

public StartPage() {

initComponents();

{

try {

for (LookAndFeelInfo info : UIManager.getInstalledLookAndFeels())

{

if ("Nimbus".equals(info.getName())) {

UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (Exception e) {

}

}

**For “Log in”**

String user = t1.getText();

String pass = new String(p1.getPassword());

String username;

pass = md5hash.passwordsalted(pass);

if (user.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please Enter your Username");

} else if (pass.equals("b1f4f9a523e36fd969f4573e25af4540")) {

JOptionPane.showMessageDialog(this, "Please Enter your Password");

} else {

try {

Statement stmt = myQueries.getStmt();

String query = "select username,password,type from users where username='" + user + "';";

ResultSet rs = stmt.executeQuery(query);

rs.next();

String DBuser = rs.getString("username");

String DBpassword = rs.getString("password");

String DBtype = rs.getString("type");

if (DBuser.equals(user) && DBpassword.equals(pass)) {

if (DBtype.equals("admin")) {

username = DBuser;

MainPage a = new MainPage(username);

a.setVisible(true);

this.dispose();

} else if (DBtype.equals("issuer")) {

username = DBuser;

MainPageissuer a = new MainPageissuer(username);

a.setVisible(true);

this.dispose();

}

} else {

JOptionPane.showMessageDialog(this, "Username or Password is Incorrect");

}

} catch (Exception e) {

if (e.getMessage().equals("Illegal operation on empty result set.")) {

JOptionPane.showMessageDialog(this, "No such Username Exists");

} else {

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

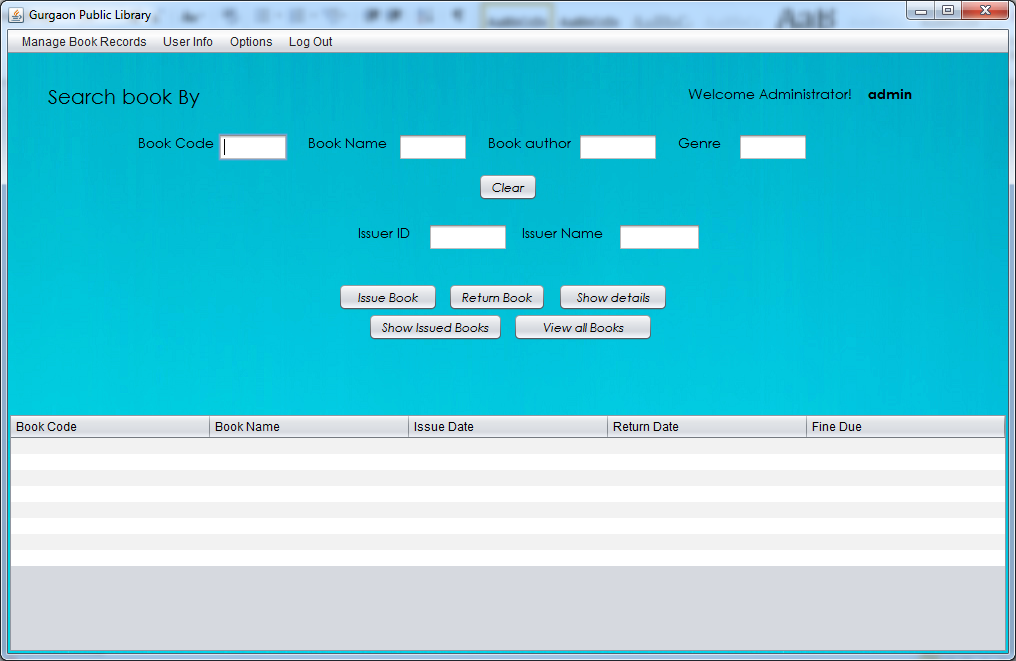
}

}

**For “SignUp “**

SignupIssuer a= new SignupIssuer();

a.setVisible(true);



**For “Show Issued Books”**

jScrollPane1.setVisible(true);

jScrollPane2.setVisible(false);

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

int FineDue = 0;

int fine;

String query = null;

int issuerId = 0;

int n = 0;

String username = t2.getText();

try

{

issuerId = Integer.parseInt(t1.getText());

}

catch (NumberFormatException e)

{

n = 1;

}

if (n == 1)

{

if (username.isEmpty())

{

JOptionPane.showMessageDialog(this, "Please enter IssuerID or IssuerName");

}

else

{

query = "select username,issuerid from users where username='" + username + "';";

}

}

else

{

query = "select username,issuerid from users where issuerId='" + issuerId + "';";

}

if (query != null)

{

try {

Statement stmt = myQueries.getStmt();

ResultSet rs = stmt.executeQuery(query);

rs.next();

if(!(username.equals("all")))

{

username = rs.getString("username");

query = "select fine from settings;";

ResultSet rs0 = stmt.executeQuery(query);

rs0.next();

fine = rs0.getInt("fine");

query = "SELECT \* FROM accounts where username='" + username + "';";

ResultSet rs1 = stmt.executeQuery(query);

model.setRowCount(0);

while (rs1.next())

{

int Bookid = rs1.getInt("Bookcode");

String Bookname = rs1.getString("Bookname");

Date iDate = rs1.getDate("issueDate");

Date rDate = rs1.getDate("ReturnDate");

Calendar issueDate = Calendar.getInstance();

issueDate.setTime(iDate);

Calendar returnDate = Calendar.getInstance();

returnDate.setTime(rDate);

Calendar curdate = Calendar.getInstance();

if (curdate.after(returnDate))

{

while (curdate.after(returnDate))

{

returnDate.add(Calendar.DAY\_OF\_MONTH, 1);

FineDue++;

}

returnDate.add(Calendar.DAY\_OF\_MONTH, -FineDue);

FineDue = (FineDue - 1) \* fine;

}

int iyear = issueDate.get(Calendar.YEAR);

int imonth = issueDate.get(Calendar.MONTH) + 1;

int idayOfMonth = issueDate.get(Calendar.DAY\_OF\_MONTH);

String idate = "" + iyear + "/" + imonth + "/" + idayOfMonth;

int ryear = returnDate.get(Calendar.YEAR);

int rmonth = returnDate.get(Calendar.MONTH) + 1;

int rdayOfMonth = returnDate.get(Calendar.DAY\_OF\_MONTH);

String rdate = "" + ryear + "/" + rmonth + "/" + rdayOfMonth;

model.addRow(new Object[]{Bookid, Bookname, idate, rdate, FineDue});

FineDue = 0;

}

}

else

{

query = "select fine from settings;";

ResultSet rs0 = stmt.executeQuery(query);

rs0.next();

fine = rs0.getInt("fine");

query = "SELECT \* FROM accounts order by username;";

ResultSet rs1 = stmt.executeQuery(query);

model.setRowCount(0);

while (rs1.next())

{

int Bookid = rs1.getInt("Bookcode");

username = rs1.getString("username");

Date iDate = rs1.getDate("issueDate");

Date rDate = rs1.getDate("ReturnDate");

Calendar issueDate = Calendar.getInstance();

issueDate.setTime(iDate);

Calendar returnDate = Calendar.getInstance();

returnDate.setTime(rDate);

Calendar curdate = Calendar.getInstance();

if (curdate.after(returnDate))

{

while (curdate.after(returnDate))

{

returnDate.add(Calendar.DAY\_OF\_MONTH, 1);

FineDue++;

