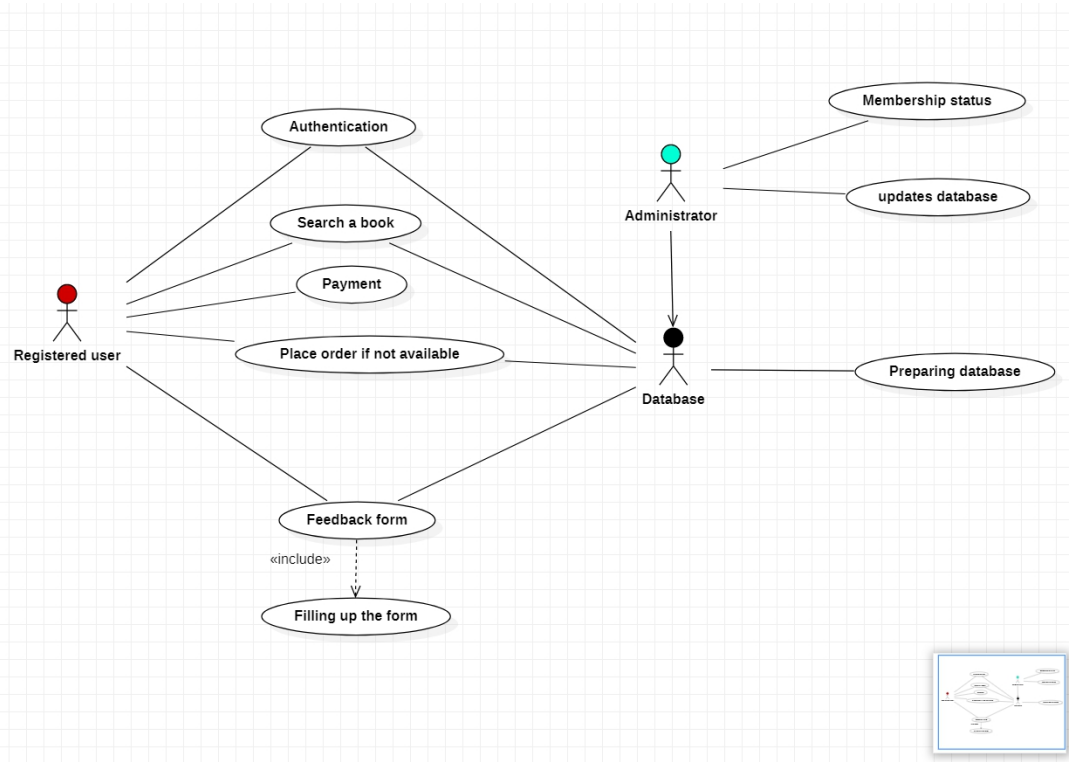
2.5 ASSUMPTIONS AND DEPENDENCIES :

1. The administrator as well as the customer must have basic knowledge of computer and English language .

2. The administrator may send the code to verify the source of information .

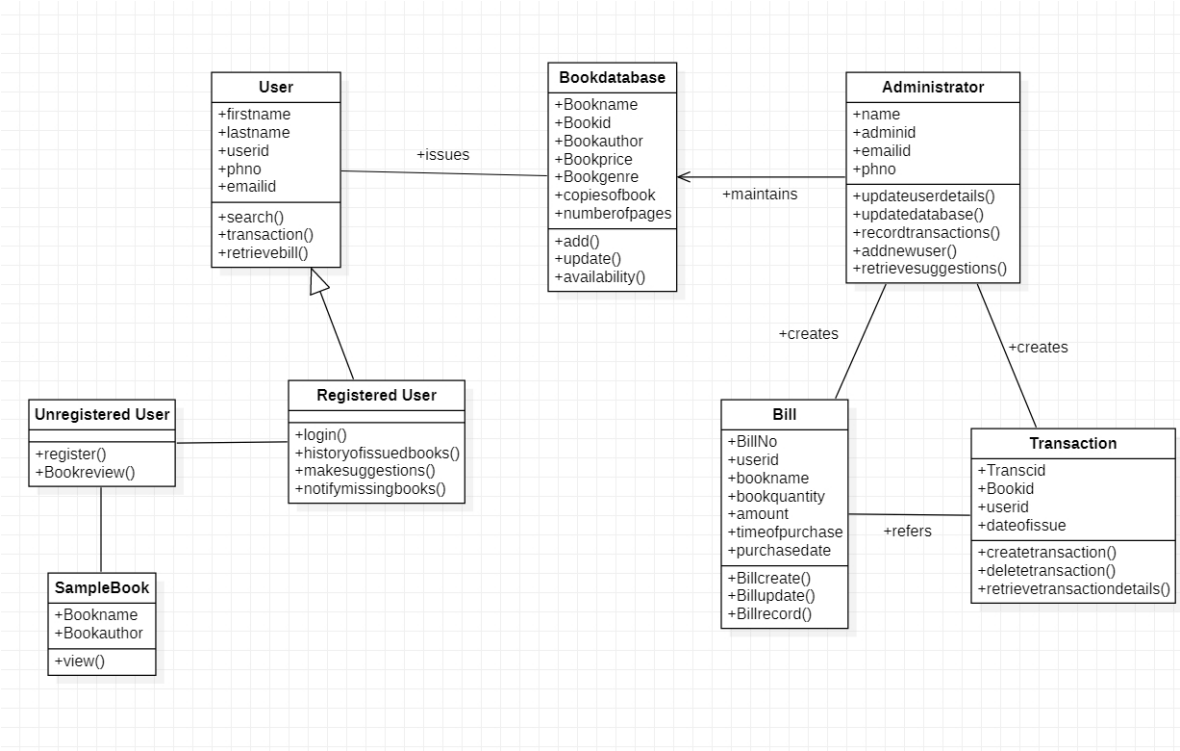
UMLDIAGRAMS :

1. Use Case Diagram :



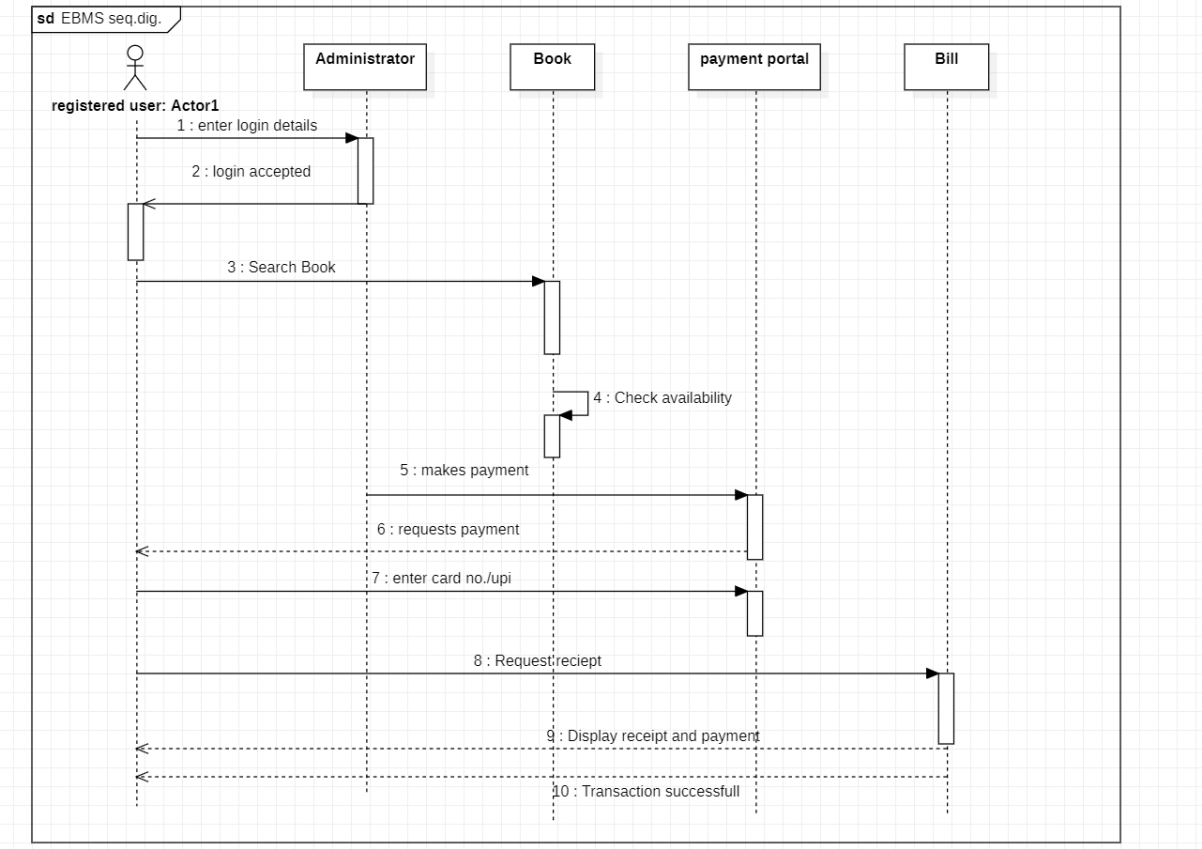
The **use cases** and actors in **use-case diagrams** describe what the system does and how the actors **use** it, but not how the system operates internally. **Use-case diagrams** illustrate and define the context and requirements of either an entire system or the important parts of the system.

2. Class Diagram :



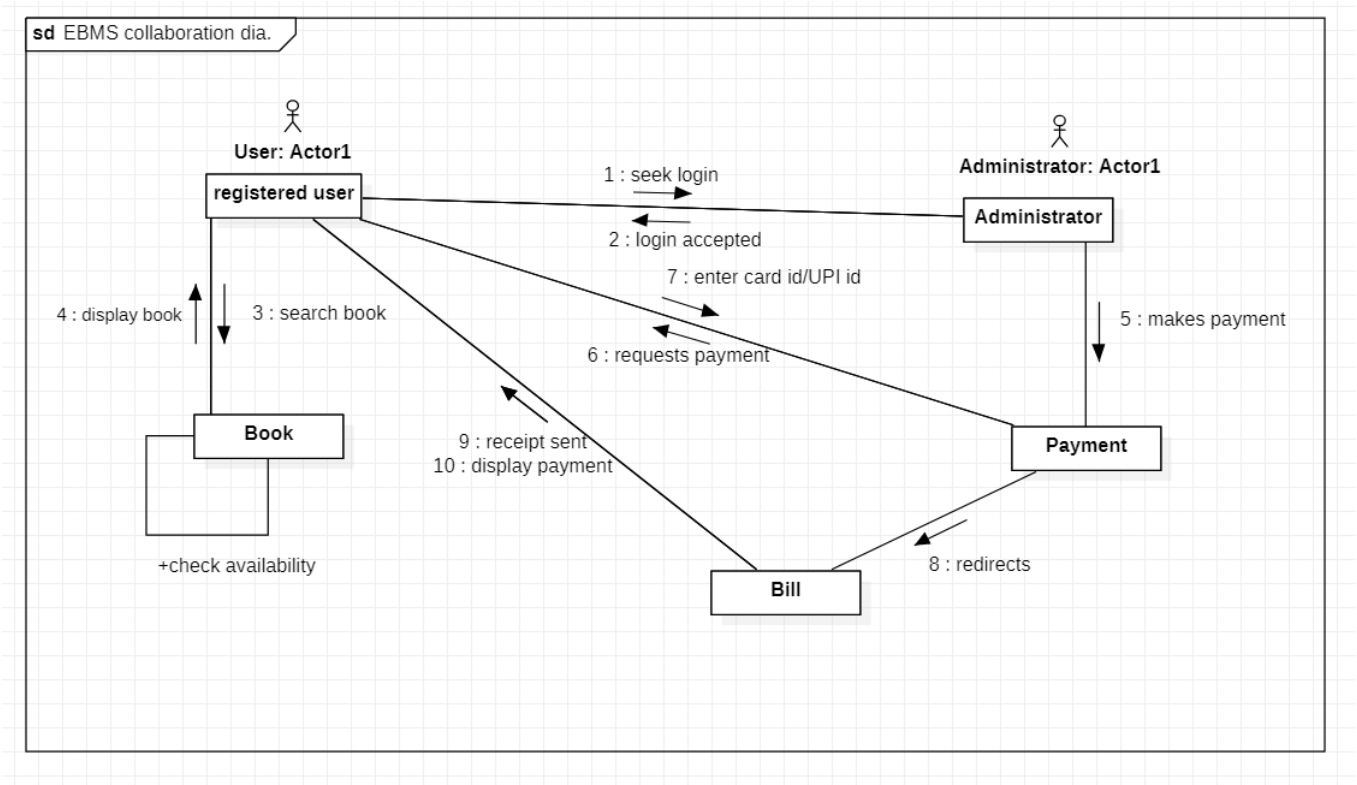
Class diagram is a static **diagram**. ... The **class diagrams** are widely used in the modelling of object-oriented systems because they are the only UML **diagrams**, which can be mapped directly with object-oriented languages. **Class diagram** shows a collection of **classes**, interfaces, associations, collaborations, and constraints.

3. Sequence Diagram :



A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

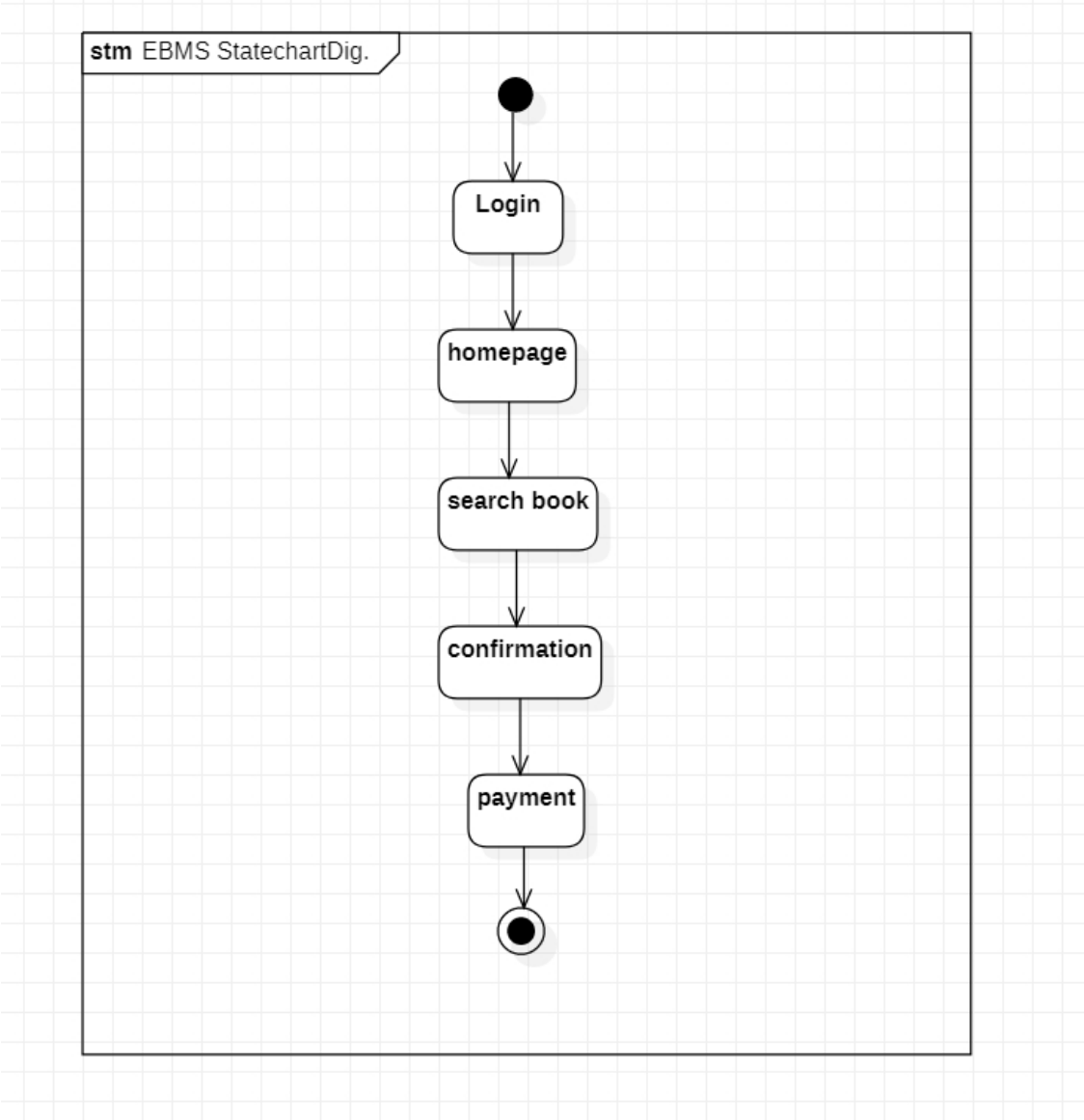
4. Collaboration Diagram :



