# RIPHAH INTERNATIONAL UNIVERSITY

## Faculty of Computing

## Semester Project Report - Fall 2022

<Book Shop Inventory Management >

## Project Team

| Full Name of Student | SAP-ID | Program | Contact Number | Email Address |
|---|---|---|---|---|
| M.Huzaifa Khan | 46878 | SE | 03115888641 | hk9349881@gmail.com |
| | | | | |
| | | | | |

**Project Supervisor:** **SAAD WAZIR**
(Lecturer)

# Introduction / abstract

In this project "Book Shop Inventory Management" I am going to make a management system for book Shop. This project manages all the stock record which is present in the shop.

- **Problem identification**

  As we know that "Book Shop Inventory Management" was bit difficult as manually we create this project so that the user easily fulfills all the requirements and papers to manage inventory.

- **Description**

  Here we are try to develop such type system which is provide the automation on the any type of the bookshop. That means a shop which has the type system which provides the facility to the customers of the shop to purchase the books from the shop without any complexity. For example, any customer wants to purchase any book from the shop than first of all customer just choose the stream of the book than he/she can see the more than one type of books there and then he/she can choose the specific book from there. And then purchase it by paying price on bookshop cash counter and receives its invoice.

  OOP takes into consideration the making of reusable, particular code that can be effectively expanded and kept up with. By coordinating the "Book Shop Inventory Management" the executives framework as a bunch of interconnected objects, it becomes more straightforward to add new highlights and make changes to the framework depending on the situation.

# System features

There are many features in the System Like :

- User Should be able to Login
- User Should be able to see the Main Menu
- User Should be able to clock on any option
- User Should be able to add category and see category on table
- User Should be able to add brand and see brands on table
- User Should be able to add product and see products on table
- User Should be able to generate bill
- Admin should also be able to add new user

## 1. List of inputs to the system

- User name
- Password
- Category Name / Status
- Brand Name / Status
- Product
- Cost Price
- Retail Price

- Quantity
- Description
- Product Code
- Paid Amount
- (ADD, EDIT, DELETE)-BUTTONS

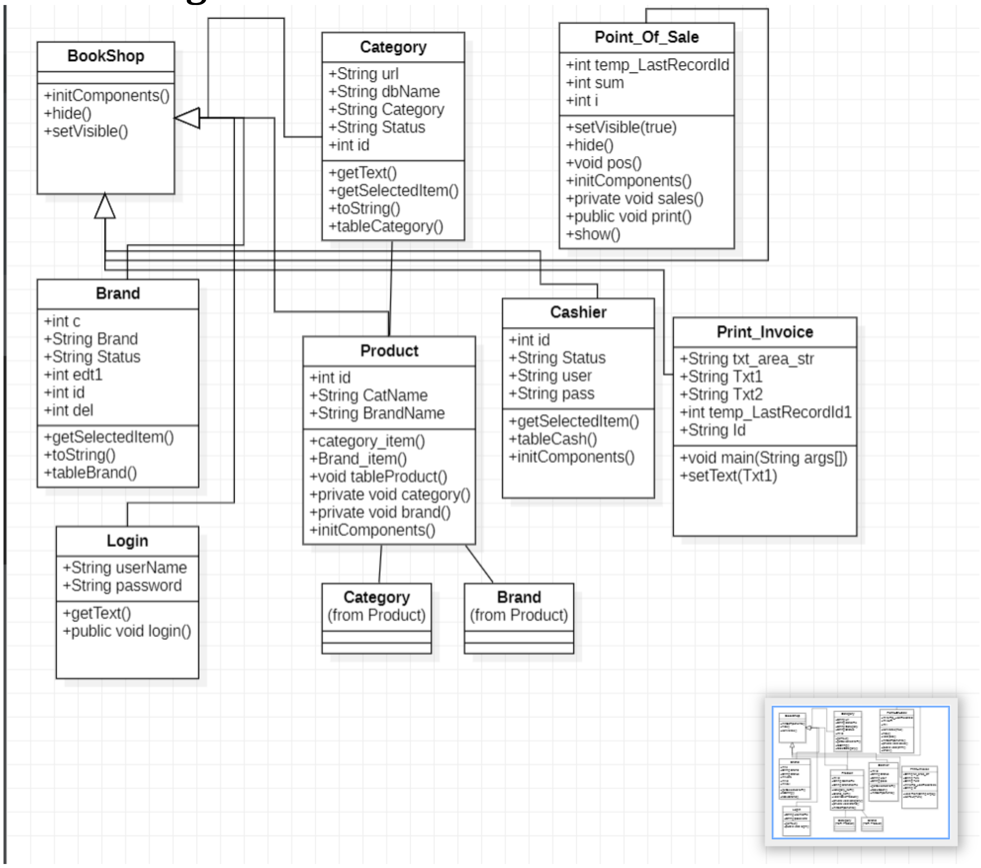
## 2. List of output from the system

- Data shown on Category Table
- Data shown on Brand Table
- Data shown on Product Table
- Data shown on Point Of Sale Table
- Data shown on Cashier Table