}

returnDate.add(Calendar.DAY\_OF\_MONTH, -FineDue);

FineDue = (FineDue - 1) \* fine;

}

int iyear = issueDate.get(Calendar.YEAR);

int imonth = issueDate.get(Calendar.MONTH) + 1;

int idayOfMonth = issueDate.get(Calendar.DAY\_OF\_MONTH);

String idate = "" + iyear + "/" + imonth + "/" + idayOfMonth;

int ryear = returnDate.get(Calendar.YEAR);

int rmonth = returnDate.get(Calendar.MONTH) + 1;

int rdayOfMonth = returnDate.get(Calendar.DAY\_OF\_MONTH);

String rdate = "" + ryear + "/" + rmonth + "/" + rdayOfMonth;

model.addRow(new Object[]{Bookid, username, idate, rdate, FineDue});

FineDue = 0;

}

}

} catch (Exception e) {

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

**For “View all books”**

jScrollPane1.setVisible(false);

jScrollPane2.setVisible(true);

DefaultTableModel model1 = (DefaultTableModel) jTable2.getModel();

try {

Statement stmt = myQueries.getStmt();

String query = "SELECT \* FROM books;";

ResultSet rs = stmt.executeQuery(query);

model1.setRowCount(0);

while (rs.next())

{

int Bookid = rs.getInt("Bookcode");

String Bookname = rs.getString("Bookname");

String author = rs.getString("author");

String genre = rs.getString("genre");

model1.addRow(new Object[]{Bookid, Bookname, author, genre});

}

}

catch (Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}

**For ”issue Book”**

DefaultTableModel model1 = (DefaultTableModel) jTable2.getModel();

int count;

int bookcode = 0;

int n = 0;

String bookname = null;

String query;

int issuerid = 0;

int m = 0;

String username = t2.getText();

String query1 = null;

int r = 0;

int row = jTable2.getSelectedRow();

if (row == -1)

{

JOptionPane.showMessageDialog(this, "Please Select a Book");

}

else

{

bookcode = (Integer) model1.getValueAt(row, 0);

bookname = (String) model1.getValueAt(row, 1);

}

try

{

issuerid = Integer.parseInt(t1.getText());

}

catch (NumberFormatException e)

{

m = 1;

}

if (!(m == 1))

{

query1 = "select issuerid , username from users where issuerid=" + issuerid + ";";

}

else if (!(username.isEmpty()))

{

query1 = "select issuerid , username from users where username='" + username + "';";

}

else if (!(n == 1 && username.isEmpty()))

{

JOptionPane.showMessageDialog(this, "Please Enter any one of the user identifiers");

}

else

{

JOptionPane.showMessageDialog(this, "Please Enter the User details");

}

if (row != -1 && query1 != null)

{

try {

Statement stmt = myQueries.getStmt();

ResultSet rs1 = stmt.executeQuery(query1);

rs1.next();

issuerid = rs1.getInt("issuerid");

username = rs1.getString("username");

query = "select count(\*) from accounts where username = '" + username + "';";

ResultSet rs2 = stmt.executeQuery(query);

rs2.next();

count = rs2.getInt("count(\*)");

query = "select \* from settings;";

ResultSet rs3 = stmt.executeQuery(query);

rs3.next();

int issuetime = rs3.getInt("issuetime");

int maxbooks = rs3.getInt("maxbooks");

if (count >= maxbooks)

{

r = JOptionPane.showConfirmDialog(this, "This user has already issued " + maxbooks + " books do you still want to continue?", "Warning!", JOptionPane.YES\_NO\_OPTION);

}

if (r == 0)

{

Calendar curdate;

curdate = Calendar.getInstance();

Calendar returndate;

curdate.add(Calendar.DAY\_OF\_MONTH, issuetime);

returndate = curdate;

curdate = Calendar.getInstance();

int iyear = curdate.get(Calendar.YEAR);

int imonth = curdate.get(Calendar.MONTH) + 1;

int idayOfMonth = curdate.get(Calendar.DAY\_OF\_MONTH);

int ryear = returndate.get(Calendar.YEAR);

int rmonth = returndate.get(Calendar.MONTH) + 1;

int rdayOfMonth = returndate.get(Calendar.DAY\_OF\_MONTH);

String idate = "" + iyear + "/" + imonth + "/" + idayOfMonth;

String rdate = "" + ryear + "/" + rmonth + "/" + rdayOfMonth;

query = "insert into accounts values(" + issuerid + ",'" + username + "'," + bookcode + ",'" + bookname + "','" + idate + "','" + rdate + "');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(this, "Book Succesfully issued");

}

}

catch (Exception f)

{

if (f.getMessage().equals("Duplicate entry '" + bookcode + "' for key 'PRIMARY'"))

{

JOptionPane.showMessageDialog(this, "This book has already been issued by you or someone else");

}

else

{

JOptionPane.showMessageDialog(this, f.getMessage());

}

}

}

**For ”Return book”**

jScrollPane1.setVisible(true);

jScrollPane2.setVisible(false);

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

int bookcode = 0;

int n = 0;

String bookname = null;

String query;

String query1 = null;

int row = jTable1.getSelectedRow();

int issuerid = 0;

int m = 0;

String username = t2.getText();

if (row == -1)

{

JOptionPane.showMessageDialog(this, "Please Select a Book");

}

else

{

bookcode = (Integer) model.getValueAt(row, 0);

bookname = (String) model.getValueAt(row, 1);

}

try

{

issuerid = Integer.parseInt(t1.getText());

}

catch (NumberFormatException e)

{

m = 1;

}

if (!(m == 1))

{

query1 = "select issuerid , username from users where issuerid=" + issuerid + ";";

}

else if (!(username.isEmpty()))

{

query1 = "select issuerid , username from users where username='" + username + "';";

}

else if (!(n == 1 && username.isEmpty()))

{

JOptionPane.showMessageDialog(this, "Please Enter any one of the user identifiers");

}

else

{

JOptionPane.showMessageDialog(this, "Please Enter the User details");

}

if (row != -1 && query1 != null)

{

try

{

Statement stmt = myQueries.getStmt();

ResultSet rs = stmt.executeQuery(query1);

rs.next();

issuerid = rs.getInt("issuerid");

username = rs.getString("username");

query = "delete from accounts where bookcode=" + bookcode + " and bookname='" + bookname + "'and issuerid=" + issuerid + " and username='" + username + "';";

stmt.executeUpdate(query);

jButton1.doClick();

JOptionPane.showMessageDialog(this, "Book Succesfully Returned");

}

catch (Exception f)

{

JOptionPane.showMessageDialog(this, f.getMessage());

}

}

**For ”show details”**

DefaultTableModel model1 = (DefaultTableModel) jTable2.getModel();

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

int row = jTable2.getSelectedRow();

int row1 = jTable1.getSelectedRow();

int bookcode;

if (row == -1)

{

if (row1 == -1)

{

JOptionPane.showMessageDialog(this, "Please Select a Book");

}

else

{

bookcode = (Integer) model.getValueAt(row1, 0);

bookDetails a = new bookDetails(bookcode);

a.setVisible(true);

row1=-1;

