# Book Publication System



R.H.M.S Osith Bandara Rajanayake

81 Batch

[osith94@gmail.com](mailto:osith94@gmail.com)

941581730v

Introduction

This documentation is about a Book Publication System that I made using JAVA programming language. I use NetBeans IDE 7.4 to develop this System and the software is based on JAVA language and I use SQL Query Browser to maintain and save data in this system.

This system is a primary level Book Publication System which we can manage few operation.

The whole system based on these main sections.

1. Authors
2. Books
3. Publication
4. Invoice
5. Settings
6. Search

First of all I will consider about these sections.

Author

From this interface we can manage Authors area in this manner, we can add new authors to the database, remove authors from data base that we entered and also we can keep this database as our stock.

Books

This section helps to manage the whole records of books that available in this Company. We can enter the ISBN number and we can keep this data base with connected Authors data base.

Publication

In this section we can manage all Outlets that we publish our products that mean or books we can do the calculations from this interface.

Invoice

According to our require there is an outlet in the Publishing Company customers can buy books from this Outlets. But we are not handling a customer data base to customers’ data because our main target is whole selling.

Search

There is a section to make quick search about the data base it helps you to make you work accuracy. We can search by the book name form this section then we can find the details about that specific section.

Setting

And I create a section to configure the user settings and make new users to the system.

I develop this Inventory Control System using JAVA language and this is an assessment given by PCJT Java Institute Colombo branch.

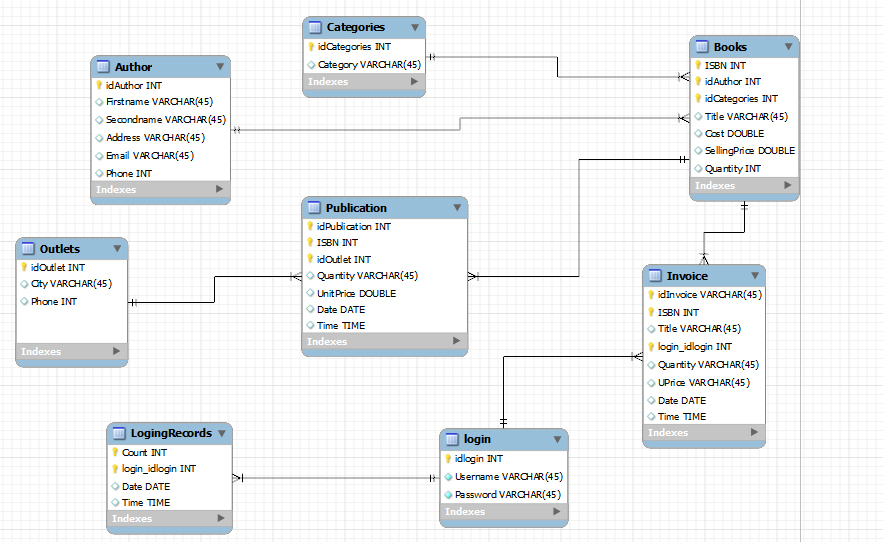
I gave my regrets to my Lecturer Mr.Isuru Randika for giving me knowledge to develop such a project using JAVA programming language.

**This System is Open to you Sir to check my faults,**

**Password – java**

**Username – java**

EER Diagram for Book Publication System



Login Interface

* **Username – java**
* **Password – java**

Login Interface Coding

String us;

String ps;

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

try {

Statement st1 = Db.getconneConnection().createStatement();

ResultSet rs = st1.executeQuery("SELECT \* FROM login WHERE username='"+txt\_username.getText()+"'");

while(rs.next()){

us = rs.getString("username");

ps = rs.getString("password");

}

String a = new String(txt\_password.getPassword());

if(us.equals(txt\_username.getText())){

if(ps.equals(a)){

main m = new main();

m.setVisible(true);

this.dispose();

}else{

JOptionPane.showMessageDialog(this,"Password Incorrect..!","Error",2);

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);

}

private void txt\_usernameKeyReleased(java.awt.event.KeyEvent evt) {

int i = evt.getKeyCode();

if(i==10){

txt\_password.grabFocus();

}

}

private void txt\_passwordKeyReleased(java.awt.event.KeyEvent evt) {

int i = evt.getKeyCode();

if(i==10){

try {

Statement st1 = Db.getconneConnection().createStatement();

ResultSet rs = st1.executeQuery("SELECT \* FROM login WHERE username='"+txt\_username.getText()+"'");

while(rs.next()){

us = rs.getString("username");

ps = rs.getString("password");

}

String a = new String(txt\_password.getPassword());

if(us.equals(txt\_username.getText())){

if(ps.equals(a)){

main m = new main();

m.setVisible(true);

this.dispose();

}else{

JOptionPane.showMessageDialog(this,"Password Incorrect..!","Error",2);

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

Main Menu (Main Screen)