-

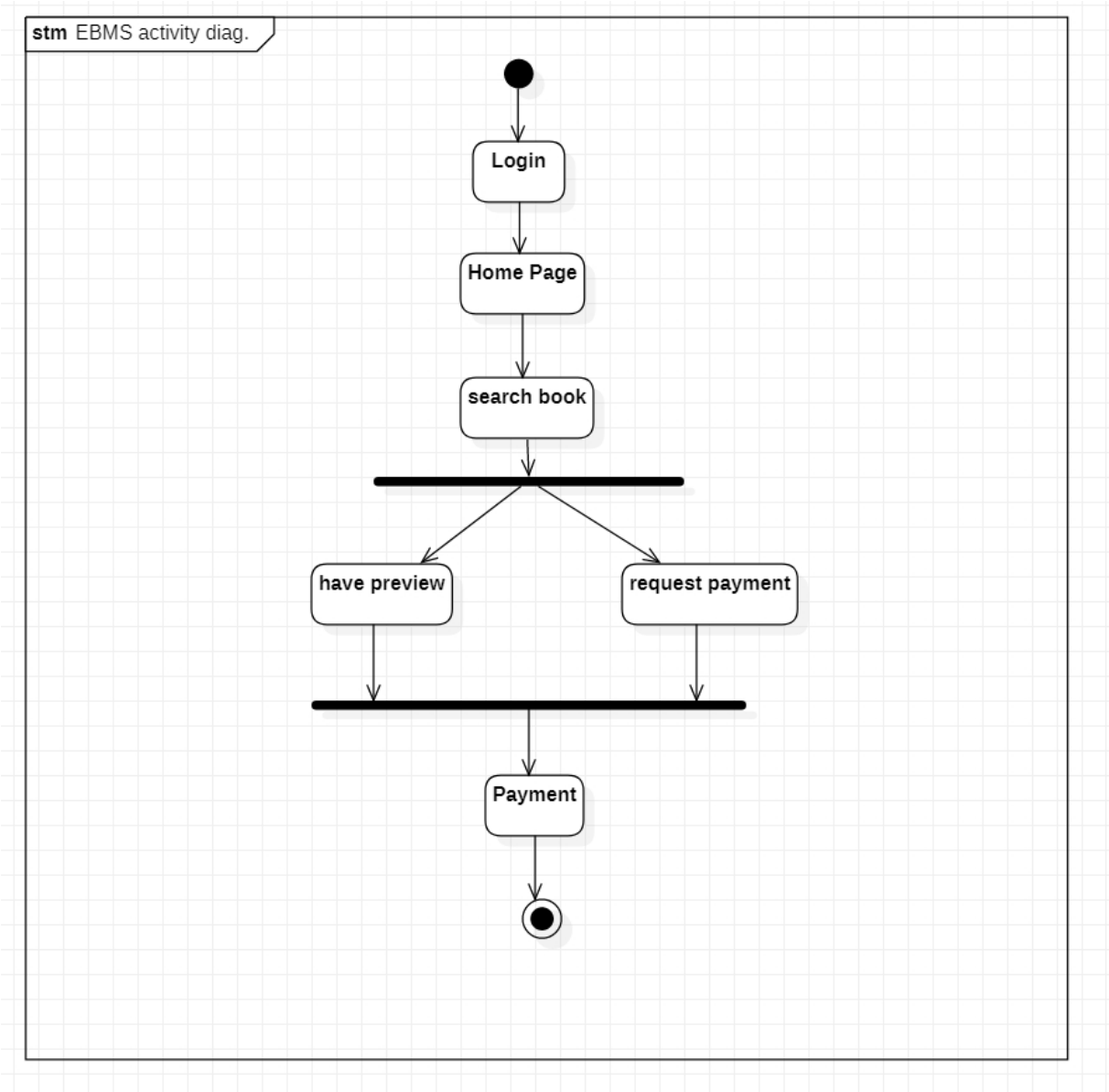
A **collaboration diagram**, also known as a communication **diagram**, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These **diagrams** can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

5. State Chart Diagram :



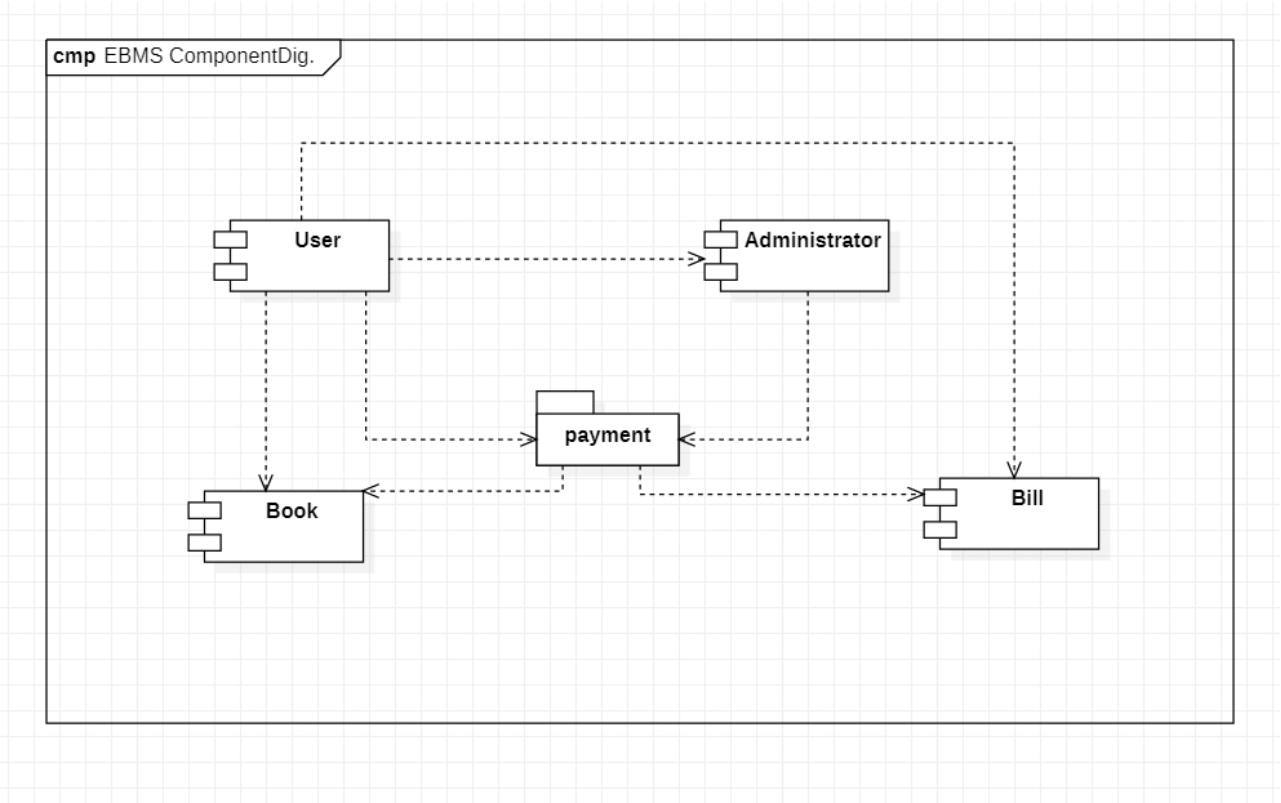
State chart diagram describes the flow of control from one **state** to another **state**. **States** are **defined** as a condition in which an object exists and it changes when some event is triggered. The most important purpose of **State chart diagram** is to model lifetime of an object from creation to termination.

6. Activity Diagram :



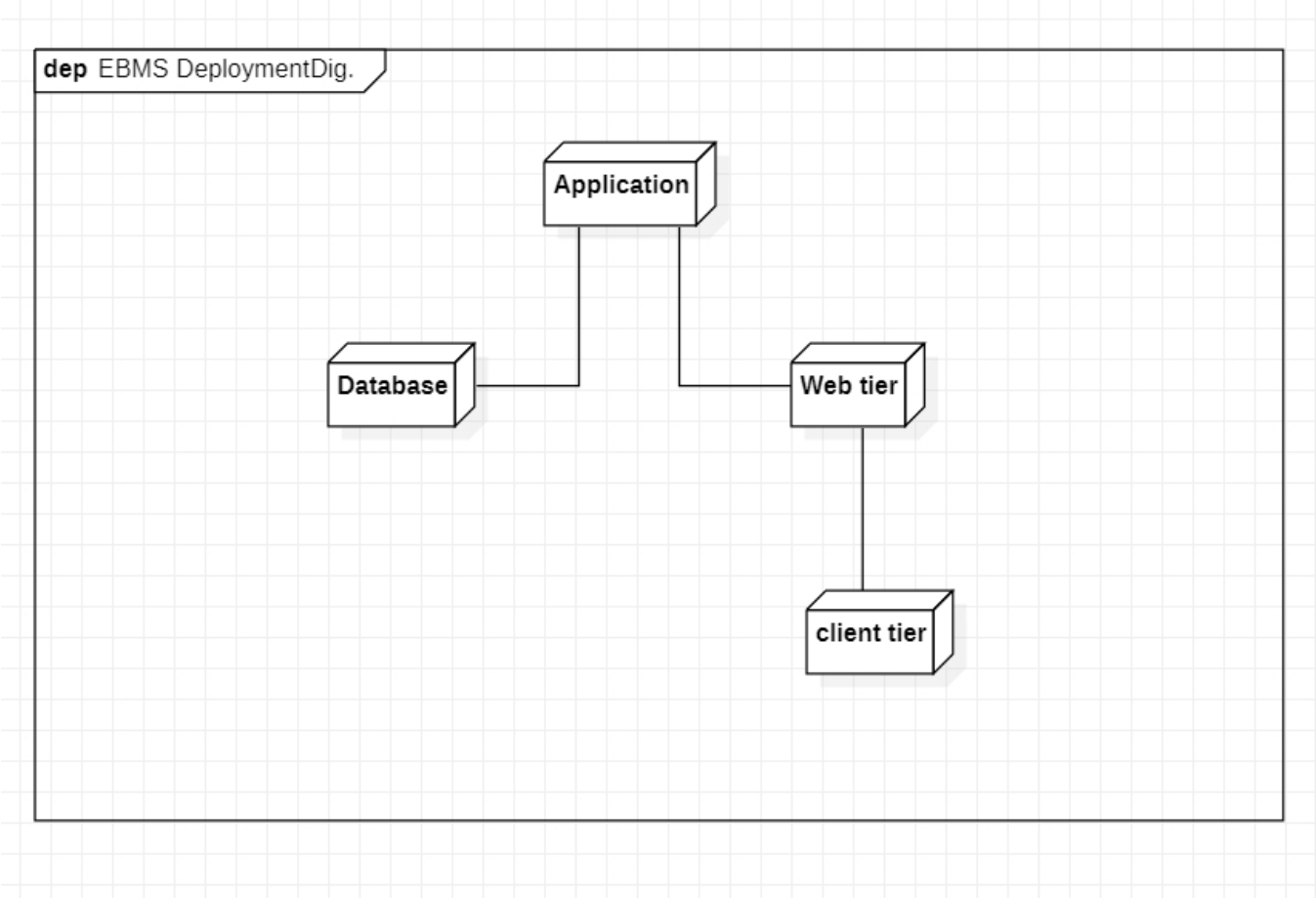
Activity diagram is another important **diagram** in UML to describe the dynamic aspects of the system. **Activity diagram** is basically a flowchart to represent the flow from one **activity** to another **activity**. The **activity** can be described as an operation of the system. The control flow is drawn from one operation to another.

7. Component Diagram :



Component diagram is a special kind of **diagram** in UML. ... Thus from that point of view, **component diagrams** are used to visualize the physical **components** in a system. These **components** are libraries, packages, files, etc. **Component diagrams** can also be described as a static implementation view of a system.

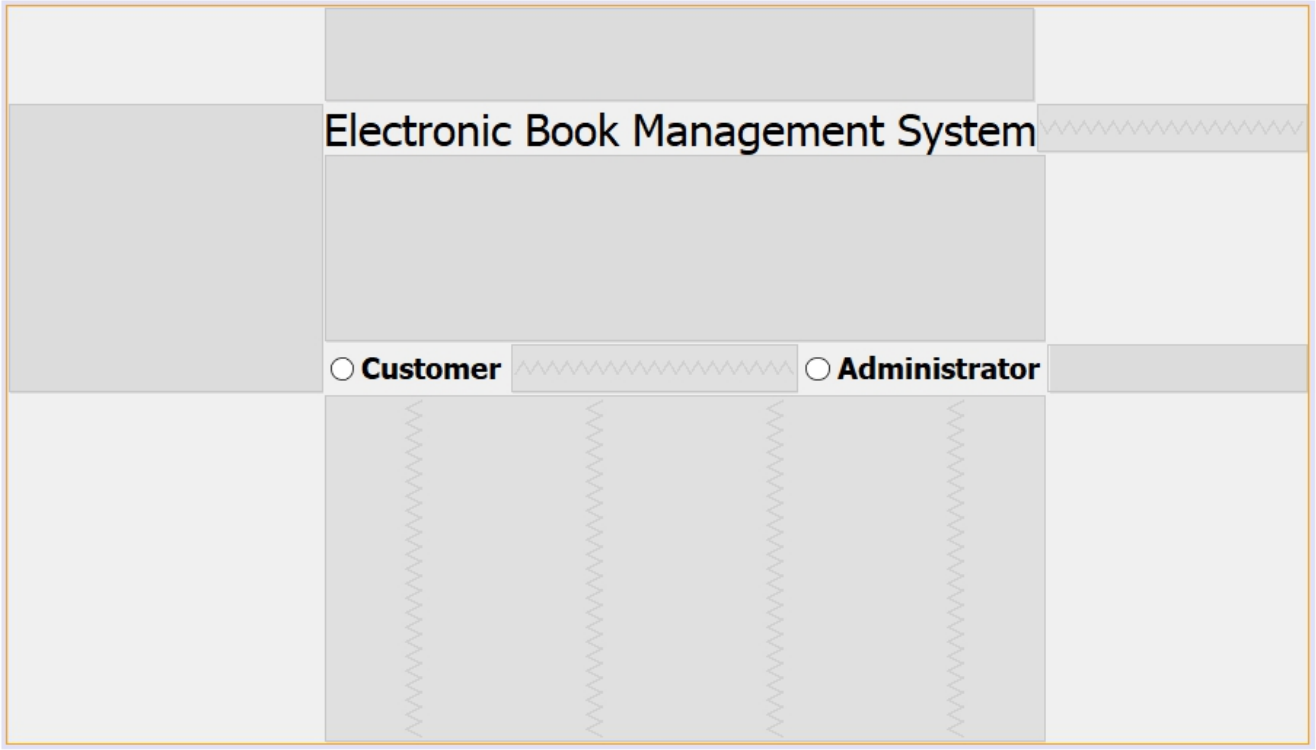
8. Deployment Diagram :



A **deployment diagram** is a UML **diagram** type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. **Deployment diagrams** are typically used to visualize the physical hardware and software of a system.