}

}

else

{

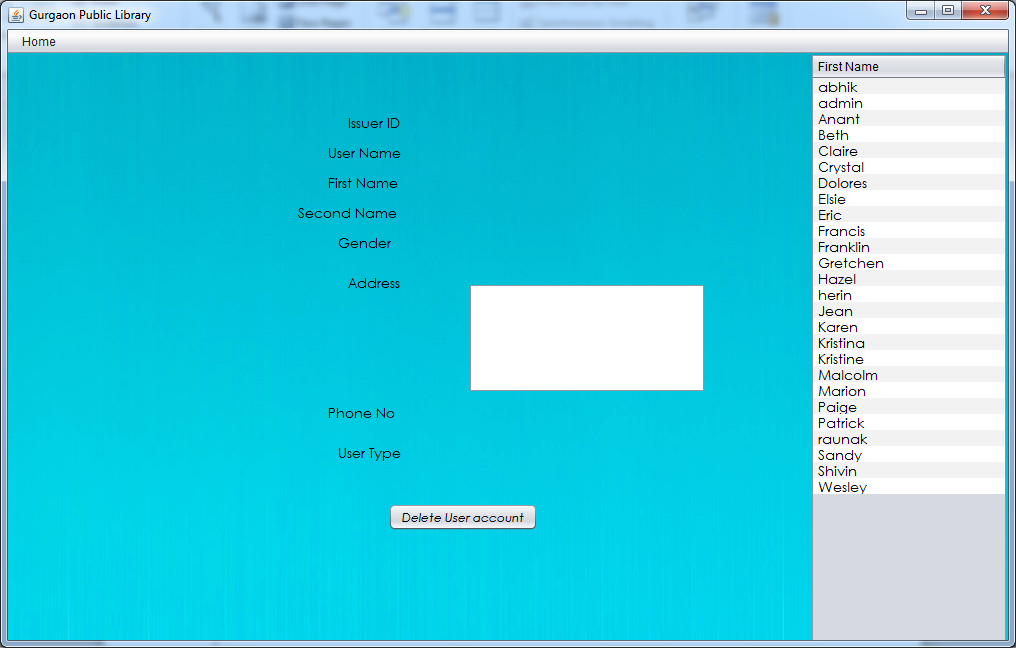
bookcode = (Integer) model1.getValueAt(row, 0);

bookDetails a = new bookDetails(bookcode);

a.setVisible(true);

row=-1;

}



public class ALLusers extends javax.swing.JFrame {

String username;

public ALLusers() {

initComponents();

}

public ALLusers(String username1) {

initComponents();

username=username1;

DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();

try

{

Class.forName("java.sql.DriverManager");

Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root", "DPS");

Statement stmt = (Statement) con.createStatement();

String query="SELECT issuerid,username,type,fname FROM users;";

ResultSet rs = stmt.executeQuery(query);

model1.setRowCount(0);

while(rs.next())

{

String fname = rs.getString("fname");

model1.addRow (new Object[] { fname});

}

}catch(Exception e)

{

}

}

private void jMenu1MouseClicked(java.awt.event.MouseEvent evt) {

MainPage a=new MainPage(username);

a.setVisible(true);

this.dispose();

}

private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();

int row=jTable1.getSelectedRow();

String fname= (String)model1.getValueAt(row, 0);

try

{

Class.forName("java.sql.DriverManager");

Connection con = (Connection)

DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root", "DPS");

Statement stmt = (Statement) con.createStatement();

String query="SELECT \* FROM users where fname ='"+fname+"';";

ResultSet rs = stmt.executeQuery(query);

rs.next();

int issuerid = rs.getInt("issuerid");

String username1 = rs.getString("username");

String lname = rs.getString("lname");

String add = rs.getString("address");

String gender = rs.getString("gender");

Long phoneno = rs.getLong("phoneno");

String type = rs.getString("type");

t1.setText(issuerid+"");

t2.setText(username1);

t3.setText(fname);

t4.setText(lname);

t6.setText(add);

t5.setText(gender);

t7.setText(phoneno+"");

t8.setText(type);

}catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

**For “Delete User Account”**

DefaultTableModel model1 = (DefaultTableModel) jTable1.getModel();

String username1=t2.getText();

int r = JOptionPane.showConfirmDialog(this,"Are you sure you want to Delete this user account?", "Alert!", JOptionPane.YES\_NO\_OPTION);

try

{

if(r==0)

{

Statement stmt = myQueries.getStmt();

String query = "Delete from users where username='"+username1+"'and type= 'issuer';";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(this, "Account succesfully deleted");

model1.removeRow(jTable1.getSelectedRow());

}

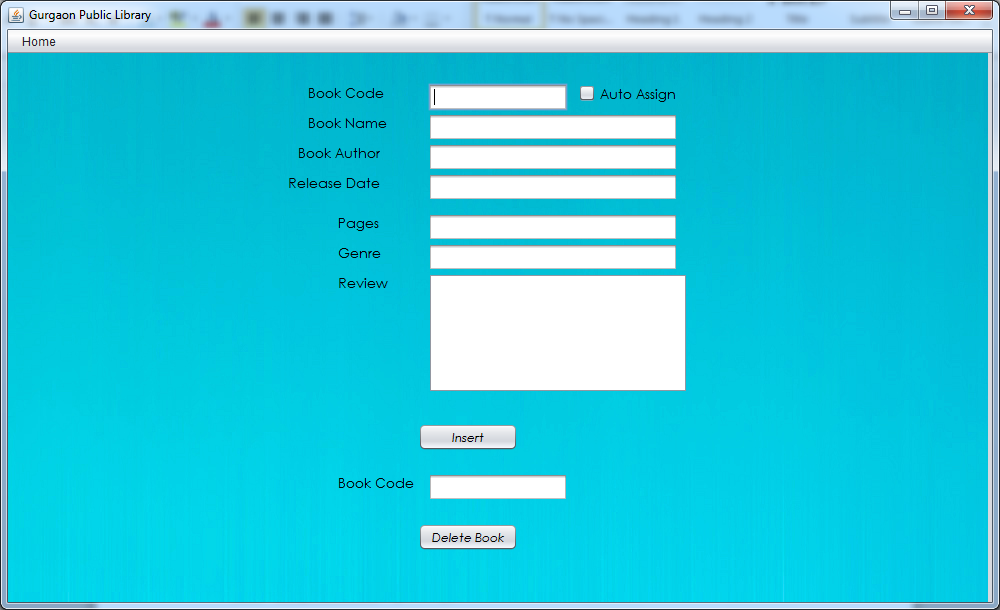
}

catch (Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}



private void jMenu3MouseClicked(java.awt.event.MouseEvent evt) {

MainPage a = new MainPage(username);

a.setVisible(true);

this.dispose();

}

**For “Insert”**

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

int code=0;

int n=0;

try

{

code = Integer.parseInt(t1.getText());

}

catch(NumberFormatException e)

{

n=1;

}

String name=t2.getText();

String author=t3.getText();

String genre=t4.getText();

String query=null;

if(n==1&&cb1.isSelected()==false)

{

JOptionPane.showMessageDialog (this, "Please Enter the Book Code");

}

else if(name.isEmpty())

{

JOptionPane.showMessageDialog (this, "Please Enter the Book Name");

}

else if(author.isEmpty())

{

JOptionPane.showMessageDialog (this, "Please Enter the author");

}

else if(genre.isEmpty())

{

JOptionPane.showMessageDialog (this, "Please Enter the genre");

}

else

{

try{

Class.forName("java.sql.DriverManager");

Connection con = (Connection) DriverManager.getConnection("jdbc:mysql://localhost:3306/project","root", "DPS");