Main Menu Coding

package book;

import com.sun.jmx.snmp.BerDecoder;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Date;

import java.sql.\*;

import java.sql.ResultSet;

import java.sql.Statement;

import java.sql.Types;

import java.text.SimpleDateFormat;

import java.util.Vector;

import javax.swing.ImageIcon;

import javax.swing.JOptionPane;

import javax.swing.Timer;

import javax.swing.table.DefaultTableModel;

/\*\*

\*

\* @author Acer i7

\*/

public class main extends javax.swing.JFrame {

Connection con;

PreparedStatement psss;

ResultSet rsss;

public main() {

initComponents();

LoadID();

frm();

pub();

IN();

settimer();

main.setVisible(true);

java.util.Date d = new java.util.Date();

SimpleDateFormat sdf = new SimpleDateFormat("yyyy:MM:dd");

String s = sdf.format(d);

lbl\_DATE.setText(s);

}

private void settimer() {

Timer t = new Timer(1000, new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

java.util.Date a = new java.util.Date();

SimpleDateFormat s = new SimpleDateFormat("HH:mm:ss");

String z = s.format(a);

lbl\_TIME.setText(z);

}

});

t.start();

}

void frm() {

Addnewbook.setVisible(false);

main.setVisible(true);

Setting.setVisible(false);

Publication.setVisible(false);

Authors.setVisible(false);

Invoice.setVisible(false);

Shops.setVisible(false);

Search.setVisible(false);

}

private void lbl\_addnewbookMouseClicked(java.awt.event.MouseEvent evt) {

Addnewbook.setVisible(true);

main.setVisible(false);

Setting.setVisible(false);

Publication.setVisible(false);

Authors.setVisible(false);

Invoice.setVisible(false);

Shops.setVisible(false);

Search.setVisible(false);

type();

DefaultTableModel dtm;

dtm = (DefaultTableModel) tb\_book.getModel();

dtm.setRowCount(0);

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("ISBN"));

v.add(rs.getString("authorid"));

v.add(rs.getString("name"));

v.add(rs.getString("type"));

v.add(rs.getString("cost"));

v.add(rs.getString("selling\_price"));

v.add(rs.getString("quantity"));

dtm.addRow(v);

}

} catch (Exception e) {

}

}

private void lbl\_shopMouseClicked(java.awt.event.MouseEvent evt) {

Addnewbook.setVisible(false);

main.setVisible(false);

Setting.setVisible(false);

Publication.setVisible(false);

Authors.setVisible(false);

Invoice.setVisible(false);

Shops.setVisible(true);

Search.setVisible(false);

DefaultTableModel dtm = (DefaultTableModel) tb\_outlet.getModel();

dtm.setRowCount(0);

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM outlets");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idoutlet"));

v.add(rs.getString("outletname"));

v.add(rs.getString("city"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void lbl\_settingMouseClicked(java.awt.event.MouseEvent evt) {

Addnewbook.setVisible(false);

main.setVisible(false);

Setting.setVisible(true);

Publication.setVisible(false);

Authors.setVisible(false);

Invoice.setVisible(false);

Shops.setVisible(false);

Search.setVisible(false);

}

private void lbl\_purchaseMouseClicked(java.awt.event.MouseEvent evt) {

Addnewbook.setVisible(false);

main.setVisible(false);

Setting.setVisible(false);

Publication.setVisible(true);

Authors.setVisible(false);

Invoice.setVisible(false);

Shops.setVisible(false);

Search.setVisible(false);

try {

DefaultTableModel dtm = (DefaultTableModel) tb\_publication.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM publication");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idpublication"));

v.add(rs.getString("ISBN"));

v.add(rs.getString("idoutlets"));

v.add(rs.getString("quantity"));

v.add(rs.getString("unit\_price"));

v.add(rs.getString("sub\_total"));

v.add(rs.getString("date"));

v.add(rs.getString("time"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

}

private void lbl\_invoiceMouseClicked(java.awt.event.MouseEvent evt) {

Addnewbook.setVisible(false);

main.setVisible(false);

Setting.setVisible(false);

Publication.setVisible(false);

Authors.setVisible(false);

Invoice.setVisible(true);

Shops.setVisible(false);

Search.setVisible(false);

}

private void lbl\_authorMouseClicked(java.awt.event.MouseEvent evt) {

Addnewbook.setVisible(false);

main.setVisible(false);

Setting.setVisible(false);

Publication.setVisible(false);

Authors.setVisible(true);

Invoice.setVisible(false);

Shops.setVisible(false);

Search.setVisible(false);

try {

DefaultTableModel dtm = (DefaultTableModel) tb\_author.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM authors");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idauthor"));

v.add(rs.getString("first\_name"));

v.add(rs.getString("last\_name"));

v.add(rs.getString("address"));

v.add(rs.getString("email"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

private void lbl\_addnewbookMouseEntered(java.awt.event.MouseEvent evt) {

lbl\_addnewbook.setVisible(true);