Source Code :

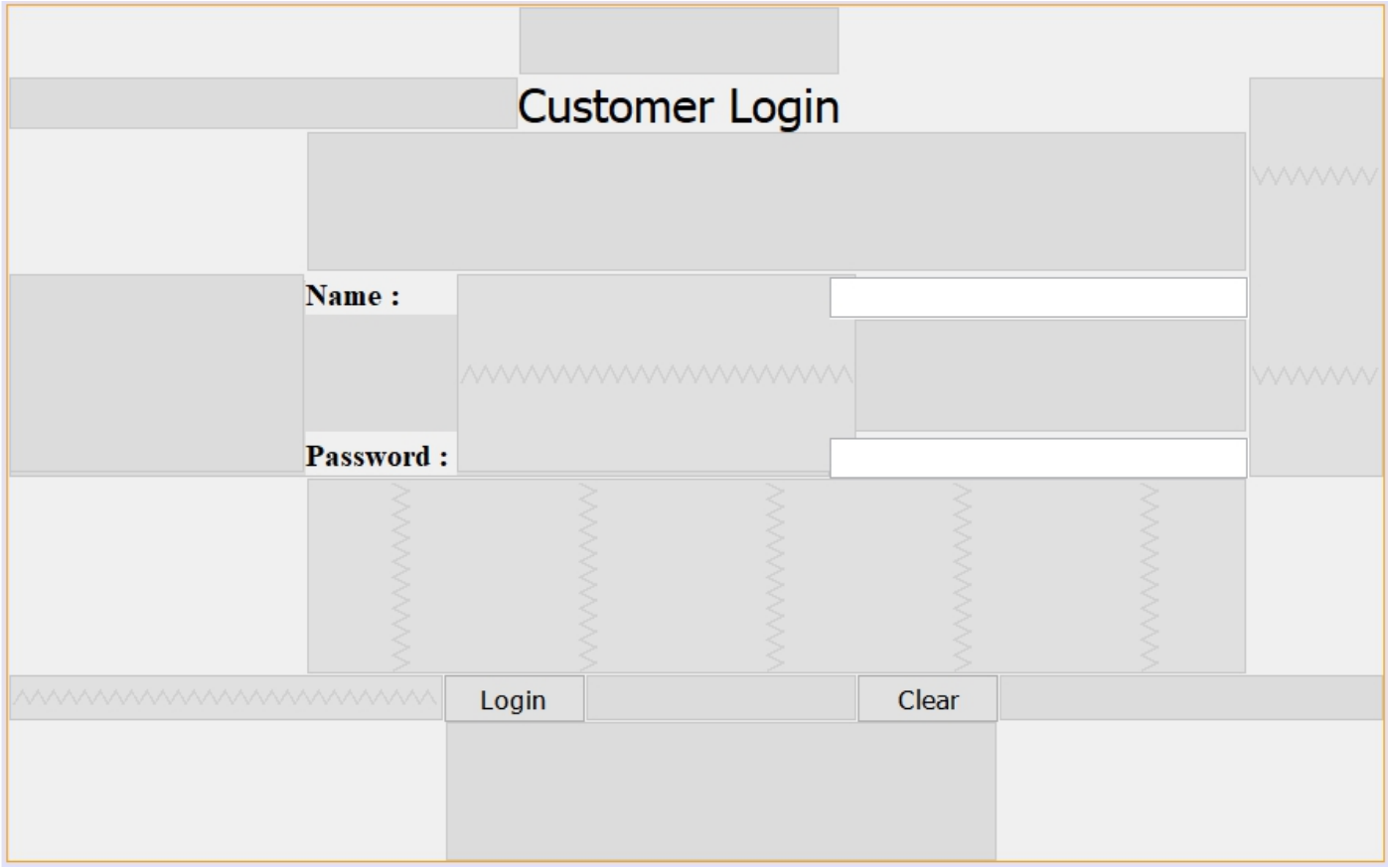
//Opening Frame :




```
public class Opening extends javax.swing.JFrame {
   public Opening() {
      initComponents();
   }
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      new User_Reg1().setVisible(true);
      dispose();

   }
   private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      new Admin().setVisible(true);
      dispose();
   }
```

```
public static void main(String args[]) {
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Opening().setVisible(true);
        }
    });
}
```

//Customer Login Frame



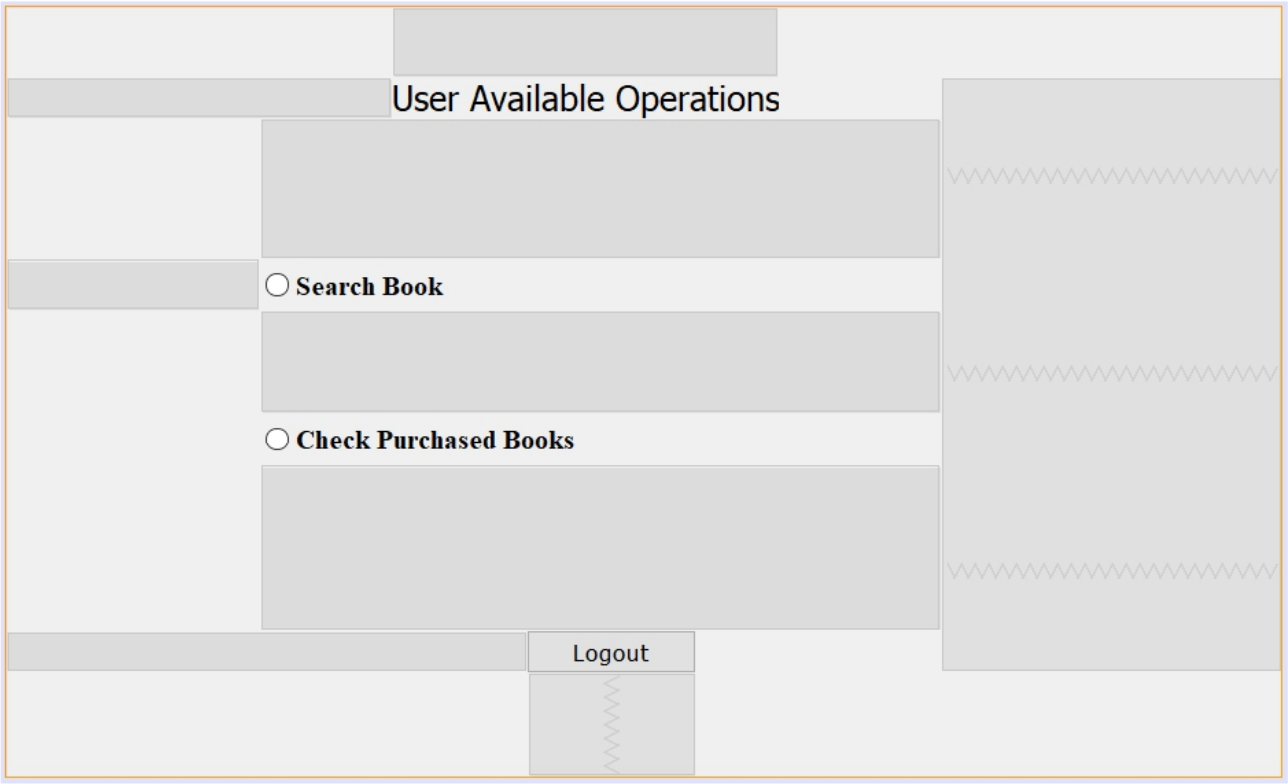

```
import java.sql.*;
import javax.swing.JOptionPane;
public class User_Reg1 extends javax.swing.JFrame {
   public User_Reg1() {
      initComponents();
   }
   private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      nametxt.setText("");
      passwordtxt.setText("");
   }
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String name = nametxt.getText();
    String word = new String(passwordtxt.getPassword());
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
        Statement st=con.createStatement();
        String query="select First_Name , Password from reg_user";
        ResultSet rs=st.executeQuery(query);
        int count=0;
        while(rs.next())
        {
            count++;
            String user_name=rs.getString("First_Name");
            String pass_word=rs.getString("Password");

            if(name.equals(user_name) )
            {
                if(word.equals(pass_word))
                {
                    User_operations op =new User_operations(name);
                     op.setVisible(true);
                    dispose();
                    break;
                }
           else
                {
                    JOptionPane.showMessageDialog(null,"Wrong Password!");
                    nametxt.setText("");
                    passwordtxt.setText("");
                    break;
                }}
```

```
            else if(rs.next()==false)
            {
               JOptionPane.showMessageDialog(null,"Invalid UserID");

               new UnRegUser().setVisible(true);
               nametxt.setText("");
                passwordtxt.setText("");
 }
            else
            {
               continue;
}}  }
      catch(Exception e)
      {
         JOptionPane.showMessageDialog(null,e);
      }}

public static void main(String args[]) {
      /* Create and display the form */
      java.awt.EventQueue.invokeLater(new Runnable() {
         public void run() {
            new User_Reg1().setVisible(true);
         }
      });
   }
```

```
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JTextField nametxt;
    private javax.swing.JPasswordField passwordtxt;
    // End of variables declaration
}
```

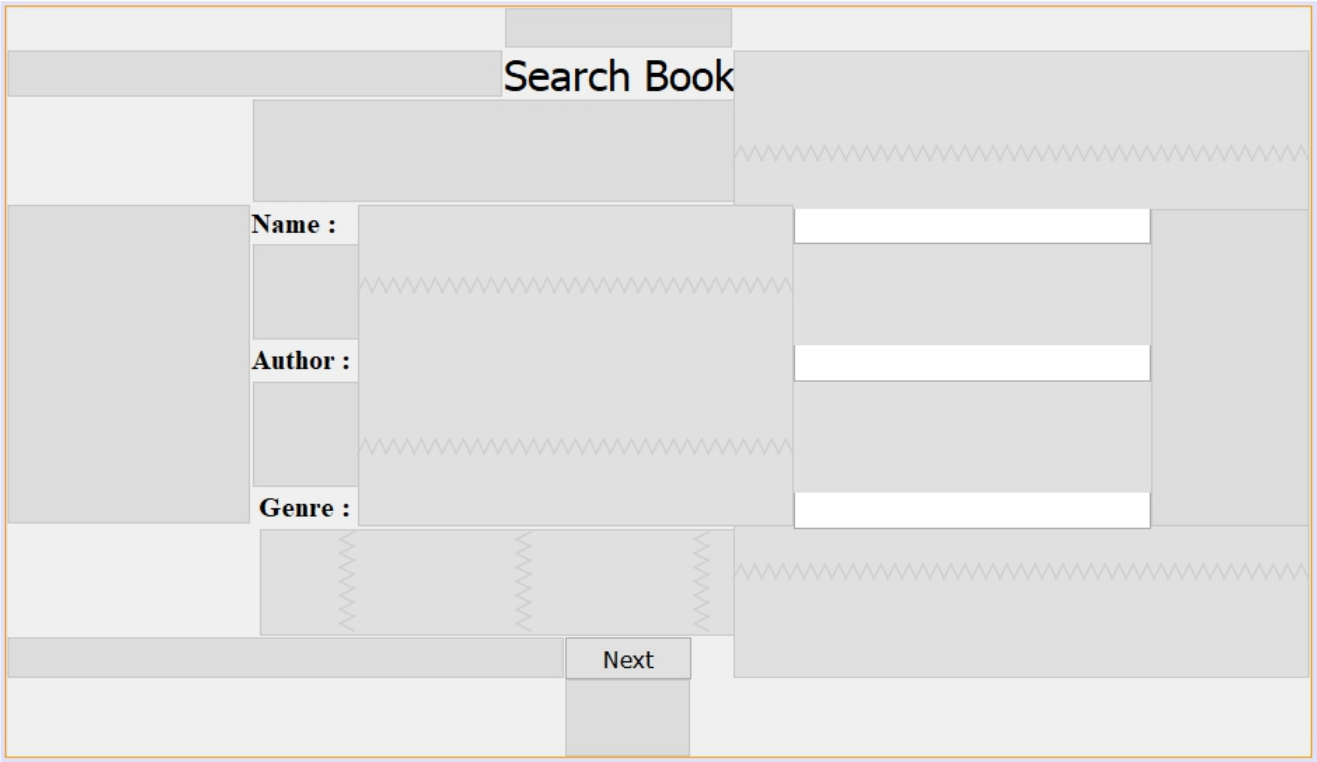
//User Operations Frame




```
public class User_operations extends javax.swing.JFrame {
   public String user_name;
   public User_operations() {
      initComponents();
   }

   User_operations(String name) {
   initComponents();
   user_name = name;
   }
   private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
     new Search_Book(user_name).setVisible(true);
   }
```

```
    private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Check_purchased(user_name).setVisible(true);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new User_operations().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    // End of variables declaration
}
```

//Search Book Frame

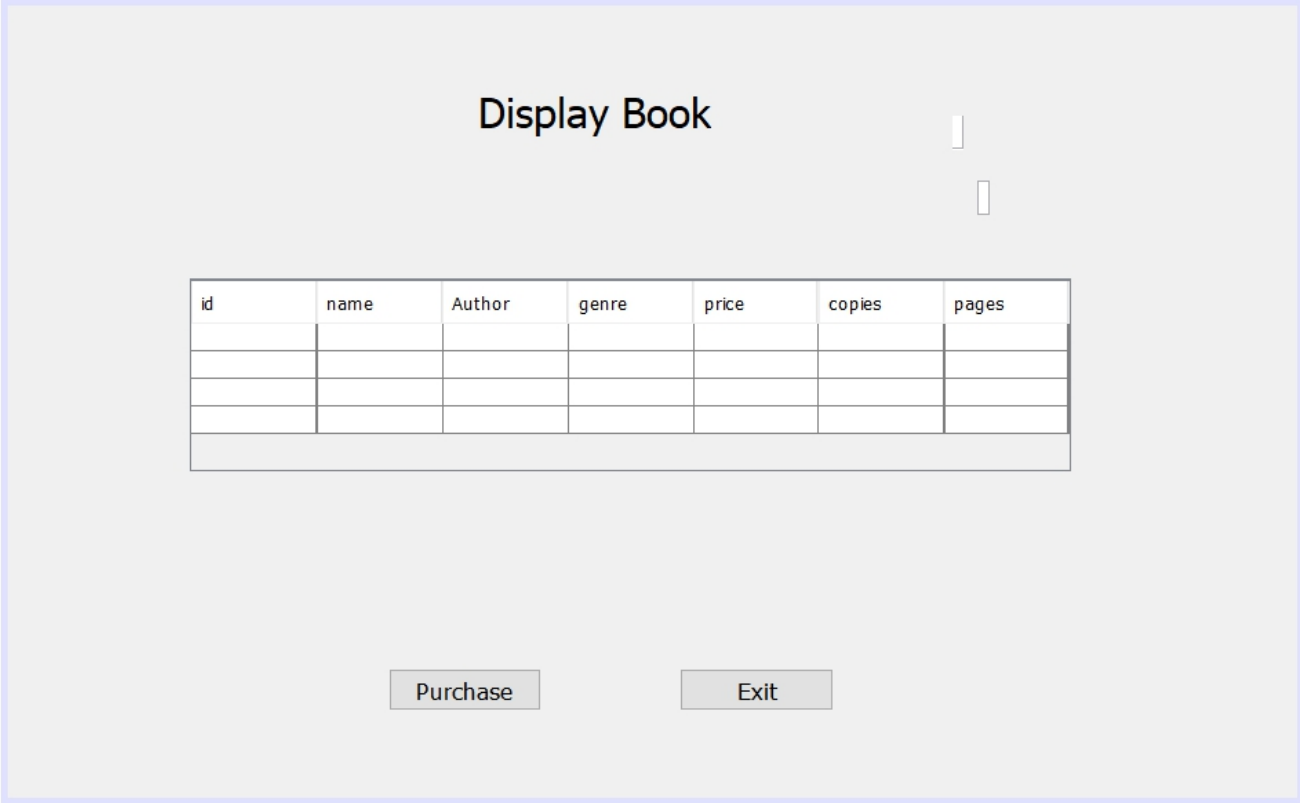



```
import java.sql.*;
import javax.swing.JOptionPane;
public class Search_Book extends javax.swing.JFrame {
   public Search_Book() {
      initComponents();
   }
   String n1;
   public Search_Book(String na) {
      initComponents();
      n1 = na; //n1 - customer name from User_Reg1
   }
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here
     String book_name = jTextField1.getText();
    String book_author = author.getName();
    String book_genre = genre.getName();
      try{
       Class.forName("com.mysql.jdbc.Driver");
       Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
       Statement st=con.createStatement();
       String query="select name,author,genre from book_info";
       ResultSet rs=st.executeQuery(query);
       int count=0;
       while(rs.next())
       {
          count++;

          String n=rs.getString("name");
          String a=rs.getString("author");
           String g=rs.getString("genre");
           if(book_name.equals(n))
           {
               book_display book = new book_display(n,n1);
               book.setVisible(true);
              dispose();
               break;
           }
           else if(rs.next()== false)
           {
               JOptionPane.showMessageDialog(null,"Book Not Found!!");
          }
```

```
            else
            {
                continue;
            }} }

        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,e);
} }
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Search_Book().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField author;
    private javax.swing.JTextField genre;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

//Display Book Frame



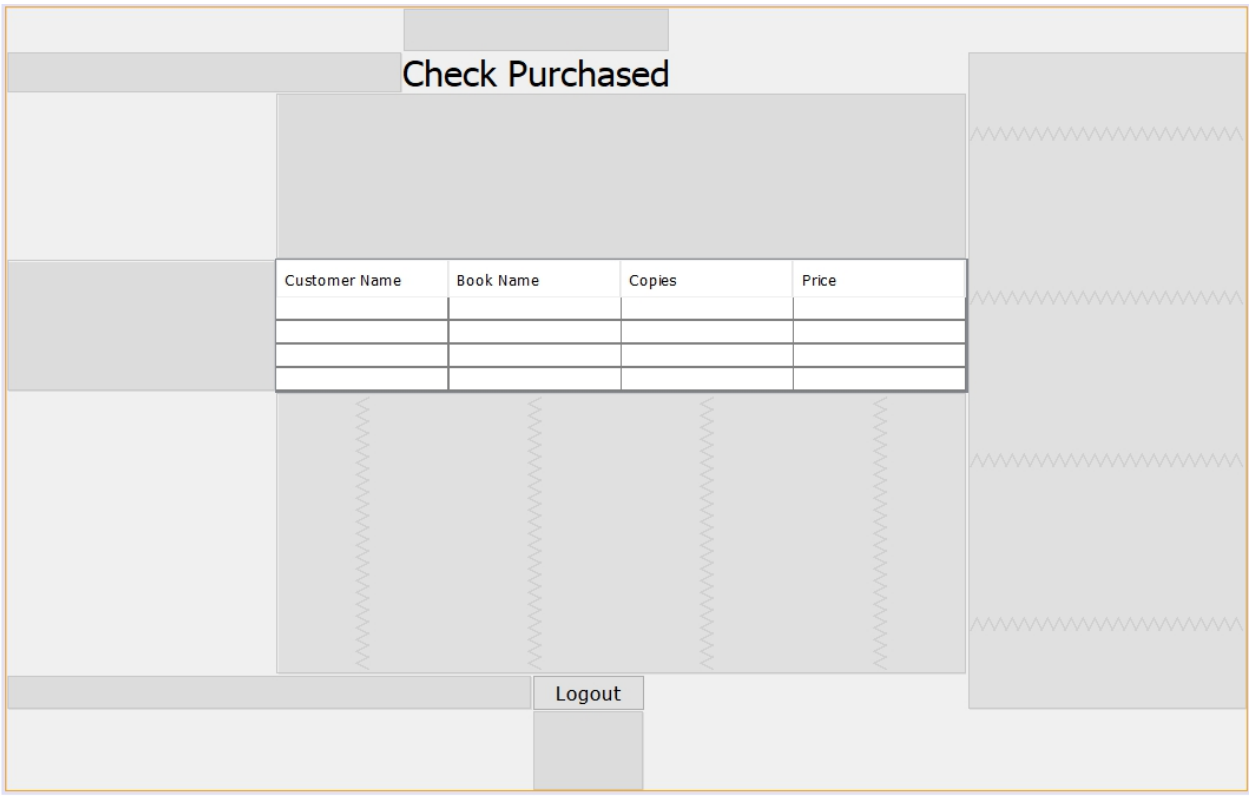