Statement stmt = (Statement) con.createStatement();

if(cb1.isSelected())

{

query="select max(bookcode) from books;";

ResultSet rs=stmt.executeQuery(query);

rs.next();

code=rs.getInt("max(bookcode)")+1;

t1.setText(""+code);

}

query="INSERT INTO books VALUES('"+code+"','"+name+"','"+author+"','"+genre+"');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this, "Book Record Succesfully inserted");

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

}

private void t1KeyTyped(java.awt.event.KeyEvent evt) {

char kc = evt.getKeyChar();

if(!(kc>='0'&& kc<='9'))

{

evt.consume();

}

}

private void cb1ActionPerformed(java.awt.event.ActionEvent evt) {

t1.setEditable(false);

JOptionPane.showMessageDialog (this, "Warning! This may result in duplicate entry of the same book");

}

**For “Delete Book”**

int bookcode=0;

int n=0;

String query=null;

try

{

bookcode= Integer.parseInt(t1.getText());

}

catch(Exception e)

{

n=1;

}

if(n==1)

{

JOptionPane.showMessageDialog (this, "Please enter the book code!");

}

else

{

try

{

Statement stmt = myQueries.getStmt();

query="Delete from books where bookcode='"+bookcode+"';";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(this, "Book Record succesfully Deleted");

}

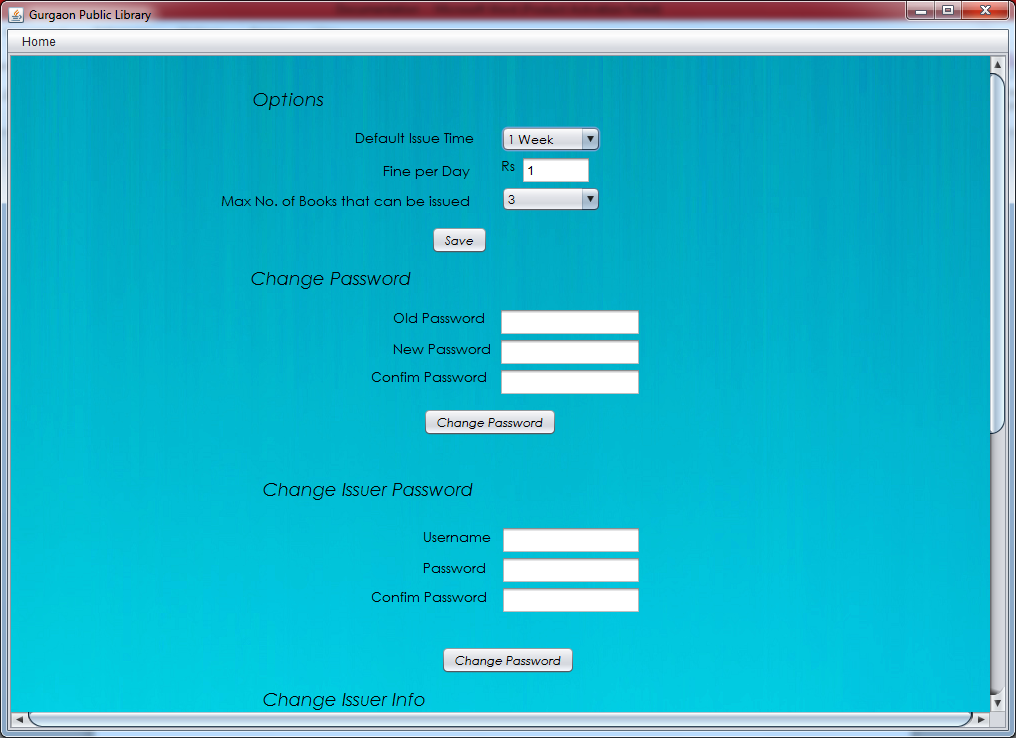
catch(Exception e)

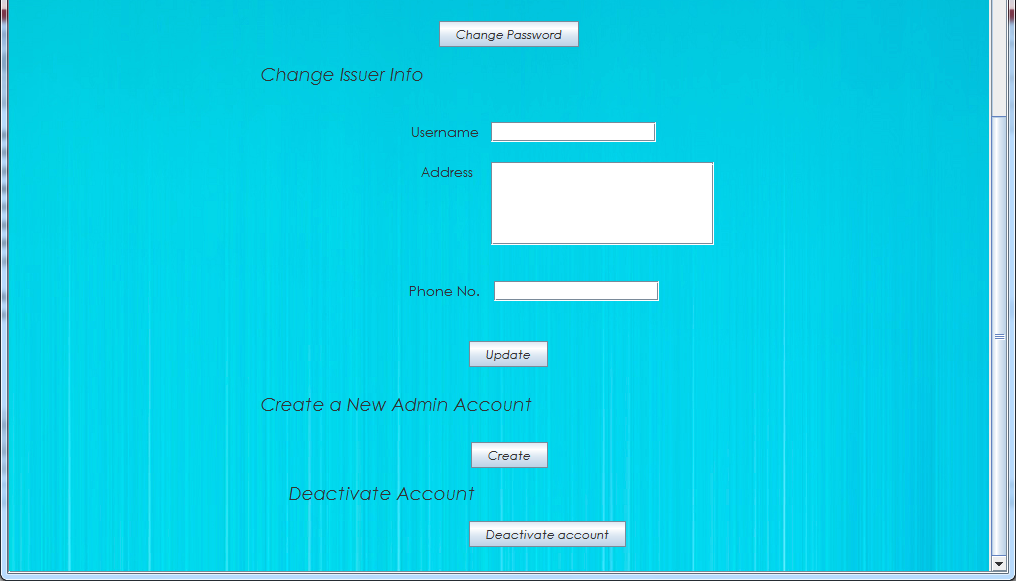
{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}





public class Options extends javax.swing.JFrame {

String username;

public Options() {

initComponents();

}

public Options(String username1) {

initComponents();

username = username1;

try

{

Statement stmt = myQueries.getStmt();

String query = "select \* from settings;";

ResultSet rs = stmt.executeQuery(query);

rs.next();

int issuetime = rs.getInt("issuetime");

int fine = rs.getInt("fine");

int maxbooks = rs.getInt("maxbooks");

t2.setText("" + fine);

cb2.setSelectedIndex(maxbooks-1);

if (issuetime == 1)

{

cb1.setSelectedIndex(0);

}

else

{

cb1.setSelectedIndex(issuetime/7);

}

}

catch (Exception e)

{

JOptionPane.showMessageDialog(null, e.getMessage());

}

}

**For “Change password(admin)”**

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

String oldpass = new String(p1.getPassword());

oldpass= md5hash.passwordsalted(oldpass);

String a= new String (p2.getPassword());

String b= new String (p3.getPassword());

String newpass;

String query;

try{

Statement stmt = myQueries.getStmt();

query="select password from users where username='"+username+"';";

ResultSet rs =stmt.executeQuery(query);

rs.next();

String BDoldpass=rs.getString("password");

if(BDoldpass.equals(oldpass))

{ if((a.equals(b)))

{

newpass=a;

newpass=md5hash.passwordsalted(newpass);

query="update users set password='"+newpass+"' where username='"+username+"'and password='"+oldpass+"'and type='issuer';";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Password Successfully Changed");

}

else

{

JOptionPane.showMessageDialog (this, "Passwords do not match");