ImageIcon II = new ImageIcon(getClass().getResource("newBook2.jpg"));

lbl\_addnewbook.setIcon(II);

}

private void lbl\_addnewbookMouseExited(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("newBook1.jpg"));

lbl\_addnewbook.setIcon(II);

}

private void lbl\_shopMouseEntered(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("shop2.jpg"));

lbl\_shop.setIcon(II);

}

private void lbl\_shopMouseExited(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("shop1.jpg"));

lbl\_shop.setIcon(II);

}

private void lbl\_purchaseMouseEntered(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("purchase2.jpg"));

lbl\_purchase.setIcon(II);

}

private void lbl\_purchaseMouseExited(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("purchase1.jpg"));

lbl\_purchase.setIcon(II);

}

private void lbl\_invoiceMouseEntered(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("invoice1.jpg"));

lbl\_invoice.setIcon(II);

}

private void lbl\_invoiceMouseExited(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("invoice2.jpg"));

lbl\_invoice.setIcon(II);

}

private void lbl\_authorMouseEntered(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("author1.jpg"));

lbl\_author.setIcon(II);

}

private void lbl\_authorMouseExited(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("author2.jpg"));

lbl\_author.setIcon(II);

}

private void lbl\_settingMouseEntered(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("settings1.jpg"));

lbl\_setting.setIcon(II);

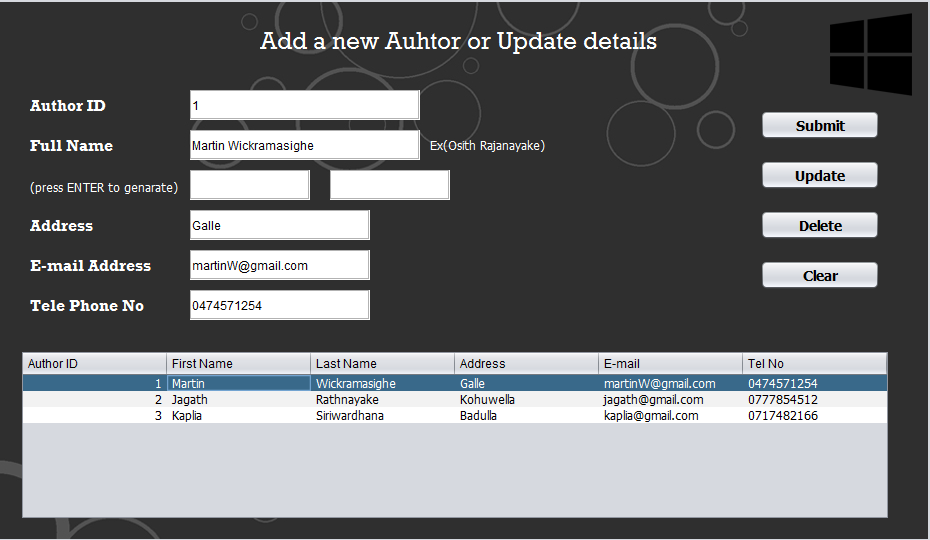
}

private void lbl\_settingMouseExited(java.awt.event.MouseEvent evt) {

ImageIcon II = new ImageIcon(getClass().getResource("settings2.jpg"));

lbl\_setting.setIcon(II);

Book Interface



Book Interface Coding

# Submit

try {

// ISBN, idauthor, name, type, cost, selling\_price, quantity

Db.getconneConnection().createStatement().executeUpdate("INSERT INTO books (ISBN,authorid,name,type,cost,selling\_price,quantity) VALUES ('" + txt\_ISBM.getText() + "','" + txt\_Author.getText() + "','" + txt\_Bookname.getText() + "','" + combo\_type.getSelectedItem().toString() + "','" + txt\_cost.getText() + "','" + txt\_SellPrice.getText() + "','" + txt\_Quantity.getText() + "')");

DefaultTableModel dtm = (DefaultTableModel) tb\_book.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("ISBN"));

v.add(rs.getString("name"));

v.add(rs.getString("idauthor"));

v.add(rs.getString("type"));

v.add(rs.getString("cost"));

v.add(rs.getString("selling\_price"));

v.add(rs.getString("quantity"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

# Update

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) tb\_book.getModel();

dtm.setRowCount(0);

try {

// ISBN, name, authorid, type, cost, selling\_price, quantity

Db.getconneConnection().createStatement().executeUpdate("UPDATE books SET name='" + txt\_Bookname.getText() + "',authorid='" + txt\_Author.getText() + "',type='" + combo\_type.getSelectedItem().toString() + "', cost='" + txt\_cost.getText() + "', selling\_price='" + txt\_SellPrice.getText() + "', quantity='" + txt\_Quantity.getText() + "' WHERE ISBN='" + txt\_ISBM.getText() + "'");