```
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;

public class book_display extends javax.swing.JFrame {
   public String name;
    public String cust_name; //variable to store the passed value from Search_Book JFrame
   public String book_name;//variable to store the book name
   public int p;
```

```
public book_display() {
    initComponents();
    }
 public book_display(String S,String n) {
    initComponents();
    username.setVisible(false);
    username.setEditable(false);
     username.setText(n);
       bname.setVisible(false);
    bname.setEditable(false);
     bname.setText(S);
    name =S;
     try
     {
    Class.forName("com.mysql.jdbc.Driver");
   Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
    String sql="select * from book_info where name = '"+name+"'";
     Statement st = conn.createStatement();
    ResultSet rs=st.executeQuery(sql);
     table1.setModel(DbUtils.resultSetToTableModel(rs));
    if(rs.next())
    {
        int val = rs.getInt("copies");
        if(val == 0 )
        {
            JOptionPane.showMessageDialog(null,"Out Of Stock!!");
            new Search_Book().setVisible(true);
         }}  }
    catch(Exception e){
        JOptionPane.showMessageDialog(null,e);
     }}
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        dispose();
        new User_operations().setVisible(true);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
       int result =  JOptionPane.showConfirmDialog(null,"Are u sure you want to purchase the book ");
       if(result == JOptionPane.YES_OPTION)
       {
            try
        {
        Class.forName("com.mysql.jdbc.Driver");
       Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
         Statement st1 = conn.createStatement();
        String sql="update book_info set copies = copies -1 where name = '"+name+"'";
         st1.executeUpdate(sql);
   String sql1="select price from book_info  where name = '"+name+"'";
    Statement st2 = conn.createStatement();
    ResultSet rs1 = st2.executeQuery(sql1);
    if(rs1.next())
    {
        p= rs1.getInt("price");
    }
        String sq21 ="insert into customer_purchased values(?,?,?,?) ";
        PreparedStatement pstmt2 = conn.prepareStatement(sq21);
       pstmt2.setString(1,username.getText());
       pstmt2.setString(2,bname.getText());
       pstmt2.setInt(3, 1);
       pstmt2.setInt(4, p);
       pstmt2.executeUpdate();
```

```
        }


        catch(Exception e){
            JOptionPane.showMessageDialog(null,e);
        } }}
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new book_display().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField bname;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable table1;
    private javax.swing.JTextField username;
    // End of variables declaration
}
```

//Check_purchased Frame

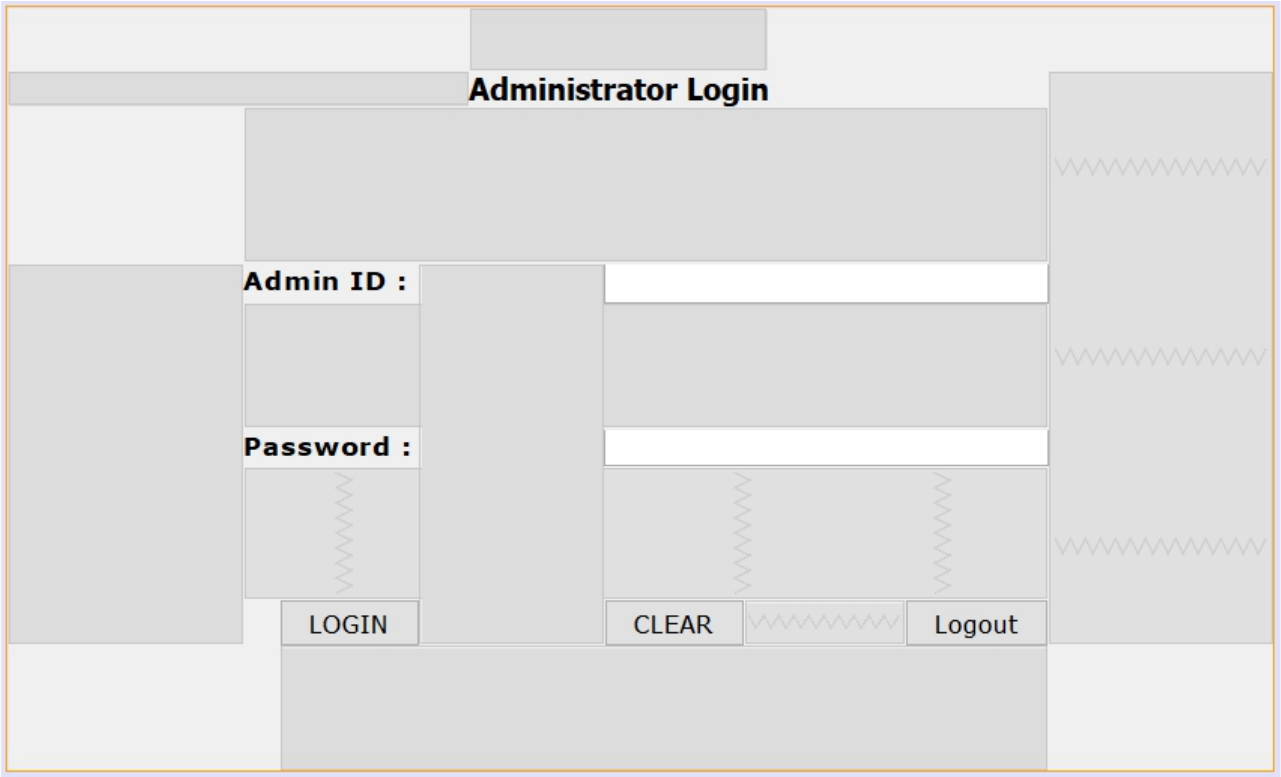



```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import net.proteanit.sql.DbUtils;
public class Check_purchased extends javax.swing.JFrame {
   public Check_purchased() {
      initComponents();
   }
```

```
public Check_purchased(String username) {
    initComponents();
    try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
       String sql = "select * from customer_purchased where customer_name = ?";
       PreparedStatement pst = con.prepareStatement(sql);


       pst.setString(1, username);


        ResultSet rs=pst.executeQuery();
     table1.setModel(DbUtils.resultSetToTableModel(rs));
       }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"message");
    }
}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Check_purchased().setVisible(true);
        }
    });
}
```

```
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable table1;
    // End of variables declaration
}
```

//Admin login frame



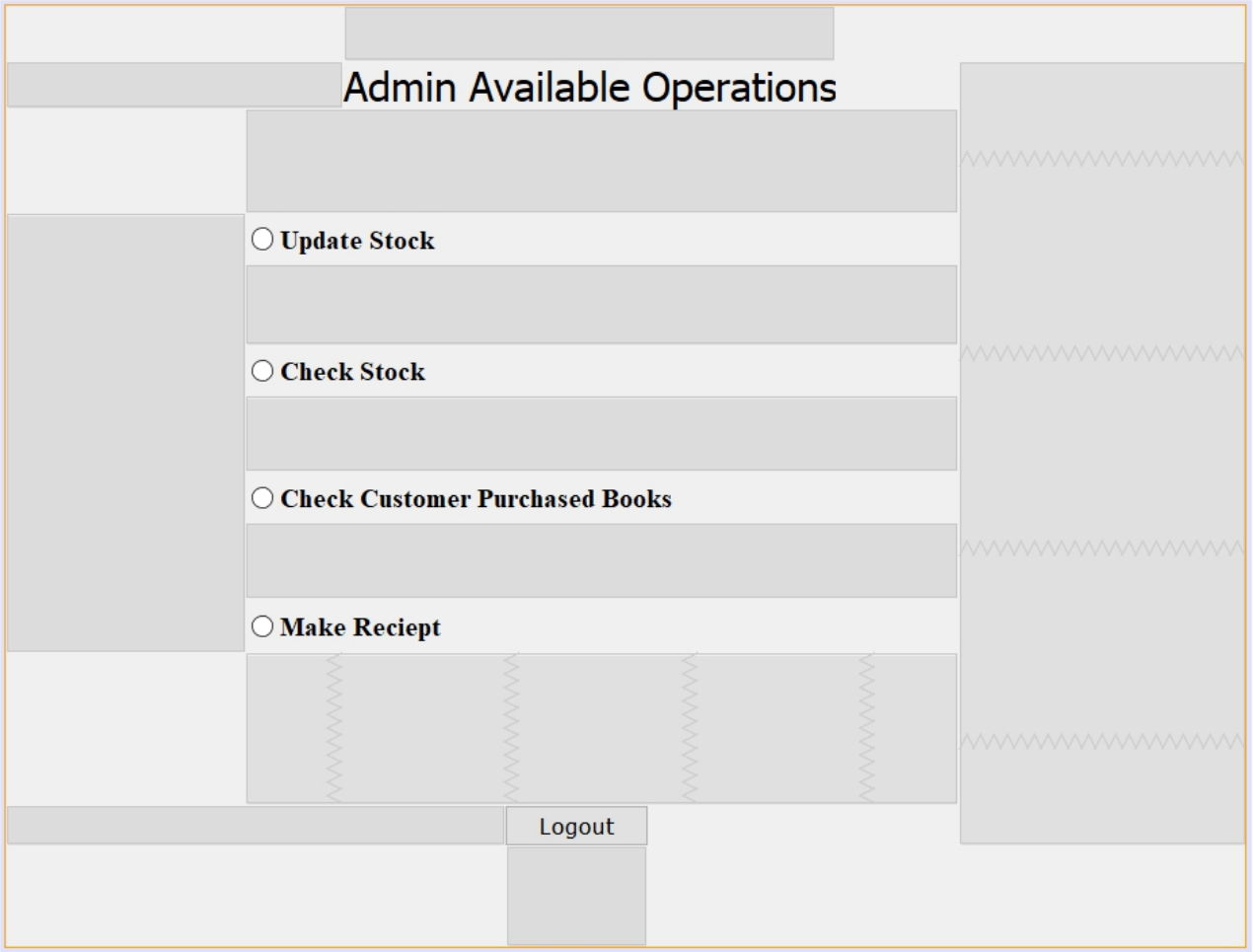

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class Admin extends javax.swing.JFrame {
   public Admin() {
      initComponents();
   }
   private void adminidActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
   }
   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      // TODO add your handling code here:
      int id1 = Integer.parseInt(adminid.getText());
```

```
    String word = new String(pasword.getPassword());
try
    {
      Class.forName("com.mysql.jdbc.Driver");
      Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
      String query="select * from admin where id = ? and Password = ?";
      PreparedStatement st=con.prepareStatement(query);
      st.setInt(1,id1);
      st.setString(2, word);
      ResultSet rs=st.executeQuery();
      if(rs.next())
      {
        new Admin_operations().setVisible(true);
        dispose();
       }
      else
      {
         JOptionPane.showMessageDialog(null,"login filled");
      } }
     catch(Exception e)
    {
      JOptionPane.showMessageDialog(null,"connection error");
    }
     }
```

```
  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
```

```
        adminid.setText("");
        pasword.setText("");
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Admin().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField adminid;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JPasswordField pasword;
    // End of variables declaration
}
```