}

}

else

{

JOptionPane.showMessageDialog (this, "Incorrect Password");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

**For “Save”**

int issuetime=0;

int fine=0;

int n=0;

int maxbooks= cb2.getSelectedIndex()+1;

try

{

fine= Integer.parseInt(t2.getText());

}

catch(NumberFormatException e)

{

n=1;

}

if(cb1.getSelectedIndex()==0)

{

issuetime=1;

}

else

{

issuetime=7\*cb1.getSelectedIndex();

}

if(n==1)

{

JOptionPane.showMessageDialog (this,"Please enter the Fine");

}

else{

try{

Statement stmt = myQueries.getStmt();

String query="update settings set fine='"+fine+"',issuetime='"+issuetime+"',maxbooks="+maxbooks+";";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Settings Changed");

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

}

**For “Change password(issuer)”**

String oldpass = new String(p1.getPassword());

oldpass= md5hash.passwordsalted(oldpass);

String a= new String (p2.getPassword());

String b= new String (p3.getPassword());

String newpass;

String query;

try{

Statement stmt = myQueries.getStmt();

query="select password from users where username='"+username+"';";

ResultSet rs =stmt.executeQuery(query);

rs.next();

String BDoldpass=rs.getString("password");

if(BDoldpass.equals(oldpass))

{ if((a.equals(b)))

{

newpass=a;

newpass=md5hash.passwordsalted(newpass);

query="update users set password='"+newpass+"' where username='"+username+"'and password='"+oldpass+"'and type='issuer';";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Password Successfully Changed");

}

else

{

JOptionPane.showMessageDialog (this, "Passwords do not match");

}

}

else

{

JOptionPane.showMessageDialog (this, "Incorrect Password");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

**For “Update”**

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

username=t3.getText();

String add=ta1.getText();

String phone=t6.getText();

try{

Statement stmt = myQueries.getStmt();

String query="update users set address='"+add+"',phone='"+phone+"' where username='"+username+"'and type='issuer');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Password Successfully Changed");

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

**For “Create”**

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

signup a= new signup();

a.setVisible(true);

}

**For “Deactivate Account”**

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

int r = JOptionPane.showConfirmDialog(this,"Are you sure you want to deactivate your account?", "Alert!", JOptionPane.YES\_NO\_OPTION);

try {

Statement stmt = myQueries.getStmt();

String query="select count(\*) from users where type = 'admin'";

ResultSet rs =stmt.executeQuery(query);

rs.next();

int count =rs.getInt("count(\*)");

if(count>1)

{

if(r==0)

{

query = "Delete from users where username='"+username+"'";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(this, "Account succesfully Deactivated");

}

}else

{

JOptionPane.showMessageDialog(this, "Sorry! There has to be one Admin account");

}

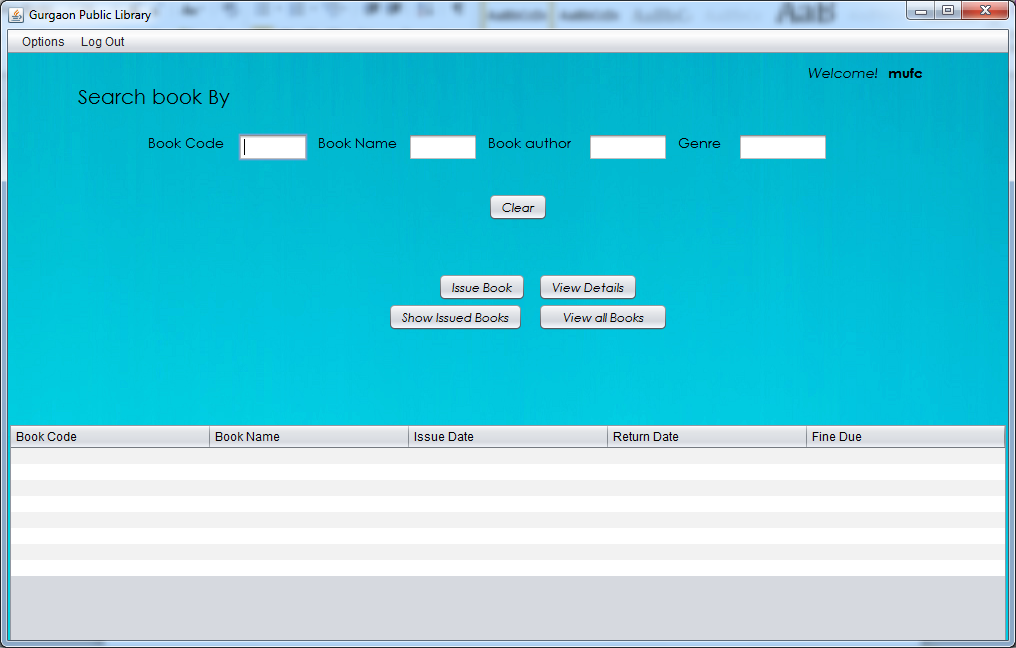
}

catch (Exception e)

{

JOptionPane.showMessageDialog(this, e.getMessage());

}}



**For “Show Issued Books”**

{

jScrollPane1.setVisible(true);

jScrollPane2.setVisible(false);

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

int FineDue = 0;

int fine;

try {

Statement stmt = myQueries.getStmt();

String query = "select fine from settings;";

ResultSet rs0 = stmt.executeQuery(query);

rs0.next();

fine = rs0.getInt("fine");

query = "SELECT \* FROM accounts where username='" + username1 + "';";

ResultSet rs = stmt.executeQuery(query);

model.setRowCount(0);

while (rs.next()) {

int Bookid = rs.getInt("Bookcode");

String Bookname = rs.getString("Bookname");

Date iDate = rs.getDate("issueDate");

Date rDate = rs.getDate("ReturnDate");

Calendar issueDate = Calendar.getInstance();

issueDate.setTime(iDate);

Calendar returnDate = Calendar.getInstance();

returnDate.setTime(rDate);

Calendar curdate = Calendar.getInstance();

if (curdate.after(returnDate)) {

while (curdate.after(returnDate)) {

returnDate.add(Calendar.DAY\_OF\_MONTH, 1);

FineDue++;

}

returnDate.add(Calendar.DAY\_OF\_MONTH, -FineDue);

FineDue = (FineDue - 1) \* fine;

}

int iyear = issueDate.get(Calendar.YEAR);

int imonth = issueDate.get(Calendar.MONTH) + 1;

int idayOfMonth = issueDate.get(Calendar.DAY\_OF\_MONTH);

String idate = "" + iyear + "/" + imonth + "/" + idayOfMonth;

int ryear = returnDate.get(Calendar.YEAR);

int rmonth = returnDate.get(Calendar.MONTH) + 1;

int rdayOfMonth = returnDate.get(Calendar.DAY\_OF\_MONTH);

String rdate = "" + ryear + "/" + rmonth + "/" + rdayOfMonth;

model.addRow(new Object[]{Bookid, Bookname, idate, rdate, FineDue});

FineDue = 0;

}

} catch (Exception e) {

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

**For “Issue Books”**

{

DefaultTableModel model1 = (DefaultTableModel) jTable2.getModel();

int bookcode = 0;

int n = 0;

String bookname = null;

String query;

int issuerid;

int count;

int row = jTable2.getSelectedRow();

if (row == -1) {

JOptionPane.showMessageDialog(this, "Please Select a Book");