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("ISBN"));

v.add(rs.getString("authorid"));

v.add(rs.getString("name"));

v.add(rs.getString("type"));

v.add(rs.getString("cost"));

v.add(rs.getString("selling\_price"));

v.add(rs.getString("quantity"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_Author.setText("");

txt\_Bookname.setText("");

txt\_ISBM.setText("");

txt\_Quantity.setText("");

txt\_SellPrice.setText("");

txt\_cost.setText("");

combo\_type.removeAllItems();

type();

}

# Delete

try {

Db.getconneConnection().createStatement().executeUpdate("DELETE FROM books WHERE ISBN='" + txt\_ISBM.getText() + "'");

} catch (Exception e) {

e.printStackTrace();

}

DefaultTableModel dtm = (DefaultTableModel) tb\_book.getModel();

dtm.setRowCount(0);

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("ISBN"));

v.add(rs.getString("authorid"));

v.add(rs.getString("name"));

v.add(rs.getString("type"));

v.add(rs.getString("cost"));

v.add(rs.getString("selling\_price"));

v.add(rs.getString("quantity"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_Author.setText("");

txt\_Bookname.setText("");

txt\_ISBM.setText("");

txt\_Quantity.setText("");

txt\_SellPrice.setText("");

txt\_cost.setText("");

combo\_type.removeAllItems();

type();

}

# Clear

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

txt\_Author.setText("");

txt\_Bookname.setText("");

txt\_ISBM.setText("");

txt\_Quantity.setText("");

txt\_SellPrice.setText("");

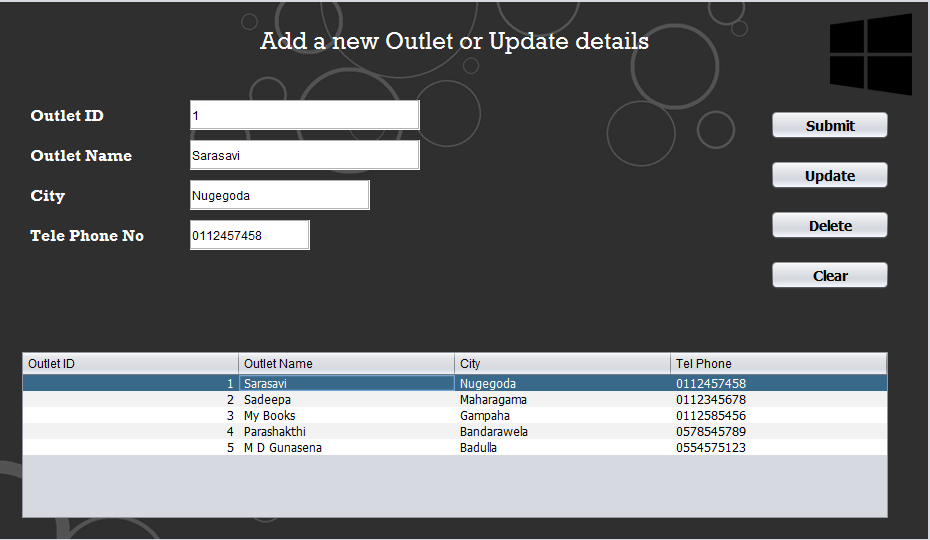
txt\_cost.setText("");

combo\_type.removeAllItems();

type();

}

Add new Outlet Interface



# Submit

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

try {

// idoutlet, outletname, city, phone

Db.getconneConnection().createStatement().executeUpdate("INSERT INTO outlets (idoutlet,outletname,city,phone) VALUES ('" + txt\_Oid.getText() + "','" + txt\_Oname.getText() + "','" + txt\_Ocity.getText() + "','" + txt\_Otel.getText() + "')");

DefaultTableModel dtm = (DefaultTableModel) tb\_outlet.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM outlets");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idoutlet"));

v.add(rs.getString("outletname"));

v.add(rs.getString("city"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

}

# Update

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) tb\_outlet.getModel();

dtm.setRowCount(0);

try {

// idoutlet, outletname, city, phone

Db.getconneConnection().createStatement().executeUpdate("UPDATE outlets SET outletname='" + txt\_Oname.getText() + "',city='" + txt\_Ocity.getText() + "',phone='" + txt\_Otel.getText() + "' WHERE idoutlet='" + txt\_Oid.getText() + "'");

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM outlets");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idoutlet"));

v.add(rs.getString("outletname"));

v.add(rs.getString("city"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_Ocity.setText("");

txt\_Oid.setText("");

txt\_Otel.setText("");

txt\_Oname.setText("");

}

# Delete

try {

// idoutlet, outletname, city, phone

Db.getconneConnection().createStatement().executeUpdate("DELETE FROM outlets WHERE idoutlet='" + txt\_Oid.getText() + "'");

} catch (Exception e) {

e.printStackTrace();

}

DefaultTableModel dtm = (DefaultTableModel) tb\_outlet.getModel();

dtm.setRowCount(0);

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM outlets");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idoutlet"));

v.add(rs.getString("outletname"));

v.add(rs.getString("city"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_Ocity.setText("");

txt\_Oid.setText("");

txt\_Otel.setText("");

txt\_Oname.setText("");