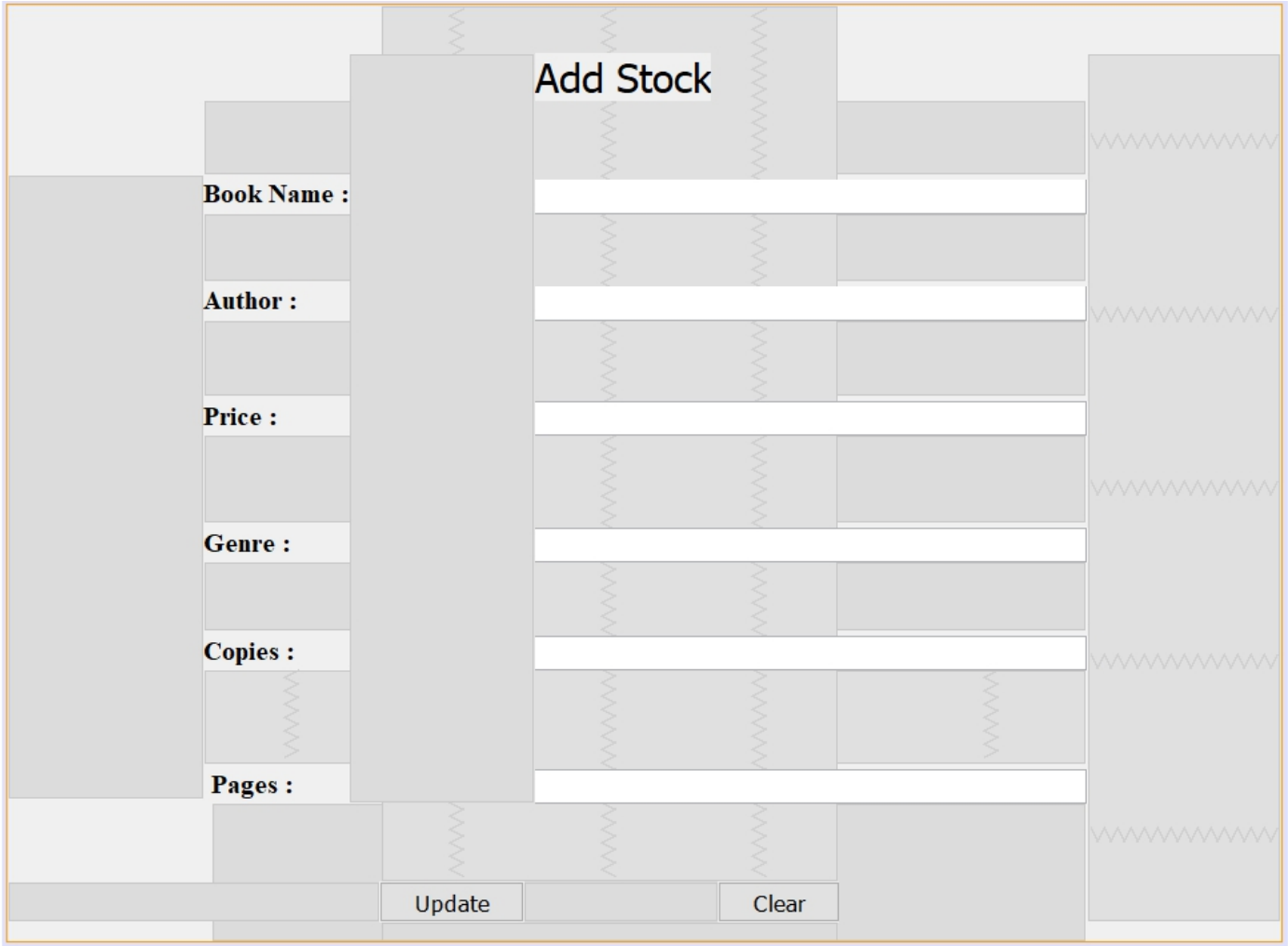
//Admin Operations Frame




```
public class Admin_operations extends javax.swing.JFrame {
    public Admin_operations() {
        initComponents();
    }
    private void jRadioButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Check_customer_purchases().setVisible(true);
    }
    private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Update_Stock().setVisible(true);
    }
```

```
    private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new CheckStock().setVisible(true);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }
    private void jRadioButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new printreciept().setVisible(true);
    }
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Admin_operations().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JRadioButton jRadioButton3;
    private javax.swing.JRadioButton jRadioButton4;
    // End of variables declaration
}
```

//Update Stock



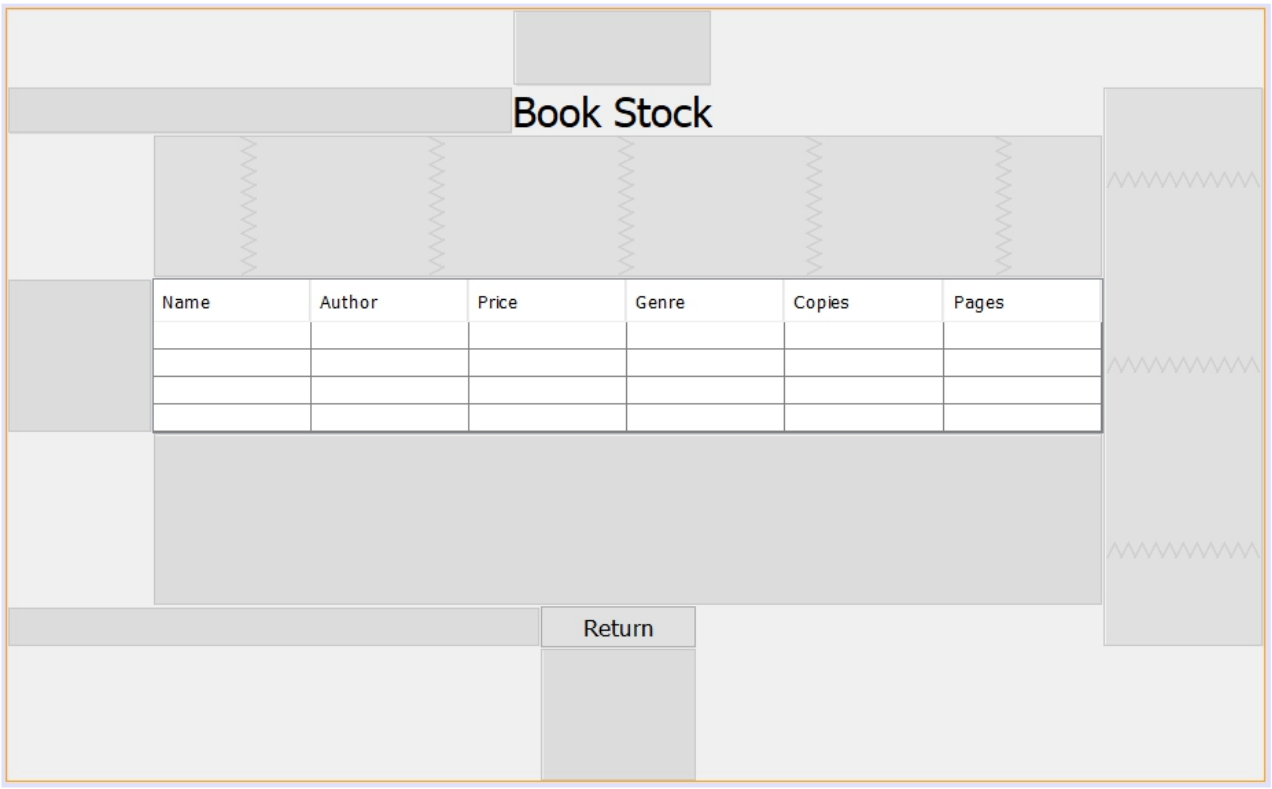

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Update_Stock extends javax.swing.JFrame {
   public Update_Stock() {
      initComponents();
   }
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here
    try
    {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
         String query="insert into book_info values(?,?,?,?,?,?)";
        String query1 = "select name from book_info where name = ?";





        PreparedStatement st=con.prepareStatement(query);
          PreparedStatement st1=con.prepareStatement(query1);
        st1.setString(1, bname.getText());
        ResultSet rs1 = st1.executeQuery();
        if(rs1.next())
        {
            Statement st2 = con.createStatement();
    String sql="update book_info set copies = copies +5 where name = '"+bname.getText()+"'";
     st1.executeUpdate(sql);
     JOptionPane.showMessageDialog(null, "updated");
        }
        else
        {
             st.setString(2, auth.getText());
            int p = Integer.parseInt(price.getText());
            st.setInt(3,p);
              st.setString(4, genre.getText());
               int cp = Integer.parseInt(copies.getText());
            st.setInt(5,cp);
```

```
            int pg = Integer.parseInt(pages.getText());
            st.setInt(6,pg);
        st.executeUpdate();
        JOptionPane.showMessageDialog(null,"update successfull");
          bname.setText("");
     auth.setText("");
      price.setText("");
       genre.setText("");
        copies.setText("");
         pages.setText("");
        }
    }
    catch(Exception e)
    {
       JOptionPane.showMessageDialog(null,"error");
    }
      }
  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    bname.setText("");
     auth.setText("");
      price.setText("");
       genre.setText("");
        copies.setText("");
         pages.setText("");

  }
```

```
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Update_Stock().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JTextField auth;
    private javax.swing.JTextField bname;
    private javax.swing.JTextField copies;
    private javax.swing.JTextField genre;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JTextField pages;
    private javax.swing.JTextField price;
    // End of variables declaration
}
```

//Check Stock Frame

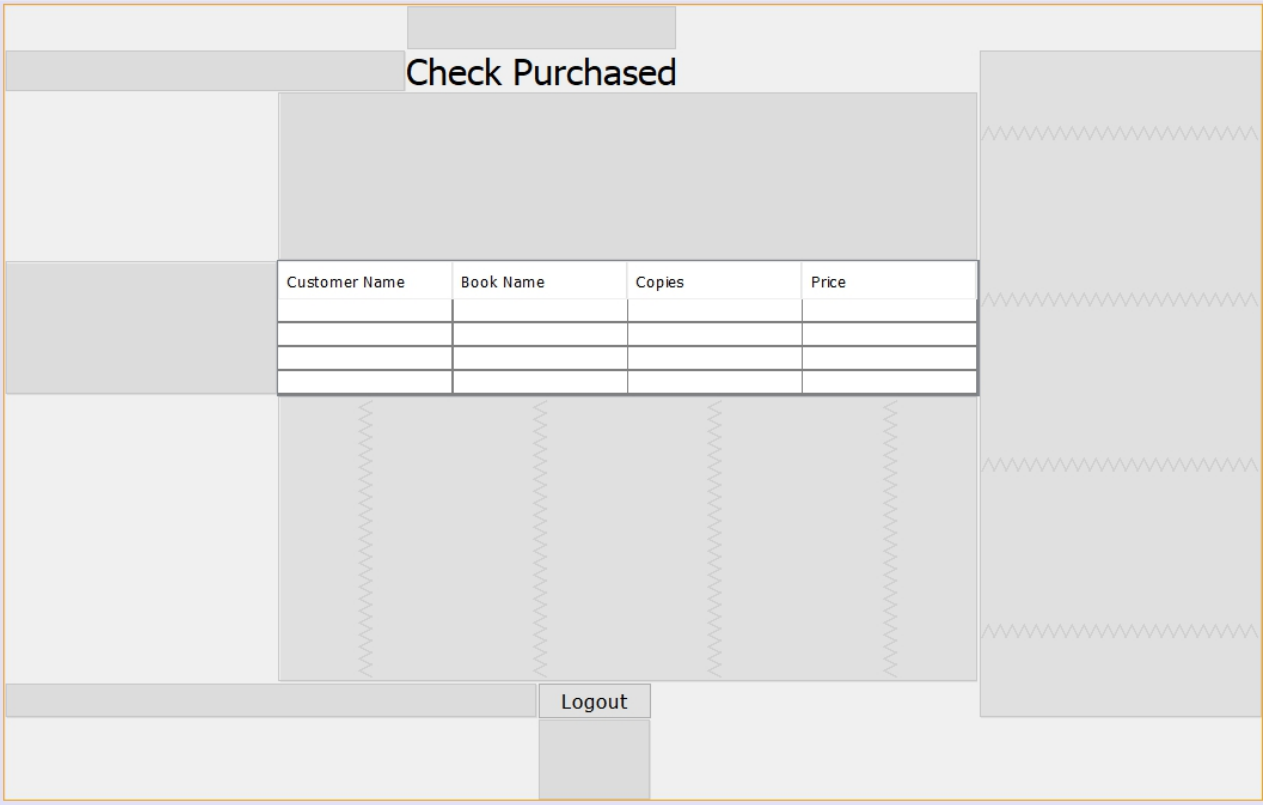



```
import java.sql.*;
import java.beans.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
```

```
public class CheckStock extends javax.swing.JFrame {
   /**
    * Creates new form CheckStock
    */
   public CheckStock() {
      initComponents();


       try{
          Class.forName("com.mysql.jdbc.Driver");
          Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
          String sql = "select * from book_info";
         PreparedStatement pst = con.prepareStatement(sql);
          ResultSet rs=pst.executeQuery(sql);


       while(rs.next())
       {
           String bname = rs.getString("name");
           String aname = rs.getString("author");
           String p =String.valueOf(rs.getInt("price"));
          String gname = rs.getString("genre");
            String c =String.valueOf(rs.getInt("copies"));
           String page =String.valueOf(rs.getInt("pages"));
           table1.setModel(DbUtils.resultSetToTableModel(rs));
       }
           dispose();
       }
       catch(Exception e)
       {
          JOptionPane.showMessageDialog(null,"message");
       }}
```

```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        new Admin_operations().setVisible(true);
    }
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new CheckStock().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable table1;
    // End of variables declaration
}
```

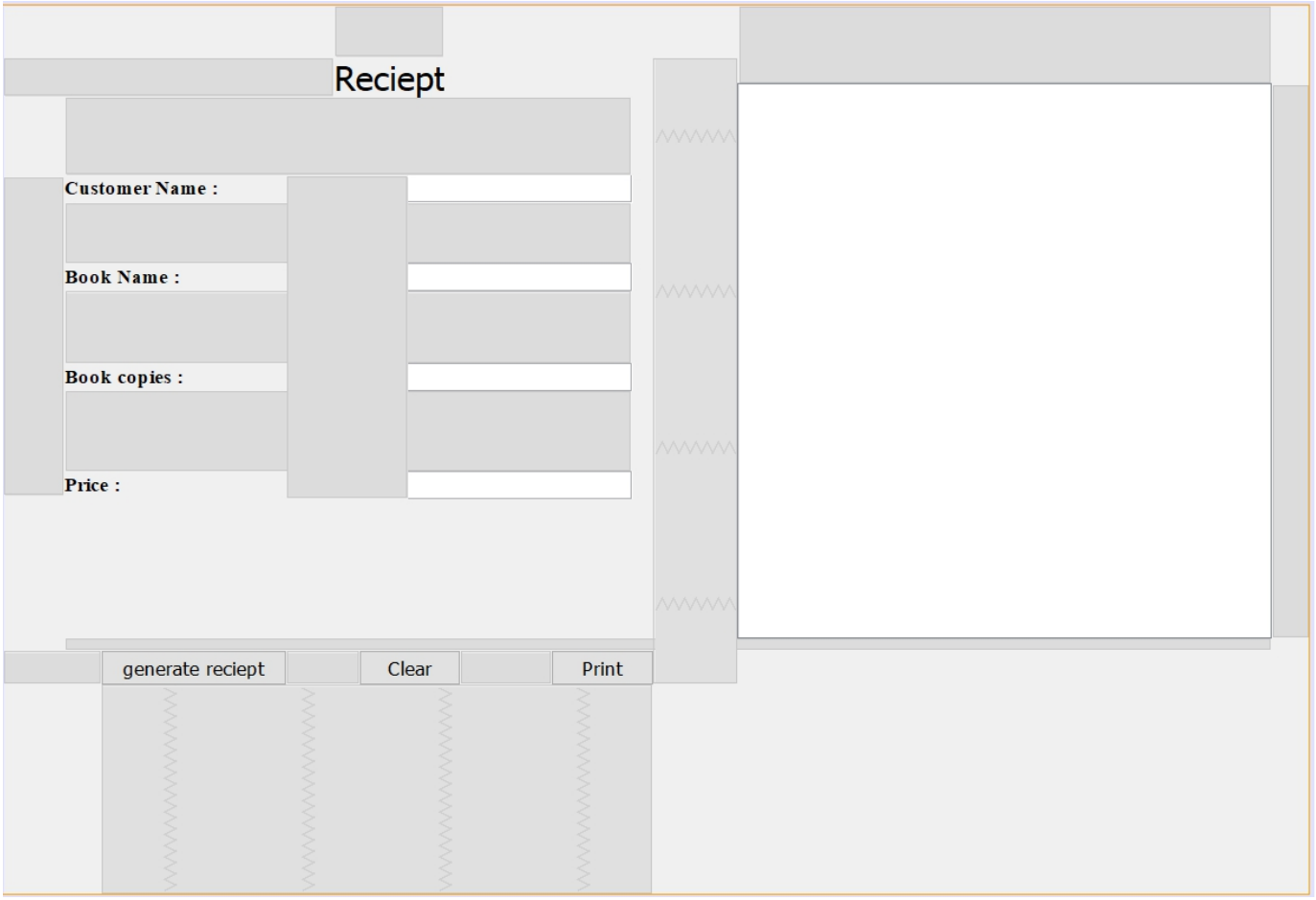
//Check_customer_ Purchases frame




```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
public class Check_customer_purchases extends javax.swing.JFrame{
   public Check_customer_purchases() {
      initComponents();
       try{
         Class.forName("com.mysql.jdbc.Driver");
         Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/books","root","");
        String sql = "select * from customer_purchased ";
        PreparedStatement pst = con.prepareStatement(sql);
```

