**import** Login.\*;

**public** **class** run {

**public** **static** **void** main(String[] args) {

LoginModel log= **new** LoginModel();

LoginView screen = **new** LoginView();

screen.setVisible(**true**);

LoginController operator = **new** LoginController(log, screen);

operator.userSignIn();

}

}

package Controller;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.FileNotFoundException;

import java.io.IOException;

import javax.swing.\*;

import View.\*;

import Login.\*;

import Model.\*;

public class AdminController

{

AdminView adminview;

AdminModel adminmodel;

LoginView loginview;

DataReader reader = new DataReader();

//constructor

public AdminController(AdminView adv) { this.adminview = adv; }

//adding functions

public void activateButtons()

{

adminview.getmntmSignOut().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

adminview.dispose();

loginview = new LoginView();

loginview.setVisible(true);

LoginModel log= new LoginModel();

LoginController operator = new LoginController(log, loginview);

operator.userSignIn();

}

});

//buttons

adminview.getbtnAddResearcher().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.addResearcher();}

});

adminview.getbtnViewResearcher().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.displayResearchers();}

});

adminview.getbtnAddBookChapter().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.addBook();}

});

adminview.getbtnAddConferenceArticle().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.addConference();}

});

adminview.getbtnAddJournalArticle().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.addJournalArticle();}

});

adminview.getbtnViewBookChapter().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.displayBookChapter();}

});

adminview.getbtnViewJournalArticle().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.displayJournalArticles();}

});

adminview.getbtnViewConferenceArticle().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.displayConferenceArticle();}

});

adminview.getbtnViewIssuedArticles().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {adminview.displayIssuedPanel();}

});

}

//validate email

//reference from https://www.tutorialspoint.com/validate-email-address-in-java

public boolean isValidEmail(String e) {

String type = "^[\\w-\_\\.+]\*[\\w-\_\\.]\\@([\\w]+\\.)+[\\w]+[\\w]$";

return e.matches(type); }

//validate contact

//reference from https://howtodoinjava.com/regex/java-regex-validate-international-phone-numbers/

public boolean isvalidContact(String c)

{

String type = "^\\+(?:[0-9] ?){6,14}[0-9]$";

return c.matches(type);

}

//adding methods

public void addResearcher()

{ adminview.getbtnAddResearcher().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfTitle().getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "Enter complete information");}

else if(!isvalidContact(adminview.gettfContact().getText())) {

JOptionPane.showMessageDialog(null, "Enter valid contact (+977 98888888)");}

else if(!isValidEmail(adminview.gettfEmail().getText())) {

JOptionPane.showMessageDialog(null, "Enter Valid Email Address (example@email.com)");}

else {

adminmodel = new AdminModel();

adminmodel.enterNewReseacher(adminview);

adminview.addResearcher();

}

}

});

}

public void addBookChapter()

{

adminview.getbtnAddBook().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfTitle().getText().isEmpty() || adminview.gettfYear().getText().isEmpty())

JOptionPane.showMessageDialog(null, "Enter complete information");

else {

adminmodel = new AdminModel();

adminmodel.enterNewBookChapter(adminview);

adminview.addBook();}}

});

}

public void addJournalArticle()

{

adminview.getbtnAddJArticle().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfArticleTitle().getText().isEmpty() || adminview.gettfYear().getText().isEmpty())

JOptionPane.showMessageDialog(null, "Enter complete information");

else {

adminmodel = new AdminModel();

adminmodel.enterNewJournalArticle(adminview);

adminview.addJournalArticle();}

}

});

}

public void addConferenceArticle()

{

adminview.getbtnAddCArticle().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfTitle().getText().isEmpty() || adminview.gettfYear().getText().isEmpty())

JOptionPane.showMessageDialog(null, "Enter complete information");

else {

adminmodel = new AdminModel();

adminmodel.enterNewConferenceArticle(adminview);

adminview.addConference(); }

}

});

}

/\*\*

\* View Table

\*/

public void viewResearchers() {

adminmodel= new AdminModel();

JScrollPane rejsp = adminmodel.getrScrollPane();

adminview.setTable(adminmodel.getresearcherTable());

adminview.setScrollPane(rejsp); }

public void viewBookChapters() {

adminmodel= new AdminModel();

JScrollPane rejsp = adminmodel.getbScrollPane();

adminview.setBookTable(adminmodel.getBookTable());

adminview.setBookScrollPane(rejsp); }

public void viewJournalAricles() {

adminmodel= new AdminModel();

JScrollPane rejsp = adminmodel.getjScrollPane();

adminview.setJournalTable(adminmodel.getJournalTable());

adminview.setJournalScrollPane(rejsp); }

public void viewConferenceArticles() {

adminmodel= new AdminModel();

JScrollPane rejsp = adminmodel.getcScrollPane();

adminview.setConferenceTable(adminmodel.getConferenceTable());

adminview.setConferenceScrollPane(rejsp); }

public void viewHires() {

adminmodel= new AdminModel();

JScrollPane rejsp = adminmodel.gethScrollPane();

adminview.setHireTable(adminmodel.getHireTable());

adminview.setHireScrollPane(rejsp); }

/\*\*

\*\*\*\*\*\*\* modify\*\*\*\*\*\*\*\*

\*/

//researcher

public void editResearcher()

{

adminview.getbtnSave().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

adminmodel = new AdminModel();

if(adminview.gettfID().getText().isEmpty()) {showSelectRowtoEdit(); }else {

try {

adminmodel.editResearcherInfo(adminview,adminview.getEditIndex());