}

# Clear

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

txt\_Ocity.setText("");

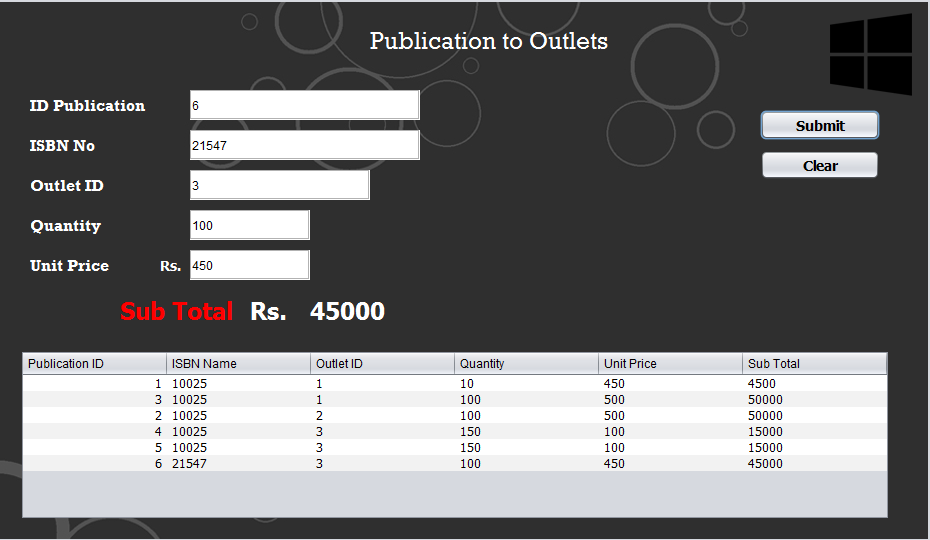
txt\_Oid.setText("");

txt\_Otel.setText("");

txt\_Oname.setText("");

}

Purchase Interface



# Submit

int q;

String isbn;

private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {

try {

isbn = txt\_Pisbn.getText();

q = Integer.parseInt(txt\_Pquantity.getText());

int p = Integer.parseInt(txt\_PunitP.getText());

int tot = (p) \* (q);

lbl\_PSubtotal.setText(tot + "");

// idpublication, ISBN, idoutlets, quantity, unit\_price, sub\_total, date, time

Db.getconneConnection().createStatement().executeUpdate("INSERT INTO publication (idpublication, ISBN, idoutlets, quantity, unit\_price, sub\_total, date, time) VALUES ('" + txt\_PID.getText() + "','" + txt\_Pisbn.getText() + "','" + txt\_Poutid.getText() + "','" + q + "','" + txt\_PunitP.getText() + "','" + lbl\_PSubtotal.getText() + "','" + lbl\_DATE.getText() + "','" + lbl\_TIME.getText() + "')");

DefaultTableModel dtm = (DefaultTableModel) tb\_publication.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM publication");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idpublication"));

v.add(rs.getString("ISBN"));

v.add(rs.getString("idoutlets"));

v.add(rs.getString("quantity"));

v.add(rs.getString("unit\_price"));

v.add(rs.getString("sub\_total"));

v.add(rs.getString("date"));

v.add(rs.getString("time"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books WHERE ISBN='" + isbn + "' ");

while (rs.next()) {

int qun = rs.getInt("quantity");

int nowq = (qun) - (q);

String now = nowq + "";

Db.getconneConnection().createStatement().executeUpdate("UPDATE books SET quantity ='" + now + "' WHERE ISBN='" + isbn + "' ");

}

} catch (Exception e) {

e.printStackTrace();

}

}

# Clear

txt\_PID.setText("");

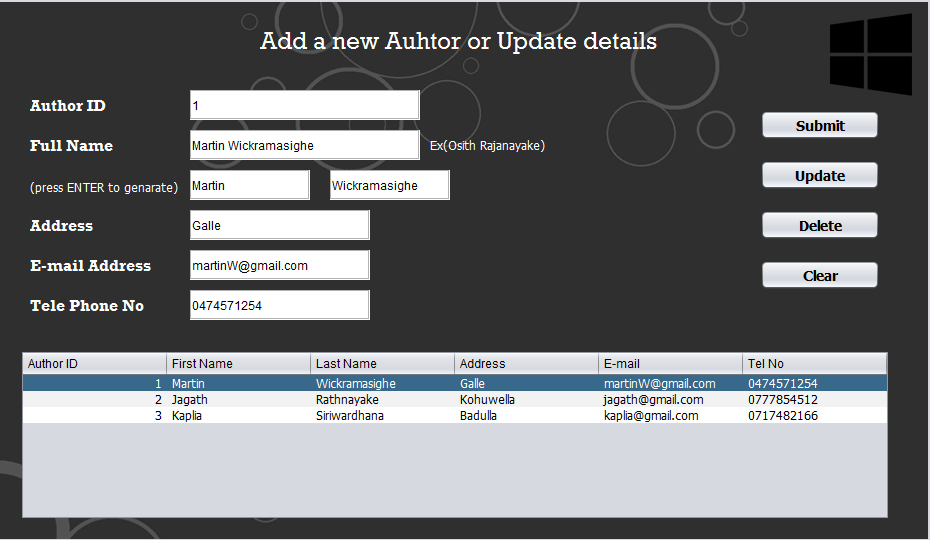
txt\_Pisbn.setText("");

txt\_Poutid.setText("");

txt\_Pquantity.setText("");

txt\_PunitP.setText("");