```
                ResultSet rs=pst.executeQuery();
        table1.setModel(DbUtils.resultSetToTableModel(rs));
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,"message");
        } }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.exit(0);
    }
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Check_customer_purchases().setVisible(true);
            } }); }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable table1;
    // End of variables declaration
}
```

//print receipt frame



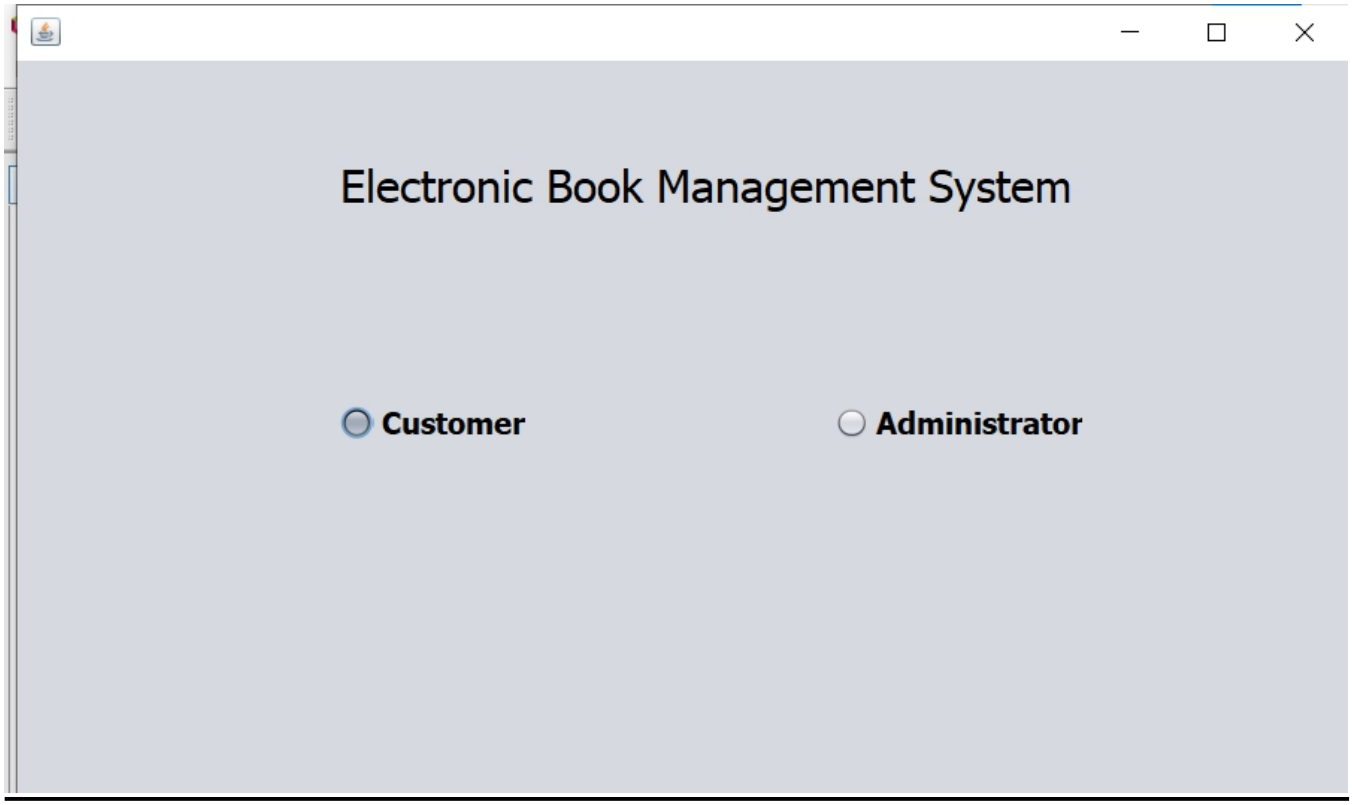

```
import java.awt.print.PrinterException;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;
public class printreciept extends javax.swing.JFrame {
    /**
     * Creates new form printreciept
     */
    public printreciept() {
        initComponents();
    }
```

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    cname.setText("");
      bname.setText("");
        copy.setText("");
          pri.setText("");
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:
        areatxt.print();
    } catch (PrinterException ex) {
        Logger.getLogger(printreciept.class.getName()).log(Level.SEVERE, null, ex);
    }
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
          Date obj = new Date();
          String date = obj.toString();
          String title = jLabel1.getText();
            int copy1 = Integer.parseInt(copy.getText());
          int price1 = Integer.parseInt(pri.getText());
  areatxt.setText(areatxt.getText()+"\n"+date+"\n\n");
  areatxt.setText(areatxt.getText()+"Customer Name : "+cname.getText()+"\n\n");
   areatxt.setText(areatxt.getText()+"Book Name : "+bname.getText()+"\n\n");
    areatxt.setText(areatxt.getText()+"Book Copies : "+copy1+"\n\n");
     areatxt.setText(areatxt.getText()+"Book Price : "+price1+"\n\n");
}
```

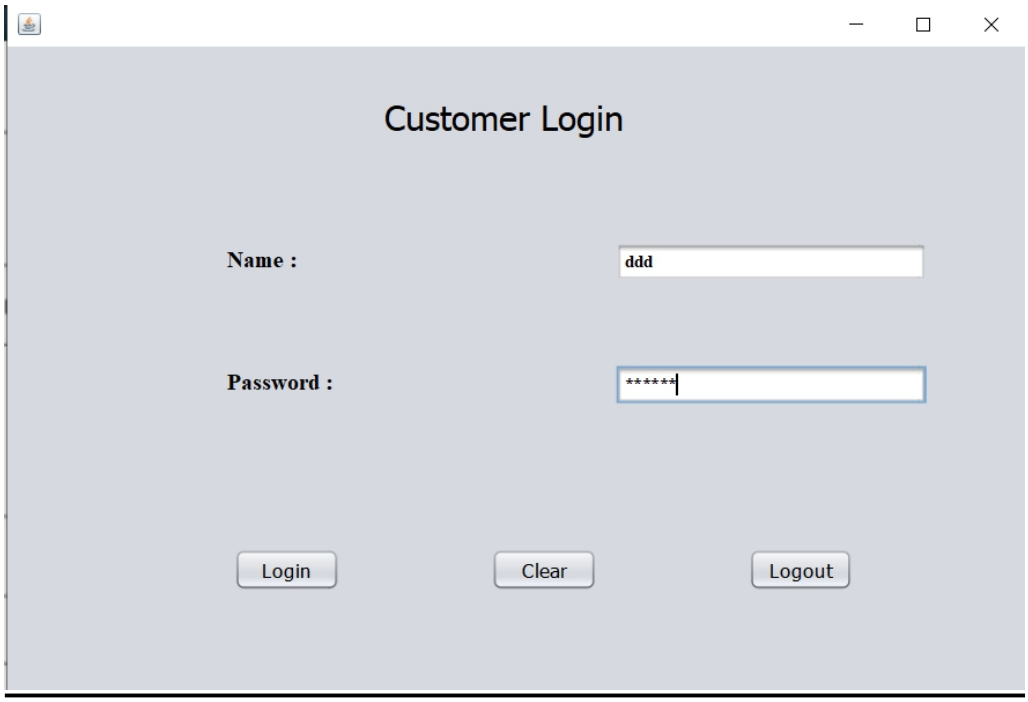
```
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new printreciept().setVisible(true);
            }}); }
    // Variables declaration - do not modify
    private javax.swing.JTextArea areatxt;
    private javax.swing.JTextField bname;
    private javax.swing.JTextField cname;
    private javax.swing.JTextField copy;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextField pri;
}
```

WINDOWS OF OPERATIONS PERFORMED :

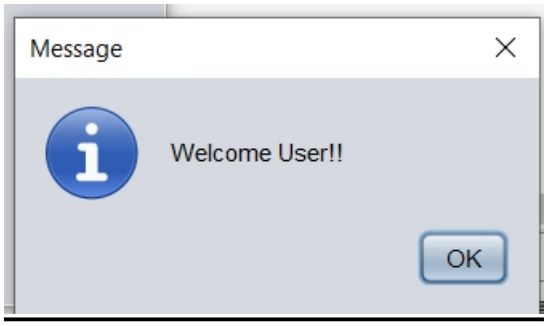
Customer logging in and purchasing Book :



Login Credentials are typed :

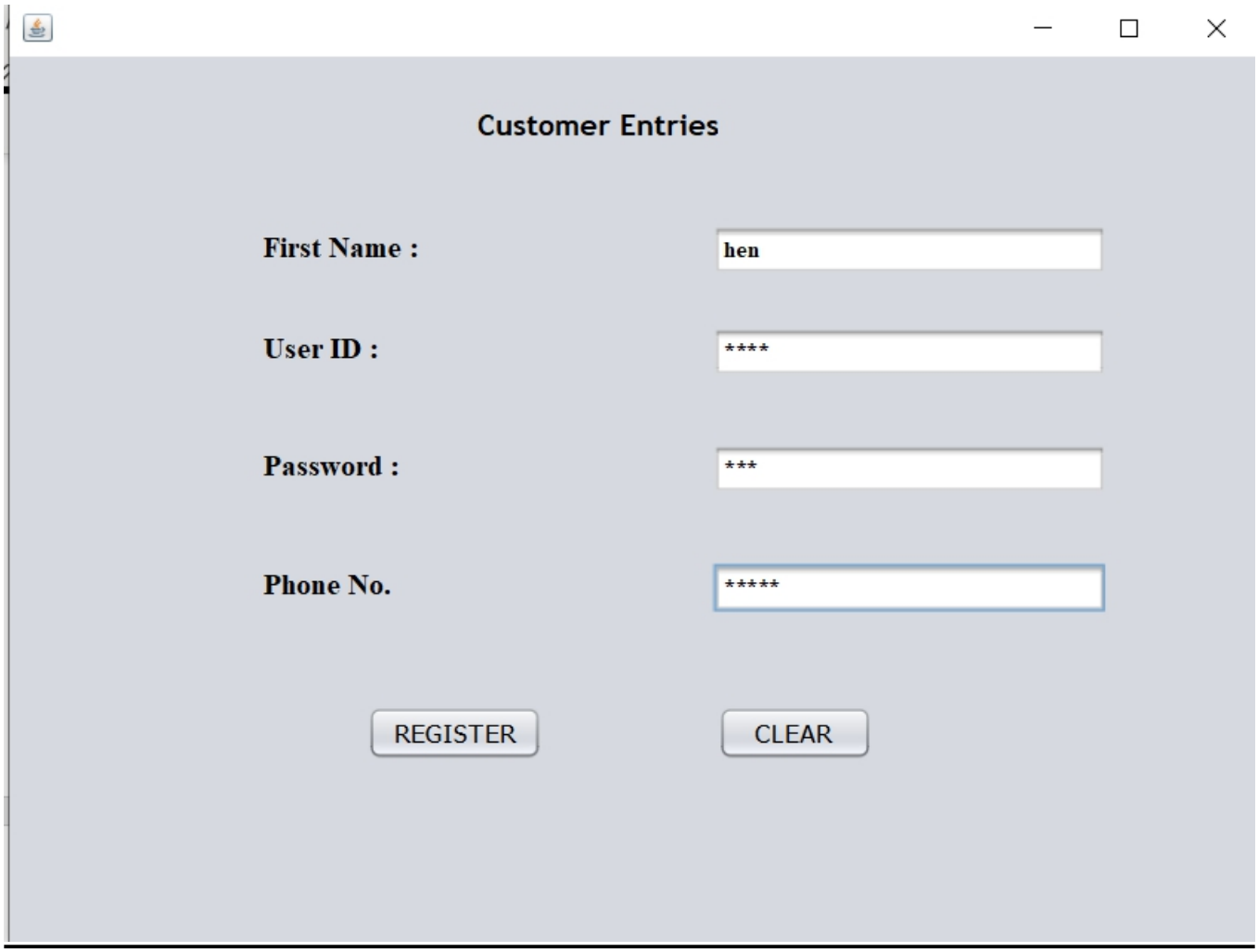


In case of correct credentials entered :



In case of incorrect credentials entered :

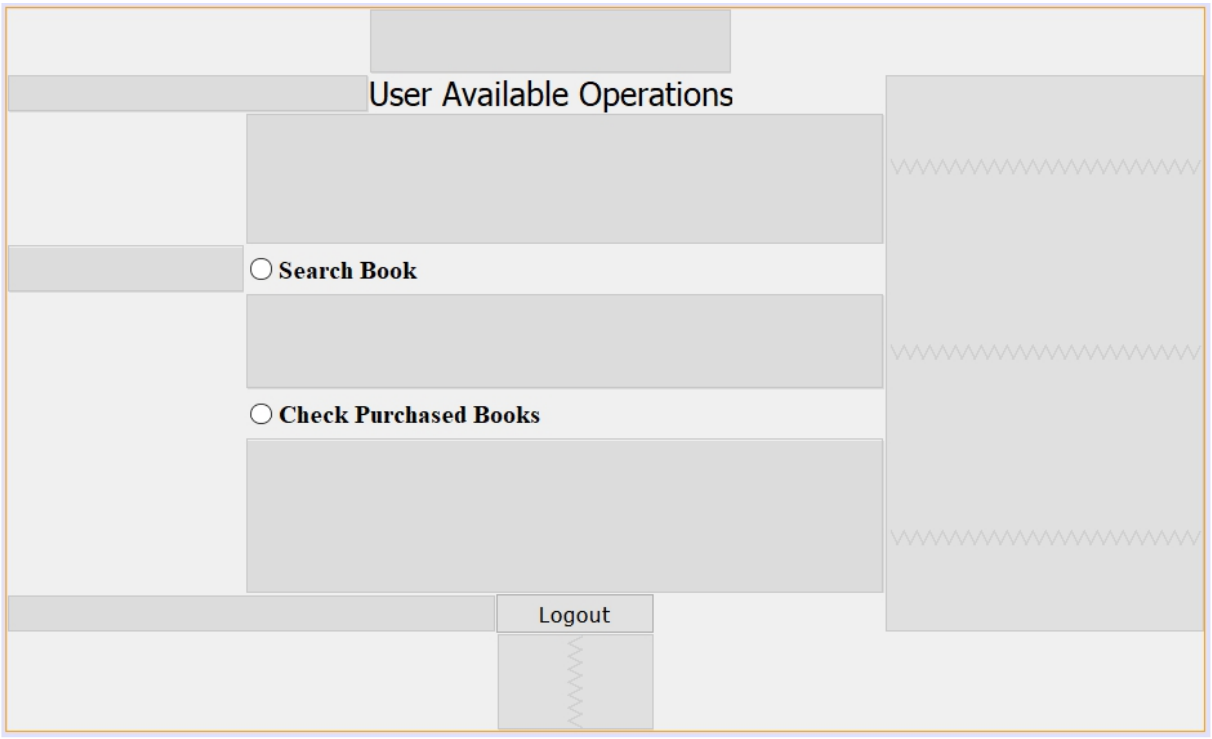
And it navigates to the registration frame :



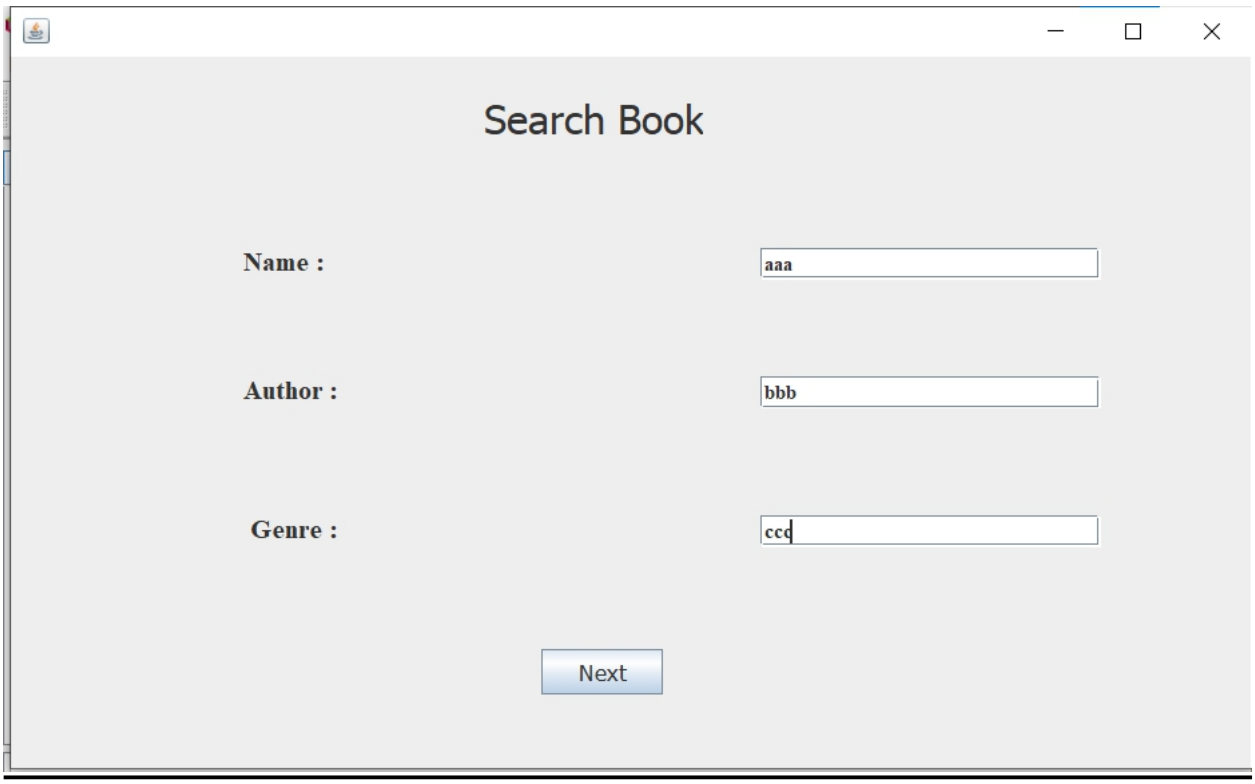
and appended to the database :