# Design phase

The design is made through Star UML

- ## UML Diagram



# Coding phase

```
private void tableCategory()
  {
     int c;
      try {
          con1 =DriverManager.getConnection("jdbc:mysql://localhost/bookshop","root","");
```

```
       pst=con1.prepareStatement("select * from category ");
       ResultSet rs=pst.executeQuery();
       ResultSetMetaData rsd=rs.getMetaData();
       c=rsd.getColumnCount();
       DefaultTableModel d=  (DefaultTableModel)TableCat.getModel();
       d.setRowCount(0);
       while(rs.next())
       {
         Vector v2 =new Vector();
         for(int i =1; i<=c; i++)
         {
           v2.add(rs.getString( "Id"));
           v2.add(rs.getString( "Category"));
           v2.add(rs.getString( "Status"));
         }
         d.addRow(v2);
       }




     } catch (SQLException ex) {
       Logger.getLogger(Category.class.getName()).log(Level.SEVERE, null, ex);
     }

  }

private void tableBrand()
  {
    int c;
     try {
       con1 =DriverManager.getConnection("jdbc:mysql://localhost/bookshop","root","");

       pst=con1.prepareStatement("select * from brand ");
       ResultSet rs=pst.executeQuery();
       ResultSetMetaData rsd=rs.getMetaData();
       c=rsd.getColumnCount();
       DefaultTableModel d=  (DefaultTableModel)TableBrand.getModel();
       d.setRowCount(0);
       while(rs.next())
       {
         Vector v2 =new Vector();
         for(int i =1; i<=c; i++)
         {
           v2.add(rs.getString( "Id"));
           v2.add(rs.getString( "Brand"));
           v2.add(rs.getString( "Status"));
         }
         d.addRow(v2);
       }




     } catch (SQLException ex) {
       Logger.getLogger(Brand.class.getName()).log(Level.SEVERE, null, ex);
     }

  }

private void sales()
    {
     DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd/MM/yyyy");
     LocalDateTime now=LocalDateTime.now();
     String date= dtf.format(now);

     String subTotal=TxtTotalPos.getText();
     String paid =TxtPaidPos.getText();
     String balance=TxtBalancePos.getText();
     int LastInsertId=0;

      try {
       con1 =DriverManager.getConnection("jdbc:mysql://localhost/bookshop","root","");
       String query="insert into sales ( Date ,Total_Amount, Paid_Amount,Balance) values ( ?,?,?,?) ";
       pst=con1.prepareStatement(query,Statement.RETURN_GENERATED_KEYS);
       pst.setString(1, date);
       pst.setString(2, subTotal);
       pst.setString(3, paid);
       pst.setString(4, balance);
       pst.executeUpdate();
       ResultSet last_id=pst.getGeneratedKeys();

       if(last_id.next())
       {
         //telling the last id of the table inserted
         LastInsertId=last_id.getInt(1);
         temp_LastRecordId = LastInsertId;
        //JOptionPane.showMessageDialog(this,LastInsertId);
       }
       //Record Saved into Sale_Product Table Code
       int rows =TablePos.getRowCount();
       String query1="insert into sales_product ( Sales_id ,Product_id, Sale_price,Qty,Total) values (
?,?,?,?,?) ";
             pst=con1.prepareStatement(query1);
             String product_Code="";
             String Price="";
             String Quantity="";
                 int toTal=0;
             for (int i=0; i<TablePos.getRowCount();i++)
             {
                 product_Code=(String)TablePos.getValueAt(i, 0);
                 Price=(String)TablePos.getValueAt(i, 2);
                 Quantity=(String)TablePos.getValueAt(i, 3);
                 toTal=(int)TablePos.getValueAt(i, 4);
                 pst.setInt(1, LastInsertId);
                 pst.setString(2, product_Code);
                 pst.setString(3, Price);
                 pst.setString(4, Quantity);
                 pst.setInt(5,  toTal);
                 pst.executeUpdate();
             }
       // query for update quantity
       String query2="Update product set Qty = Qty -? where Code=?";
       pst=con1.prepareStatement(query2);

       // forloop for qty
             for (int i=0; i<TablePos.getRowCount();i++)
       {
         product_Code=(String)TablePos.getValueAt(i, 0);
         Quantity=(String)TablePos.getValueAt(i, 3);
         pst.setString(1, Quantity);
         pst.setString(2, product_Code);
         pst.executeUpdate();
       }




       pst.addBatch();
      //JOptionPane.showMessageDialog(this,"Record Saved");
     } catch (SQLException ex) {
      Logger.getLogger(Brand.class.getName()).log(Level.SEVERE, null, ex);
     }


    }
    //Printing Function
     public void print()
    {

     Print_Invoice pobj =new Print_Invoice();


     pobj.show();


     try {
       con1 =DriverManager.getConnection("jdbc:mysql://localhost/bookshop","root","");
       String query_1="select * from sales where id=?";

       pst=con1.prepareStatement(query_1);

       pst.setString(1,Integer.toString(temp_LastRecordId));

       ResultSet rs=pst.executeQuery();

       while(rs.next())
       {

         System.out.println( pobj.txt_area_str = " ********* PRINT INVOCE ********* "+"\n"+"
******* Khan Book Depot ******** "+"\n"+" * Product Id :  "+Integer.toString( rs.getInt("Id") )+"\n"+"
* Date : "+rs.getString("Date")+"\n"+" * Total Amount :  " + rs.getInt("Total_Amount")+"\n"+" * Paid
Amount : "+Integer.toString( rs.getInt("Paid_Amount"))+"\n"+" * Remaining Amount : "+Integer.toString(
rs.getInt("Balance") )+"\n"+" ************************************************** ");

       }
       String product_Code="";
       String Price="";
       String Quantity="";
       int toTal=0;

     } catch (SQLException ex) {
      Logger.getLogger(Point_Of_Sale.class.getName()).log(Level.SEVERE, null, ex);
     }
     ;


    }
```