Authors Interface



# Submit

private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {

try {

// idauthor, first\_name, last\_name, address, email, phone

Db.getconneConnection().createStatement().executeUpdate("INSERT INTO authors (idauthor, first\_name, last\_name, address, email, phone) VALUES ('" + txt\_Aid.getText() + "','" + txt\_AFirstName.getText() + "','" + txt\_ALastName.getText() + "','" + txt\_AAdd.getText() + "','" + txt\_AEmail.getText() + "','" + txt\_Atel.getText() + "')");

DefaultTableModel dtm = (DefaultTableModel) tb\_author.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM authors");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idauthor"));

v.add(rs.getString("first\_name"));

v.add(rs.getString("last\_name"));

v.add(rs.getString("address"));

v.add(rs.getString("email"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

}

# Update

DefaultTableModel dtm = (DefaultTableModel) tb\_author.getModel();

dtm.setRowCount(0);

try {

// idauthor, first\_name, last\_name, address, email, phone

Db.getconneConnection().createStatement().executeUpdate("UPDATE authors SET first\_name='" + txt\_AFirstName.getText() + "',last\_name='" + txt\_ALastName.getText() + "',address='" + txt\_AAdd.getText() + "',email='" + txt\_AEmail.getText() + "', phone='" + txt\_Atel.getText() + "' WHERE idauthor='" + txt\_Aid.getText() + "'");

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM authors");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idauthor"));

v.add(rs.getString("first\_name"));

v.add(rs.getString("last\_name"));

v.add(rs.getString("address"));

v.add(rs.getString("email"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_FF.setText("");

txt\_AAdd.setText("");

txt\_AEmail.setText("");

txt\_AFirstName.setText("");

txt\_ALastName.setText("");

txt\_Aid.setText("");

txt\_Atel.setText("");

}

# Delete

private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel dtm = (DefaultTableModel) tb\_author.getModel();

dtm.setRowCount(0);

try {

// idauthor, first\_name, last\_name, address, email, phone

Db.getconneConnection().createStatement().executeUpdate("DELETE FROM authors WHERE idauthor='" + txt\_Aid.getText() + "'");

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM authors");

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("idauthor"));

v.add(rs.getString("first\_name"));

v.add(rs.getString("last\_name"));

v.add(rs.getString("address"));

v.add(rs.getString("email"));

v.add(rs.getString("phone"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_FF.setText("");

txt\_AAdd.setText("");

txt\_AEmail.setText("");

txt\_AFirstName.setText("");

txt\_ALastName.setText("");

txt\_Aid.setText("");

txt\_Atel.setText("");

}

# Clear

private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {

txt\_AAdd.setText("");

txt\_AEmail.setText("");

txt\_AFirstName.setText("");

txt\_ALastName.setText("");

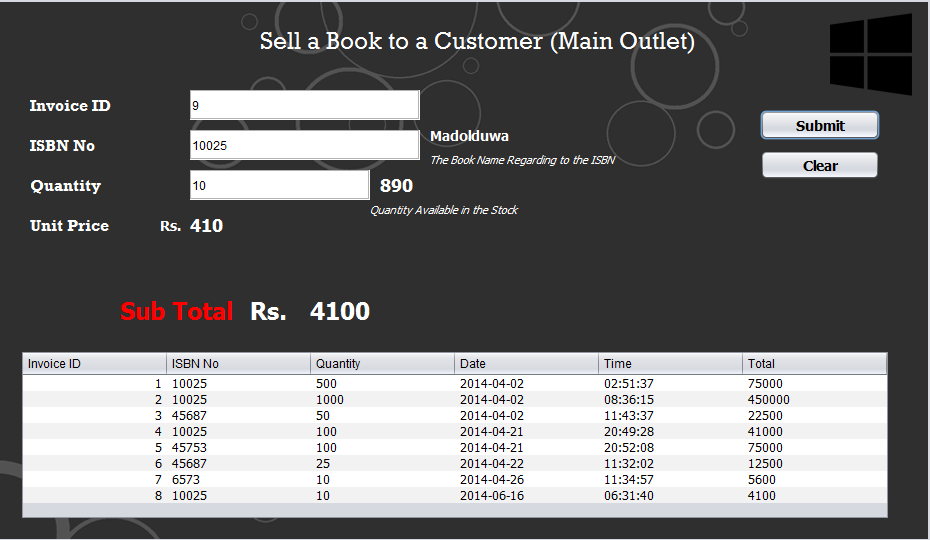
txt\_Aid.setText("");

txt\_Atel.setText("");

txt\_FF.setText("");