} else {

bookcode = (Integer) model1.getValueAt(row, 0);

bookname = (String) model1.getValueAt(row, 1);

}

if (row != -1) {

try {

Statement stmt = myQueries.getStmt();

query = "select username,issuerid from users where username = '" + username1 + "';";

ResultSet rs1 = stmt.executeQuery(query);

rs1.next();

username1 = rs1.getString("username");

issuerid = rs1.getInt("issuerid");

query = "select count(\*) from accounts where username = '" + username1 + "';";

ResultSet rs2 = stmt.executeQuery(query);

rs2.next();

count = rs2.getInt("count(\*)");

query = "select \* from settings;";

ResultSet rs3 = stmt.executeQuery(query);

rs3.next();

int issuetime = rs3.getInt("issuetime");

int maxbooks = rs3.getInt("maxbooks");

if (count >= maxbooks) {

JOptionPane.showMessageDialog(this, "Sorry! You cannot Issue more than " + maxbooks + " books");

} else {

Calendar curdate = Calendar.getInstance();

Calendar returndate;

curdate.add(Calendar.DAY\_OF\_MONTH, issuetime);

returndate = curdate;

curdate = Calendar.getInstance();

int iyear = curdate.get(Calendar.YEAR);

int imonth = curdate.get(Calendar.MONTH) + 1;

int idayOfMonth = curdate.get(Calendar.DAY\_OF\_MONTH);

int ryear = returndate.get(Calendar.YEAR);

int rmonth = returndate.get(Calendar.MONTH) + 1;

int rdayOfMonth = returndate.get(Calendar.DAY\_OF\_MONTH);

String idate = "" + iyear + "/" + imonth + "/" + idayOfMonth;

String rdate = "" + ryear + "/" + rmonth + "/" + rdayOfMonth;

query = "insert into accounts values(" + issuerid + ",'" + username1 + "'," + bookcode + ",'" + bookname + "','" + idate + "','" + rdate + "');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog(this, "Book Succesfully issued");

}

} catch (Exception f) {

if (f.getMessage().equals("Duplicate entry '" + bookcode + "' for key 'PRIMARY'")) {

JOptionPane.showMessageDialog(this, "This book has already been issued by you or someone else");

} else {

JOptionPane.showMessageDialog(this, f.getMessage());

}

}

}

}

**For “View all Books”**

{

jScrollPane1.setVisible(false);

jScrollPane2.setVisible(true);

DefaultTableModel model1 = (DefaultTableModel) jTable2.getModel();

try {

Statement stmt = myQueries.getStmt();

String query = "SELECT \* FROM books;";

ResultSet rs = stmt.executeQuery(query);

model1.setRowCount(0);

while (rs.next()) {

int Bookid = rs.getInt("Bookcode");

String Bookname = rs.getString("Bookname");

String author = rs.getString("author");

String genre = rs.getString("genre");

model1.addRow(new Object[]{Bookid, Bookname, author, genre});

}

} catch (Exception e) {

JOptionPane.showMessageDialog(this, e.getMessage());

}

}

For “Clear”

{

t5.setText("");

t6.setText("");

t7.setText("");

t8.setText("");

}

**For “View Details”**

{

DefaultTableModel model1 = (DefaultTableModel) jTable2.getModel();

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

int row = jTable2.getSelectedRow();

int row1 = jTable1.getSelectedRow();

int bookcode;

if (row == -1)

{

if (row1 == -1)

{

JOptionPane.showMessageDialog(this, "Please Select a Book");

}

else

{

bookcode = (Integer) model.getValueAt(row1, 0);

bookDetails a = new bookDetails(bookcode);

a.setVisible(true);

row1 =-1;

}

}

else

{

bookcode = (Integer) model1.getValueAt(row, 0);

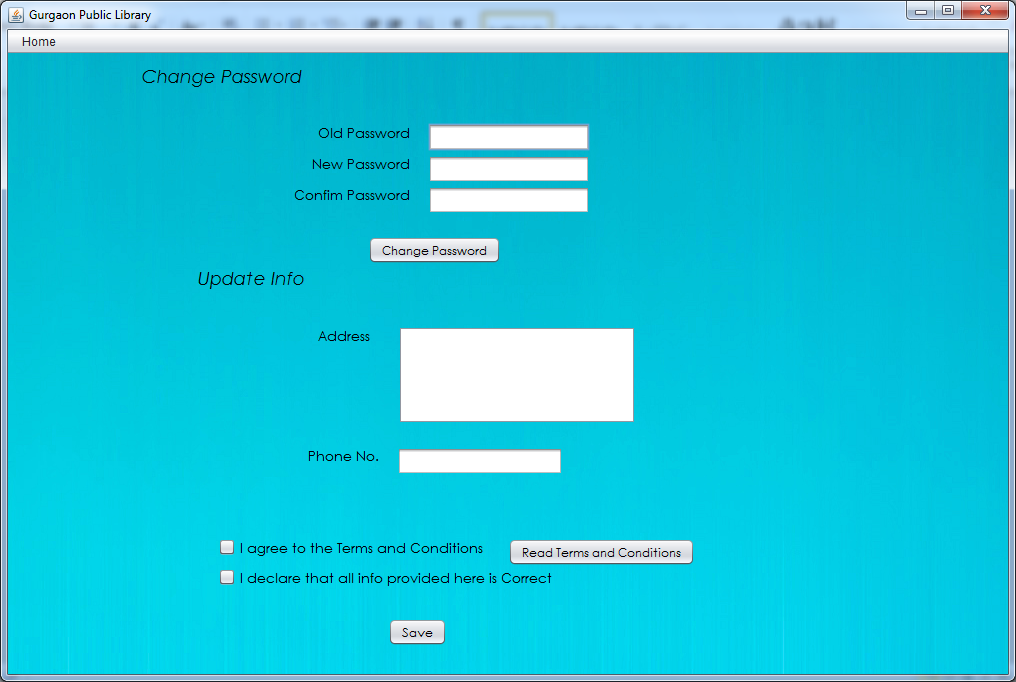
bookDetails a = new bookDetails(bookcode);

a.setVisible(true);

row= -1;

}

}



**For “Change Password”**

{

String oldpass = new String(p1.getPassword());

oldpass= md5hash.passwordsalted(oldpass);

String a= new String (p2.getPassword());

String b= new String (p3.getPassword());

String newpass;

String query;

try{

Statement stmt = myQueries.getStmt();

query="select password from users where username='"+username+"';";

ResultSet rs =stmt.executeQuery(query);

rs.next();

String BDoldpass=rs.getString("password");

if(!(oldpass.isEmpty()&&a.isEmpty()&&b.isEmpty()))

{

if(BDoldpass.equals(oldpass))

{ if((a.equals(b)))

{

newpass=a;

newpass=md5hash.passwordsalted(newpass);

query="update users set password='"+newpass+"' where username='"+username+"'and password='"+oldpass+"'and type='issuer';";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Password Successfully Changed");

}

else

{

JOptionPane.showMessageDialog (this, "Passwords do not match");

}

}

else

{

JOptionPane.showMessageDialog (this, "Incorrect Password");

}

}else

{

JOptionPane.showMessageDialog (this, "Please enter the details");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

**For ”Read Terms and Conditions”**

{

TnC a = new TnC();

a.setVisible(true);

}

**For “Save”**

{

String add=ta1.getText();