←T→			id	First_Name	Phno	Password
Edit	Copy	Delete	432	ping	5654	fsf
Edit	Copy	Delete	512	aaa	15165	qzx25
Edit	Copy	Delete	543	bbb	15902	vnrvjiet
Edit	Copy	Delete	647	ccc	16410	abc379
Edit	Copy	Delete	712	ddd	22634	xyz123
Edit	Copy	Delete	1321	hen	945	pan
Edit	Copy	Delete	8629	Akhil	284	ggg

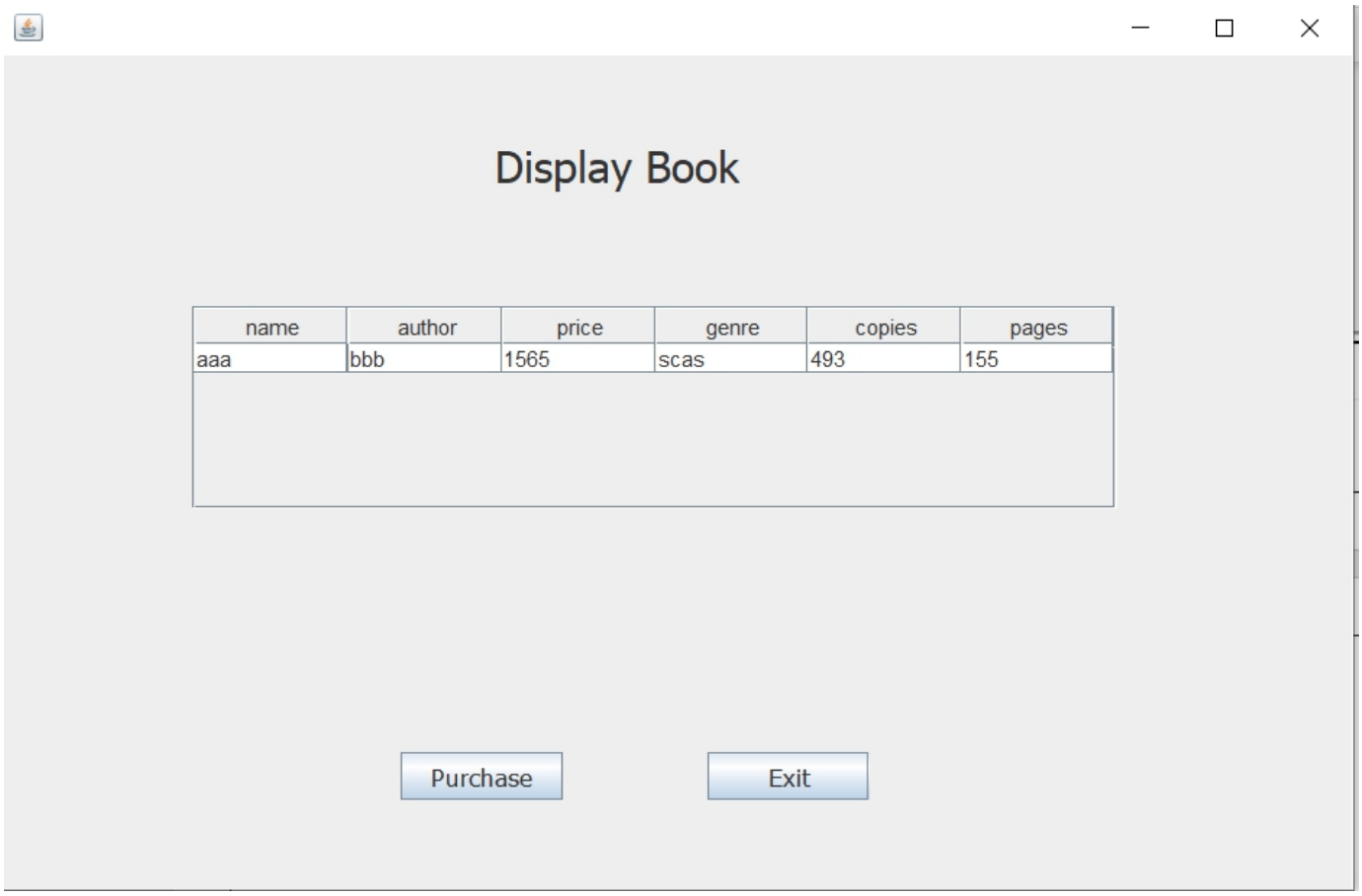
User has to select one of the operations :



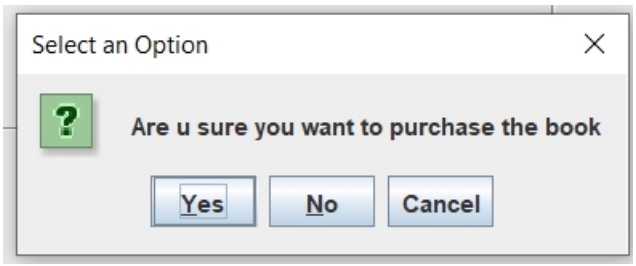
User entering details to search book :



Then the software displays the table of book info :



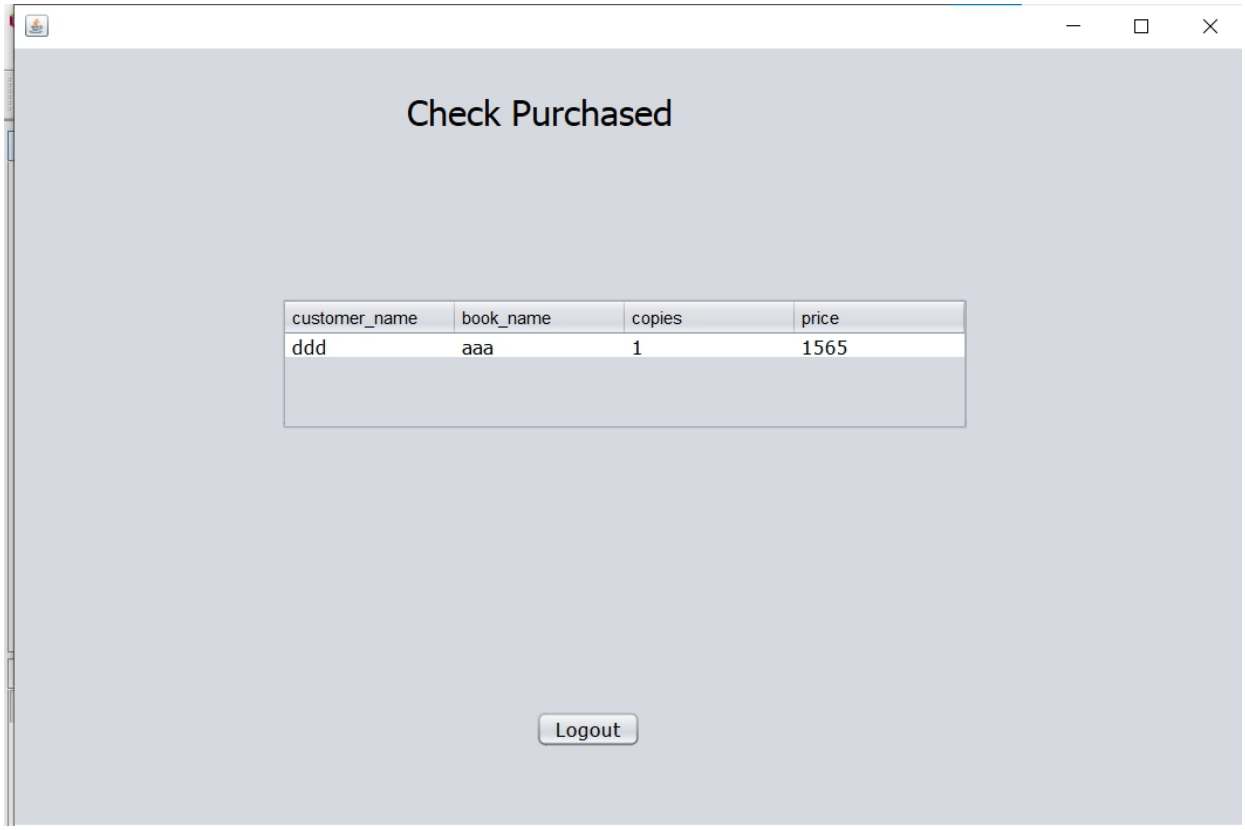
A confirmation window appears to purchase the book or not :



Then the purchased book info . is updated to it's database :

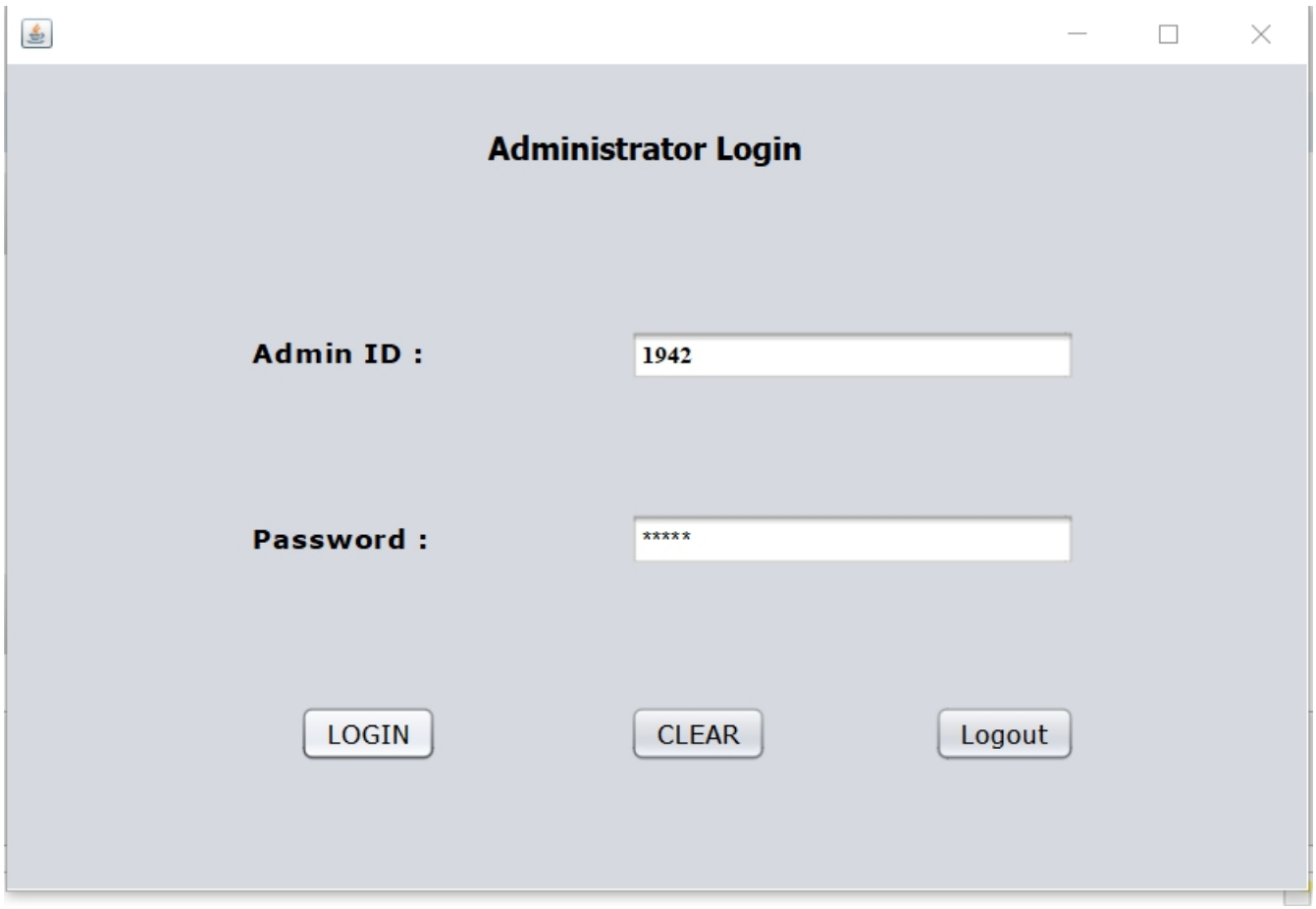
customer_name	book_name	copies	price
ddd	aaa	1	1565
	aaa	1	1565

Also the customer can his history of purchases by selecting the option from user operation window :

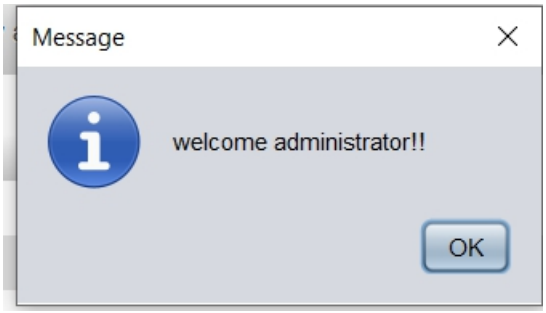


Admin :

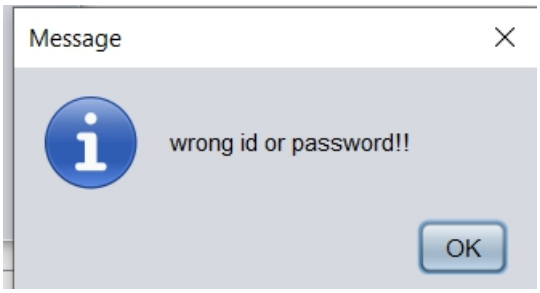
Login Frame :