}

Invoice Interface



# Submit

String Iisbn;

int IIIq;

private void jButton21ActionPerformed(java.awt.event.ActionEvent evt) {

try {

Iisbn = txt\_IIisbn.getText();

IIIq = Integer.parseInt(txt\_IIqua.getText());

int p = Integer.parseInt(txt\_IIunit.getText());

int tot = (p) \* (IIIq);

lbl\_IIsubtotal.setText(""+tot);

// idivoice, books\_ISBN, quantity, date, time, total

Db.getconneConnection().createStatement().executeUpdate("INSERT INTO sale (idivoice, books\_ISBN, quantity, date, time, total) VALUES ('" + txt\_IIid.getText() + "','" + Iisbn + "','" + IIIq + "','" + lbl\_DATE.getText() + "','" + lbl\_TIME.getText() + "','" + lbl\_IIsubtotal.getText() + "')");

DefaultTableModel dtm = (DefaultTableModel) tb\_invoice.getModel();

dtm.setRowCount(0);

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM sale");

while (rs.next()) {

Vector v = new Vector();

//idivoice, books\_ISBN, quantity, date, time, total

v.add(rs.getString("idivoice"));

v.add(rs.getString("books\_ISBN"));

v.add(rs.getString("quantity"));

v.add(rs.getString("date"));

v.add(rs.getString("time"));

v.add(rs.getString("total"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books WHERE ISBN='" + Iisbn + "' ");

while (rs.next()) {

int qun = rs.getInt("quantity");

int nowq = (qun) - (IIIq);

String now = ""+nowq;

Db.getconneConnection().createStatement().executeUpdate("UPDATE books SET quantity ='" + now + "' WHERE ISBN='" + Iisbn + "' ");

}

} catch (Exception e) {

e.printStackTrace();

}

txt\_IIid.setText("");

txt\_IIisbn.setText("");

txt\_IIqua.setText("");

txt\_IIunit.setText("");

}

# Clear

private void jButton24ActionPerformed(java.awt.event.ActionEvent evt) {

txt\_IIid.setText("");

txt\_IIisbn.setText("");

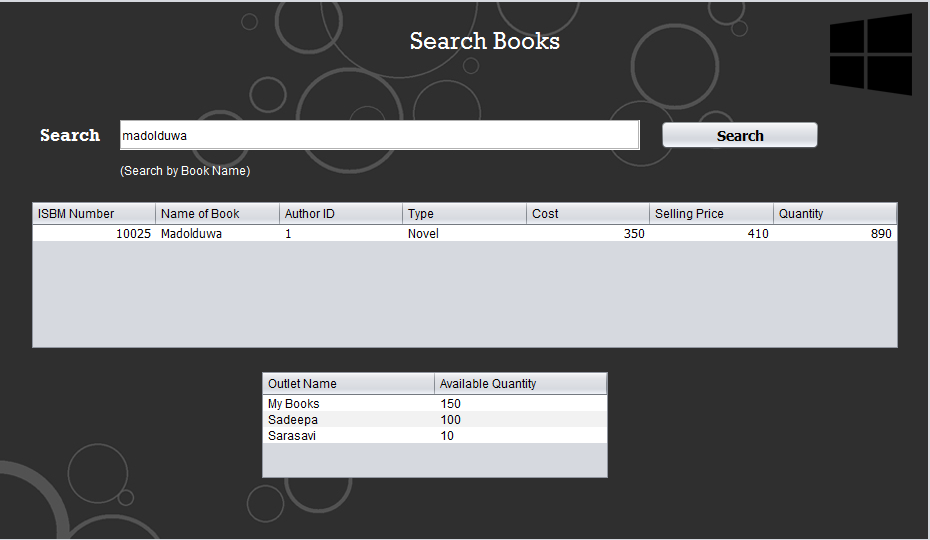
lbl\_IIsubtotal.setText("");

txt\_IIqua.setText("");

txt\_IIunit.setText("");

}

Search Interface



# Search (Key Pressed & Search)

private void txt\_SERKeyReleased(java.awt.event.KeyEvent evt) {

try {

String s = txt\_SER.getText();

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT o.outletname, publication.quantity FROM `book`.`publication` LEFT OUTER JOIN book.outlets o ON publication.idoutlets=o.idoutlet LEFT OUTER JOIN book.books b ON publication.ISBN=b.ISBN WHERE b.name = '"+ txt\_SER.getText() + "' GROUP BY o.outletname");

// ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM `book`.`publication` LEFT OUTER JOIN book.outlets o ON publication.idoutlets=o.idoutlet LEFT OUTER JOIN book.books b ON publication.ISBN=b.ISBN WHERE b.name ='" + txt\_SER.getText() + "'");