String phone=t3.getText();

if(username.isEmpty())

{JOptionPane.showMessageDialog (this, "Please enter your Username");}

else if(add.isEmpty())

{JOptionPane.showMessageDialog (this, "Please enter your address");}

else if(phone.isEmpty())

{JOptionPane.showMessageDialog (this, "Please enter your Phone Number");}

else if(cb1.isSelected()==false)

{JOptionPane.showMessageDialog (this, "Please Select Terms and Conditions");}

else if(cb2.isSelected()==false)

{JOptionPane.showMessageDialog (this, "Please acknowldge the information is correct");}

else{

try{

Statement stmt = myQueries.getStmt();

String query="update users set address='"+add+"' ,phoneno="+phone+" where username='"+username+"';";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Information added");

}

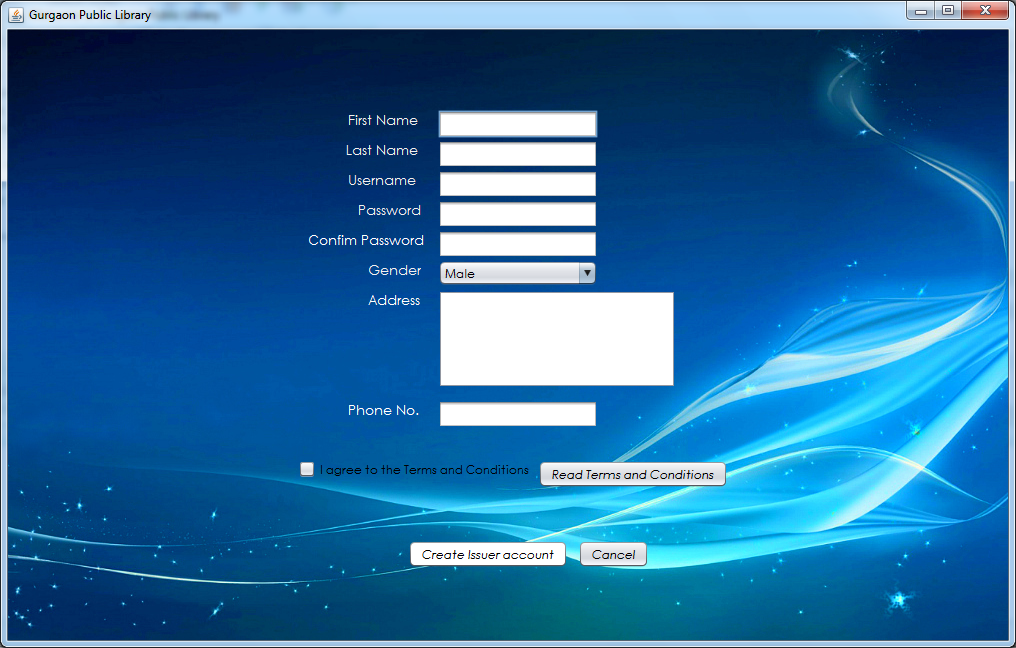
catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}}

}



**For “Create Issuer account”**

{

String name=t1.getText();

String secname=t2.getText();

String username=t3.getText();

String password= new String (p1.getPassword());

String gender= (String) comb1.getSelectedItem();

String add=ta1.getText();

String num=t5.getText();

String a= new String (p1.getPassword());

String b= new String (p2.getPassword());

String passwordmd5;

String type="issuer";

int issuerid=0;

try{

Statement stmt = myQueries.getStmt();

String query="select max(issuerId) from users;";

ResultSet rs=stmt.executeQuery(query);

rs.next();

issuerid=rs.getInt("max(issuerId)")+1;

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

if(name.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your Name");}

else if(secname.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your Second Name");}

else if(username.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your username");}

else if(add.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your address");}

else if(num.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter Phone no.");}

else if(a.isEmpty()||b.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter a Password");}

else if(cb1.isSelected()==false)

{JOptionPane.showMessageDialog(null,"Terms and Conditions not met");}

else

{

if(a.equals(b)&&!(a.isEmpty()||b.isEmpty()))

{

passwordmd5=md5hash.passwordsalted(password);

try{

Statement stmt = myQueries.getStmt();

String query="INSERT INTO users VALUES("+issuerid+",'"+name+"','"+secname+"','"+username+"','"+passwordmd5+"','"+gender+"','"+add+"','"+num+"','"+type+"');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Account Successfully created");

this.dispose();

}

catch(Exception e)

{

if(e.getMessage().equals("Duplicate entry '"+username+"' for key 'PRIMARY'"))

{

JOptionPane.showMessageDialog (this, "The username already exists");

}

else

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

}

else if(!(a.equals(b)))

{JOptionPane.showMessageDialog(null,"Passwords do not match");}

}

}

**For ”cancel”**

{

this.dispose();

}

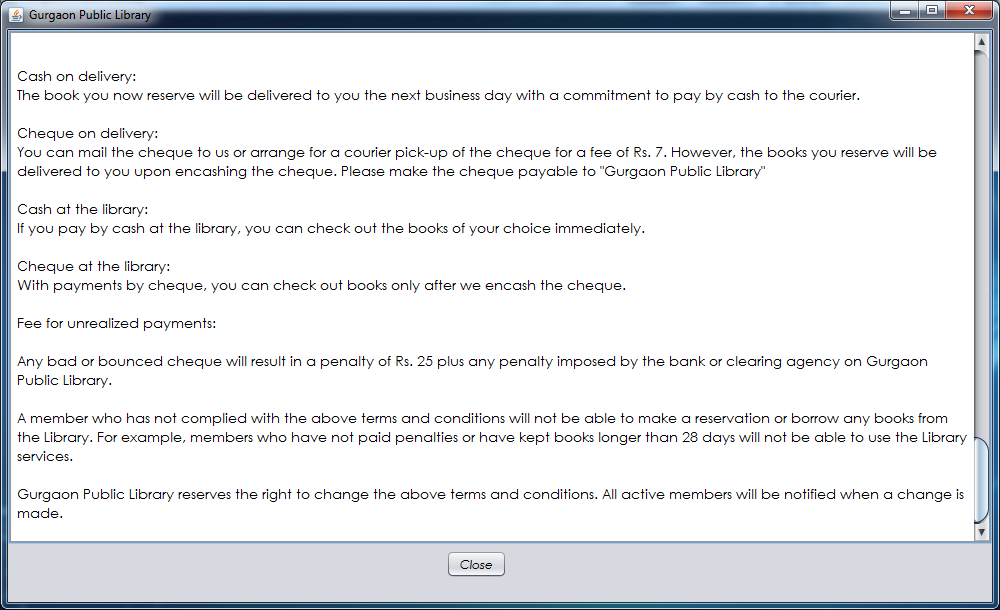
**For “Read terms and Conditions”**

{

TnC a= new TnC();

a.setVisible(true);