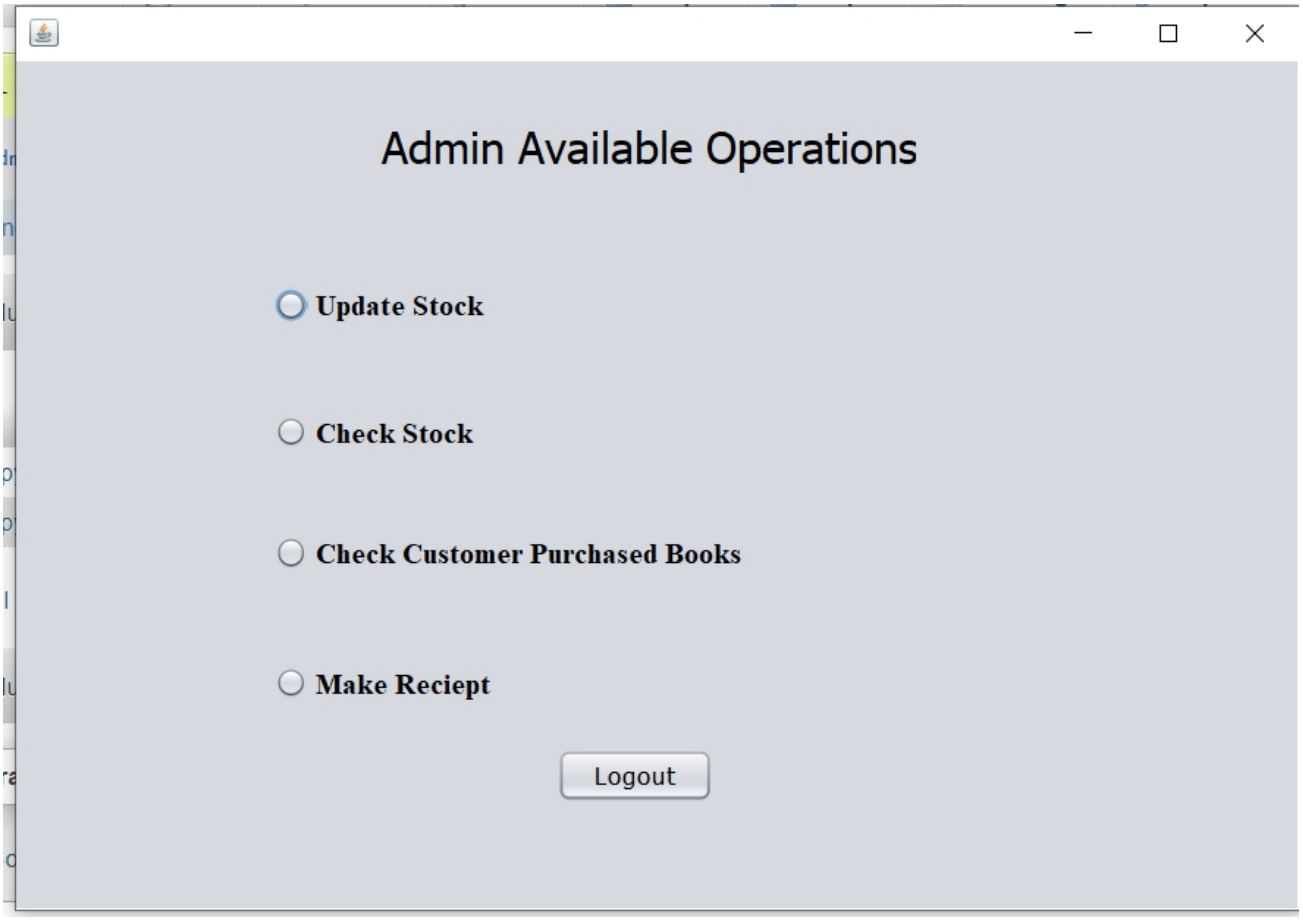
In case of correct login credentials :



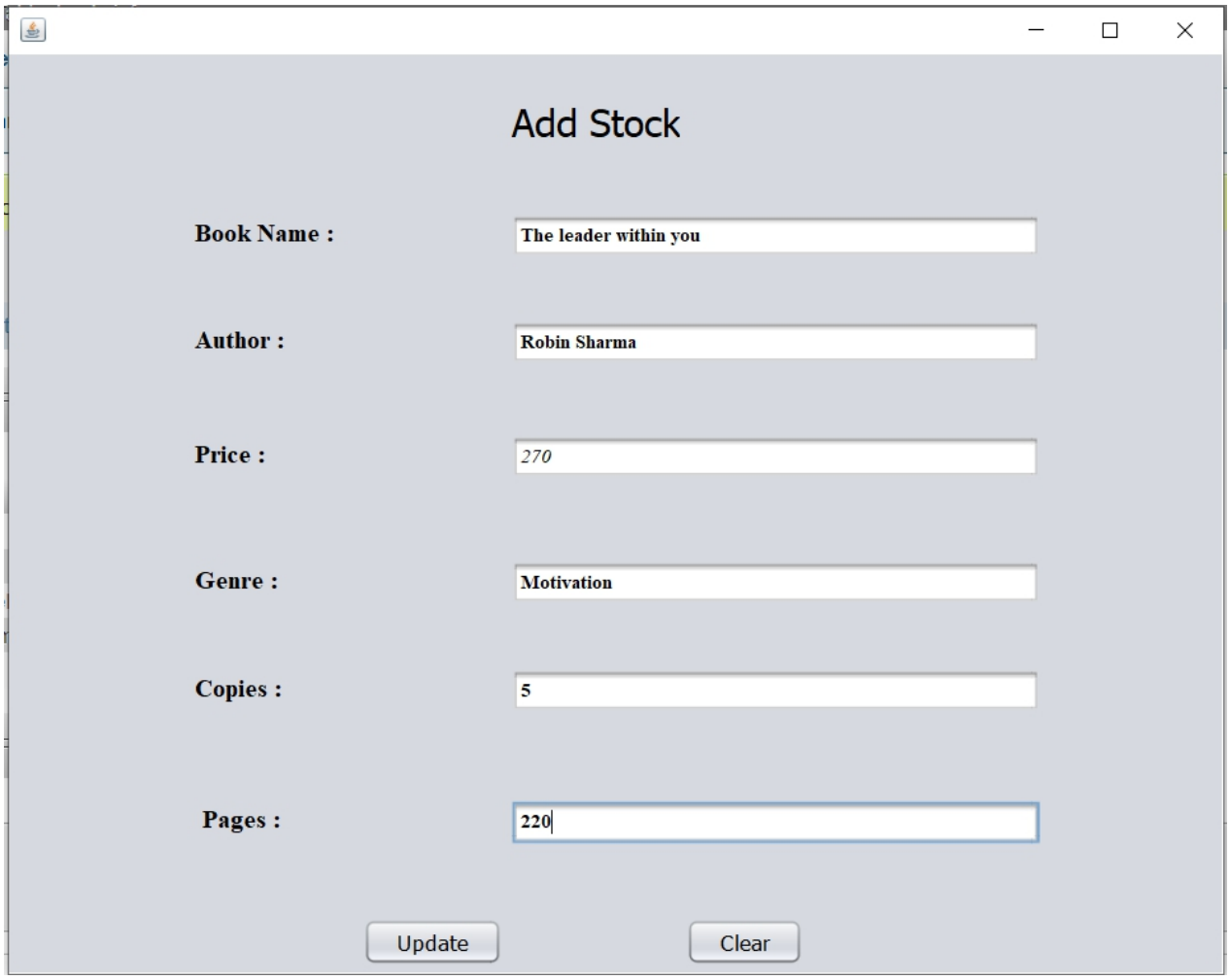
In case of wrong details entered :



Then after entering correct details admin has to select one of the following operations :



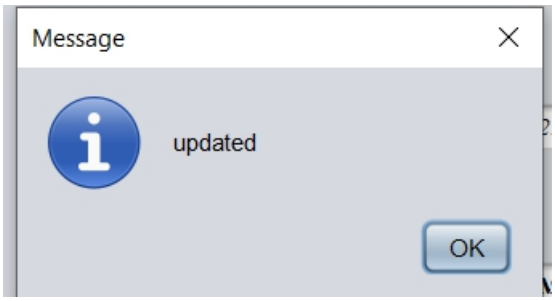
Admin updating Book Stock :



Before adding stock :

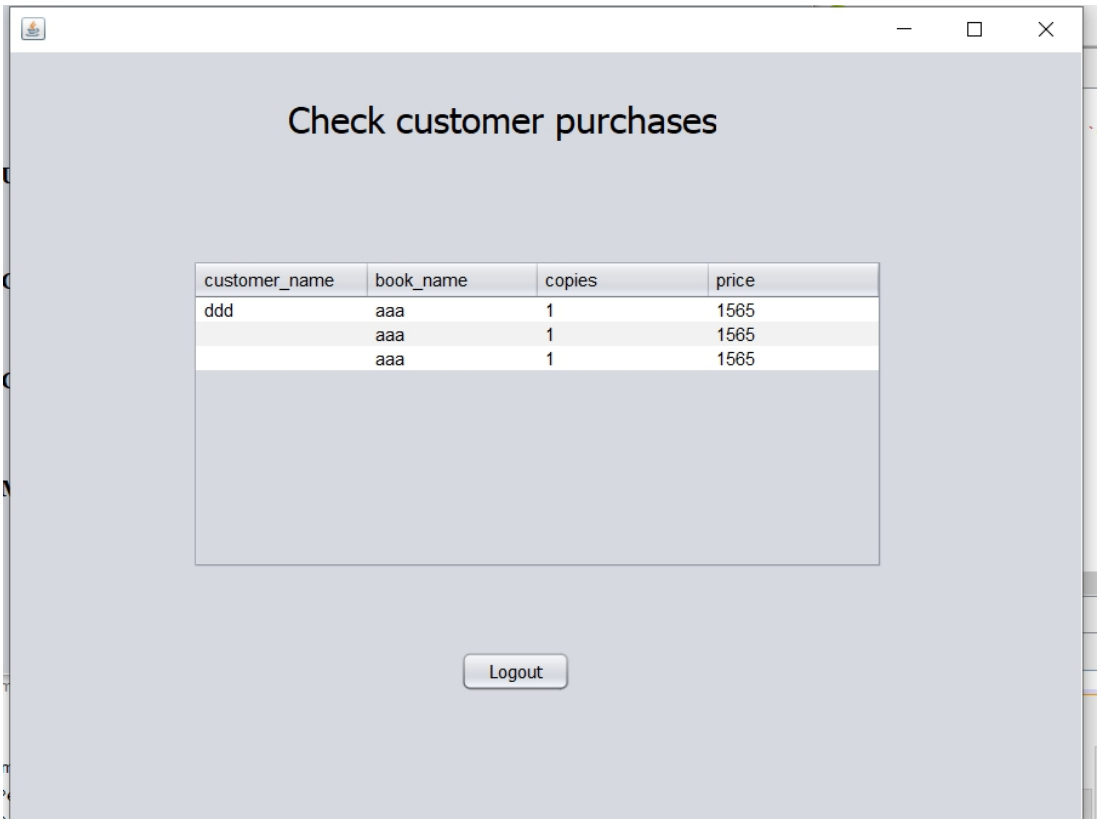
name	author	price	genre	copies	pages
Harry Potter and The Sorcerer's Stone	J.K.Rowling	1120	Fantasy	3	325
Harry Potter and The Chamber of Secrets	J.K.Rowling	870	Fantasy	4	270
The Alchemist	Paulo Colhelo	350	Adventure	5	165
The leader within you	Robin Sharma	270	Motivation	4	220
aaa	bbb	1565	scas	492	155

After adding new stock :

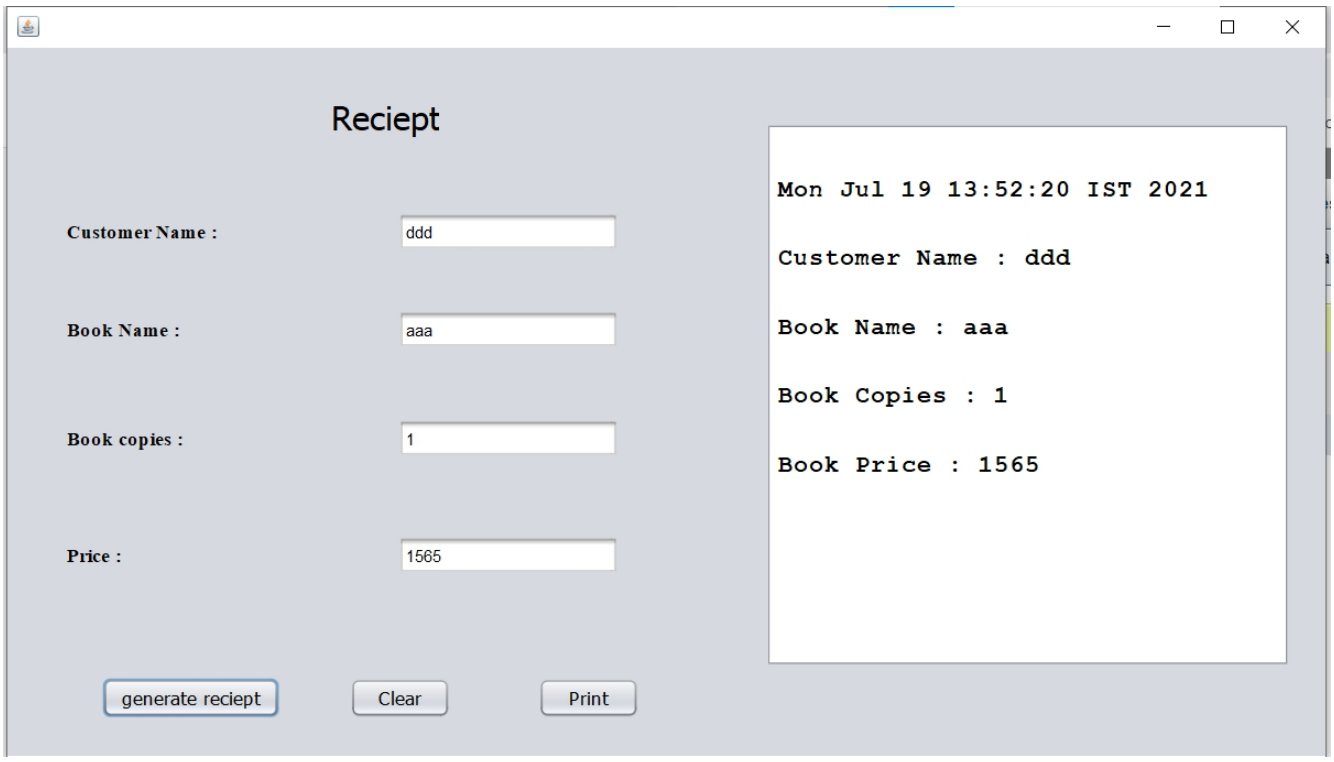


name	author	price	genre	copies	pages
Harry Potter and The Sorcerer's Stone	J.K.Rowling	1120	Fantasy	3	325
Harry Potter and The Chamber of Secrets	J.K.Rowling	870	Fantasy	4	270
The Alchemist	Paulo Colhelo	350	Adventure	5	165
The leader within you	Robin Sharma	270	Motivation	14	220
aaa	bbb	1565	scas	492	155

After checking stock from admin operations :



Admin selecting make receipt option from admin operations where the admin has to manually type the customer details and can also print the receipt :



Downloaded Receipt :

Mon Jul 19 13:52:20 IST 2021

Customer Name : ddd

Book Name : aaa

Book Copies : 1

Book Price : 1565

Test Cases For Online Book Management System

ENTRY WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1	On the Click of Customer	Software should display login credentials.	Successful
2	On the Click of Administrator	Software should display login credentials.	successful

USER OPERATION WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1	On the Click of SEARCH BOOK	Software should display search book frame.	Successful
2	On the Click of CHECK PURCHASED BOOK	Software should display table of books purchased by a customer.	successful

ADMIN OPERATION WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1	On the Click of UPDATE STOCK button	Software should display update Stock window.	Successful
2	On the Click	Software should display check	successful

	of CHECK STOCK button	stock window.	
3	On the click of CHECK CUSTOMER PURCHASED button	Software should redirect to the check customer purchased window.	successful
4	On the click of MAKE RECIEPT button	Software should redirect to the make receipt window.	

LOGIN WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1	Enter User ID , password and click on login button	Software should navigate to user/admin operations.	successful
2	Enter invalid credentials	Software should display warning message and display form to be filled.	successful
4	In case of invalid entries enter valid credentials again	Software should display user/admin operation frame.	successful

SEARCH BOOK WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1.	On the click of NEXT button	Software should display a window consisting of table which comprises of book information.	successful
2.	On the click of PURCHASE button	Software displays a window of confirmation whether to buy the book or not	successful
3.	On the click of YES button	Software appends the purchases to the database	successful

UPDATE STOCK WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1	On the click of UPDATE button	Software displays the window that updating the stock is completed .	successful
2 .	On the click of CLEAR button	Software clears all the fields entered by the administrator .	successful

CHECK STOCK WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1		Software should display table of existing book information from the database on the click of check stock from admin operations .	successful
2	On the click of RETURN button	Software displays admin operations window .	successful

CHECK PURCHASED BOOKS WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1.		Software should display a window consisting of table which comprises of books and copies which are purchased by customer on the click of check purchased books from admin operation window.	successful

RECIEPT WINDOW :

SL.No	Test Case	Excepted Result	Test Result
1	On the Click of Generate Receipt Button	Software should displays a text window alongside the fields which are entered by the admin with all the details necessary , and the person can download the receipt if necessary.	successful