DefaultTableModel dtm = (DefaultTableModel) tb\_out.getModel();

dtm.setRowCount(0);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("outletname"));

v.add(rs.getString("quantity"));

dtm.addRow(v);

}

} catch (Exception e) {

e.printStackTrace();

}

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT \* FROM books WHERE name LIKE '"+txt\_SER.getText()+"%' ");

DefaultTableModel dtm = (DefaultTableModel) tb\_SEARCH.getModel();

dtm.setRowCount(0);

while (rs.next()) {

Vector v = new Vector();

v.add(rs.getString("ISBN"));

v.add(rs.getString("name"));

v.add(rs.getString("authorid"));

v.add(rs.getString("type"));

v.add(rs.getString("cost"));

v.add(rs.getString("selling\_price"));

v.add(rs.getString("quantity"));

dtm.addRow(v);

}

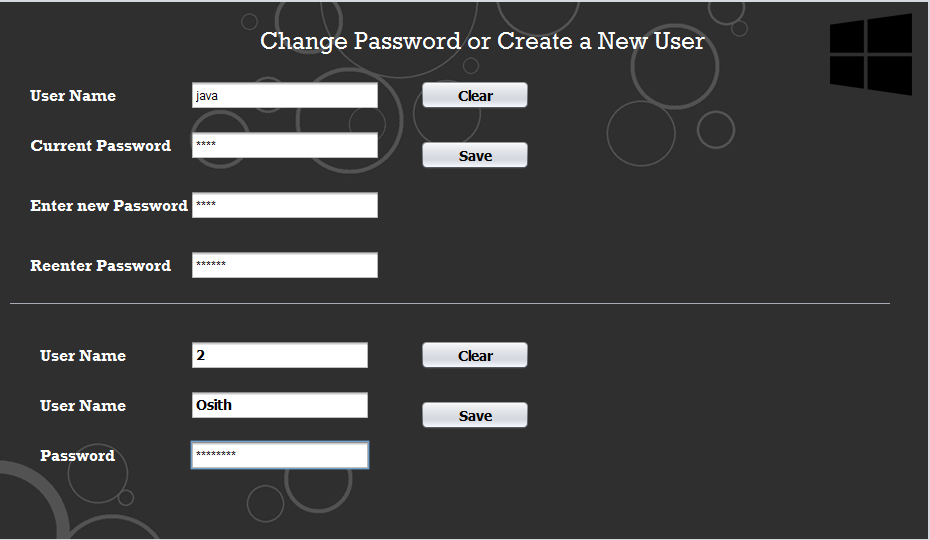
} catch (Exception e) {

e.printStackTrace();

}

}

Settings



Change Password

# Clear

private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {

txt\_PScurrent.setText("");

txt\_PSnew.setText("");

txt\_PSreenter.setText("");

txt\_Username.setText("");

}

# Save

String US;

String CPP;

private void jButton17ActionPerformed(java.awt.event.ActionEvent evt) {

String CU = new String(txt\_PScurrent.getPassword());

try {

Statement st = Db.getconneConnection().createStatement();

ResultSet rs = st.executeQuery("SELECT \* FROM login WHERE username='" + txt\_Username.getText() + "'");

while (rs.next()) {

US = rs.getString("username");

CPP = rs.getString("password");

}

String PS = new String(txt\_PSnew.getPassword());

if (CPP.equals(CU)) {

String RE = new String(txt\_PSreenter.getPassword());

if (PS.equals(RE)) {

Statement st1 = Db.getconneConnection().createStatement();

st1.executeUpdate("UPDATE login SET password='" + RE + "' WHERE username='" + txt\_Username.getText() + "'");

JOptionPane.showMessageDialog(this, "You have Changed your Password", "Change Password", 1);

}

} else {

JOptionPane.showMessageDialog(this, "Invalid Password", "Error", 2);

}

} catch (Exception e) {

e.printStackTrace();

}

}

Create a New User

# Submit

private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {

txt\_NEWPS.setText("");

txt\_newName.setText("");

}

# Clear

private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {

try {

Statement st = Db.getconneConnection().createStatement();

String pass = new String(txt\_NEWPS.getPassword());

st.executeUpdate("INSERT INTO login(username,password) VALUES('" + txt\_newName.getText() + "','" + pass + "')");

JOptionPane.showMessageDialog(this, "You have Create a NEW USER", "New User", 1);

} catch (Exception e) {

e.printStackTrace();

}

}

public void LoadID() {

try {

ResultSet rs = Db.getconneConnection().createStatement().executeQuery("SELECT MAX(loginid) FROM login");

if (rs.next()) {

int a = rs.getInt("MAX(loginid)") + 1;

String max = "" + a;

txt\_AAAAA.setText(max);

} else {

txt\_AAAAA.setText("1");

}

} catch (Exception e) {

e.printStackTrace();

}

}