# Testing phase

| **Test Case1** | **Module** | **Input** | **Variable** | **Operation** | **Output/Error** |
|---|---|---|---|---|---|
| | Login | **Name**<br>**Password** | `String username`<br>`String password` | Name / password check if true then move further page<br>Password Stored in Database | **Output:** Login Successfully.<br>**Error:** If we enter wrong password we got error |

| **Test Case2** | **Module** | **Input** | **Variable** | **Operation** | **Output/Error** |
|---|---|---|---|---|---|
| | Category | Category **Enter**<br>**Status Selected** | `String Category`<br>`String Status` | Category and Status Saved in table and also in database | **Output:** Data show in Table |

| **Test Case3** | **Module** | **Input** | **Variable** | **Operation** | **Output/Error** |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| | Brand | Brand enter<br>**Status Selected PRESS ADD, DELETE, UPDATE BUTTON** | String brand<br>String Status | bRAND and Status ADDED DELETE EDIT in table and also in database | **Output:**<br>Data Add in table<br>Data edit in table<br>Data delete from table |

| **Test Case4** | **Module** | **Input** | **Variable** | **Operation** | **Output/Error** |
|---|---|---|---|---|---|
| | Brand | Product enter<br>**Piece and product code enter PRESS ADD, DELETE, UPDATE BUTTON** | String Product<br>Int product code<br>Int retail price | Product enter **Piece and product code enter** ADDED DELETE EDIT in table and also in database | **Output:**<br>Data Add in table<br>Data edit in table<br>Data delete from table |

| **Test Case5** | **Module** | **Input** | **Variable** | **Operation** | **Output/Error** |
|---|---|---|---|---|---|
| | BookShop | Clicking on any given option in Main Menu | No variables only classes are linked | initComponents();<br>this.lname=name;<br>usermainmenu.setText(lname); | **Output:**<br>Open specific page when click on any option |

## Conclusion

The conclusion of this project is that our project is created to help peoples with the “Book Shop Inventory Management” with an easy way. Basically this project can be published online and then can be useful for future use with the fusion of GUI interface. The working of this project is that first user will get a login interface the user will enter name and password menu will display for choosing options. Then program will ask for the requirement's needed.

I use the OOP in java language to make this system. The user is able to edit, delete, insert the record also use MYSQL to make database of project in which data is stored.

Bookshop Inventory Management is an application that helps to keep track of all the book and customer records.

## Project Motivation

- First of all, the reason to choose this project is to develop the skills making management systems and also able to learn how the GUI is made and work.
- The second thing is that we have book store I want to manage the record of each and every thing in the store. By successfully doing this project we have a benefit to manage our stock.

## Future directions

In the future, we will try to cover all the states and cities till now we have covered it at a very small scale. We will also introduce new features such as Author videos and reviews so that users can get more satisfaction about his deal.