}

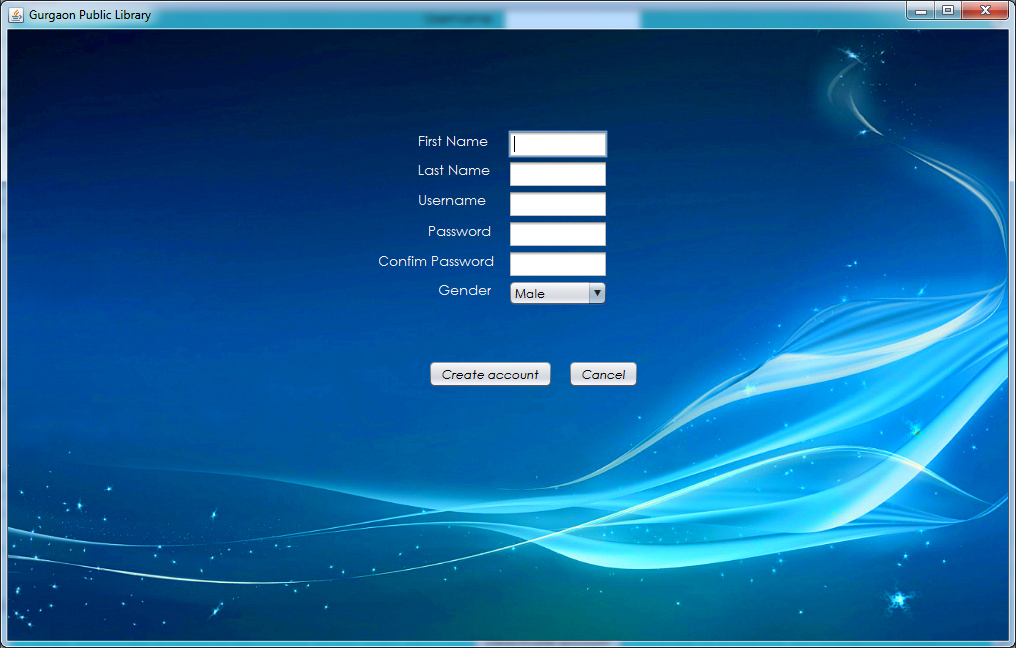


**For “Close”**

{

this.dispose();

}



**For “Create account”**

{

String name=t1.getText();

String secname=t2.getText();

String username=t3.getText();

String password= new String (p1.getPassword());

String gender= (String) combo1.getSelectedItem();

String a= new String (p1.getPassword());

String b= new String (p2.getPassword());

String passwordmd5;

String type="admin";

int issuerid=0;

try{

Statement stmt = myQueries.getStmt();

String query="select max(issuerId) from users;";

ResultSet rs=stmt.executeQuery(query);

rs.next();

issuerid=rs.getInt("max(issuerId)")+1;

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

if(name.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your Name");}

else if(secname.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your Second Name");}

else if(username.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter your username");}

else if(a.isEmpty()||b.isEmpty())

{JOptionPane.showMessageDialog(null,"Please enter a Password");}

else

{

if(a.equals(b)&&!(a.isEmpty()||b.isEmpty()))

{ passwordmd5=md5hash.passwordsalted(password);

try{

Statement stmt = myQueries.getStmt();

String query="INSERT INTO users VALUES("+issuerid+",'"+name+"','"+secname+"','"+username+"','"+passwordmd5+"','"+gender+"','Null','0000000000','"+type+"');";

stmt.executeUpdate(query);

JOptionPane.showMessageDialog (this,"Account Successfully created");

}

catch(Exception e)

{

JOptionPane.showMessageDialog (this, e.getMessage());

}

}

else if(!(a.equals(b)))

{JOptionPane.showMessageDialog(null,"Passwords do not match");}

}

}

**For “Cancel”**

{

this.dispose();

}

Validations

/\*

To allow only numbers and to restrict the number of digits till 10

\*/

private void t5KeyTyped**(**java**.**awt**.**event**.**KeyEvent evt**)** **{**

char kc **=** evt**.**getKeyChar**();**

**if** **(!(**kc **>=** '0' **&&** kc **<=** '9'**))** **{**

evt**.**consume**();**

**}**

String num **=** t5**.**getText**();**

**if** **(**num**.**length**()** **>=** 10**)** **{**

evt**.**consume**();**

**}**

**}**

/\*

To verify all info has been entered

\*/

**if(**name**.**isEmpty**())**

**{**JOptionPane**.**showMessageDialog**(null,**"Please enter your Name"**);}**

**else** **if(**secname**.**isEmpty**())**

**{**JOptionPane**.**showMessageDialog**(null,**"Please enter your Second Name"**);}**

**else** **if(**username**.**isEmpty**())**

**{**JOptionPane**.**showMessageDialog**(null,**"Please enter your username"**);}**

**else** **if(**add**.**isEmpty**())**

**{**JOptionPane**.**showMessageDialog**(null,**"Please enter your address"**);}**

**else** **if(**num**.**isEmpty**())**

**{**JOptionPane**.**showMessageDialog**(null,**"Please enter Phone no."**);}**

**else** **if(**a**.**isEmpty**()||**b**.**isEmpty**())**

**{**JOptionPane**.**showMessageDialog**(null,**"Please enter a Password"**);}**

**else** **if(**cb1**.**isSelected**()==false)**

**{**JOptionPane**.**showMessageDialog**(null,**"Terms and Conditions not met"**);}**

**else**

/\*

To verify the data entered is a number.

\*/

**try**

**{**

issuerid **=** Integer**.**parseInt**(**t1**.**getText**());**

**}**

**catch** **(**NumberFormatException e**)**

**{**

m **=** 1**;**

**}**

**Classes**

package Classes**;**

**import** java**.**math**.**BigInteger**;**

**import** java**.**security**.**MessageDigest**;**

/\*\*

\*

\* **@author** Shaleen

\*/

public class md5hash **{**

public static String password**(**String password**)**

**{**

String passwordmd5**=null;**

**try**

**{**

MessageDigest md5 **=**MessageDigest**.**getInstance**(**"MD5"**);**

md5**.**update**(**password**.**getBytes**(),**0**,**password**.**length**());**

passwordmd5**=new** BigInteger**(**1**,**md5**.**digest**()).**toString**(**16**);**

**}** **catch** **(**Exception e**)** **{**

e**.**getMessage**();**

**}**

**return** passwordmd5**;**

**}**

public static String passwordsalted**(**String password**)**

**{** String passwordmd5**=null;**

password**=** password**+**"cool"**;**

**try**

**{**

MessageDigest md5 **=**MessageDigest**.**getInstance**(**"MD5"**);**

md5**.**update**(**password**.**getBytes**(),**0**,**password**.**length**());**

passwordmd5**=new** BigInteger**(**1**,**md5**.**digest**()).**toString**(**16**);**

**}** **catch** **(**Exception e**)** **{**

e**.**getMessage**();**

**}**

**return** passwordmd5**;**

**}**

**}**

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package Classes**;**

**import** com**.**mysql**.**jdbc**.**Connection**;**

**import** com**.**mysql**.**jdbc**.**Statement**;**

**import** java**.**sql**.**DriverManager**;**

**import** javax**.**swing**.**JOptionPane**;**

/\*\*

\*

\* **@author** students

\*/

public class myQueries **{**

public static Statement getStmt**()**

**{**

Statement stmt**=null;**

**try{**

Class**.**forName**(**"java.sql.DriverManager"**);**

Connection con **=** **(**Connection**)** DriverManager**.**getConnection**(**"jdbc:mysql://localhost:3306/project"**,**"root"**,** "DPS"**);**

stmt **=** **(**Statement**)** con**.**createStatement**();**

**}**

**catch(**Exception e**)**

**{**

JOptionPane**.**showMessageDialog **(null,** e**.**getMessage**());**

**}**

**return** stmt**;**

**}**

**}**