} catch (FileNotFoundException e1) {e1.printStackTrace();}

catch (IOException e1) {e1.printStackTrace(); }

JOptionPane.showMessageDialog(null, "Changes Saved");

adminview.displayResearchers();

}

}

});

}

//book chapters

public void editBookChapters()

{

adminview.getbtnSave().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

adminmodel = new AdminModel();

if(adminview.gettfID().getText().isEmpty()){showSelectRowtoEdit(); }else {

try {

adminmodel.editBookChapterInfo(adminview,adminview.getEditIndex());

} catch (FileNotFoundException e1) {e1.printStackTrace();}

catch (IOException e1) {e1.printStackTrace(); }

JOptionPane.showMessageDialog(null, "Changes Saved");

adminview.displayBookChapter();

}

}

});

}

//journal articles

public void editJournalArticles()

{

adminview.getbtnSave().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

adminmodel = new AdminModel();

if(adminview.gettfID().getText().isEmpty()){showSelectRowtoEdit();

}else {

try {

adminmodel.editJournalInfo(adminview,adminview.getEditIndex());

} catch (FileNotFoundException e1) {e1.printStackTrace();}

catch (IOException e1) {e1.printStackTrace(); }

JOptionPane.showMessageDialog(null, "Changes Saved");

adminview.displayJournalArticles();

}

}

});

}

//conference

public void editConfernceArticles()

{

adminview.getbtnSave().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

adminmodel = new AdminModel();

if(adminview.gettfID().getText().isEmpty()){

showSelectRowtoEdit();

}else {

try {

adminmodel.editConferenceArticle(adminview,adminview.getEditIndex());

} catch (FileNotFoundException e1) {e1.printStackTrace();}

catch (IOException e1) {e1.printStackTrace(); }

JOptionPane.showMessageDialog(null, "Changes Saved");

adminview.displayConferenceArticle();

}

}

});

}

/\*\*

\* \*\*\*\*\*delete\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

//researchers

public void deleteResearcher()

{

adminview.getbtnDelete().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfID().getText().isEmpty()){ showSelectRow(); }

else {

adminmodel = new AdminModel();

adminmodel.deleteResearcherInfo(adminview, adminview.getEditIndex());

adminview.displayResearchers();

}

}

});

}

//bookchapter

public void deleteBookChapters()

{

adminview.getbtnDelete().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfID().getText().isEmpty()) {

showSelectRow();

}else {

adminmodel = new AdminModel();

adminmodel.deleteBookChapterInfo(adminview, adminview.getEditIndex());

adminview.displayBookChapter();}

}

});

}

//journal

public void deleteJournalArticles()

{

adminview.getbtnDelete().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfID().getText().isEmpty()){

showSelectRow();

}else {

adminmodel = new AdminModel();

adminmodel.deleteJournalArticle(adminview, adminview.getEditIndex());

adminview.displayJournalArticles();

}}

});

}

//conference

public void deleteConferenceArticles()

{

adminview.getbtnDelete().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfID().getText().isEmpty()) {

showSelectRow();

}else {

adminmodel = new AdminModel();

adminmodel.deleteConferenceArticle(adminview, adminview.getEditIndex());

adminview.displayConferenceArticle();

}

}

});

}

//search

public void searchResearcher()

{

adminview.getbtnSearch().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

if(adminview.gettfSearch().getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "Enter ID to search!!");}

else {

if(adminmodel.searchResearcher(adminview.gettfSearch().getText())) {

adminview.searchResearcherPanel();

adminview.getlblId().setText(adminmodel.getstr1());

adminview.getlbl2().setText(adminmodel.getstr2());

adminview.getlbl3().setText(adminmodel.getstr3());

adminview.getlbl4().setText(adminmodel.getstr4());

adminview.getlbl5().setText(adminmodel.getstr5());

adminview.getlbl6().setText(adminmodel.getstr6());

adminview.getlbl7().setText(adminmodel.getstr7());

adminview.getlbl9().setText(adminmodel.getstr8());

}

else{showNotFound();}

}

}});

}

public void searchBook()

{

adminview.getbtnSearch().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

if(adminview.gettfSearch().getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "Enter Chapter Title to search!!");}

else {

if(adminmodel.searchBook(adminview.gettfSearch().getText())) {

adminview.bookDetailPanel();

adminview.getlblId().setText(adminmodel.getstr1());

adminview.getlbl2().setText(adminmodel.getstr2());

adminview.getlbl3().setText(adminmodel.getstr3());

adminview.getlbl4().setText(adminmodel.getstr4());

adminview.getlbl5().setText(adminmodel.getstr5());

adminview.getlbl6().setText(adminmodel.getstr6());

adminview.getlbl7().setText(adminmodel.getstr7());

adminview.getlbl8().setText(adminmodel.getstr8());

}else{showNotFound();}

}

}

});

}

public void searchJournalArticle()

{

adminview.getbtnSearch().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

if(adminview.gettfSearch().getText().isEmpty()) {

showEnterTitleMsg();}

else {

if(adminmodel.searchJournal(adminview.gettfSearch().getText())) {

adminview.searchJournalPanel();

adminview.getlblId().setText(adminmodel.getstr1());

adminview.getlbl2().setText(adminmodel.getstr2());

adminview.getlbl3().setText(adminmodel.getstr3());

adminview.getlbl4().setText(adminmodel.getstr4());

adminview.getlbl5().setText(adminmodel.getstr5());

adminview.getlbl6().setText(adminmodel.getstr6());

adminview.getlbl7().setText(adminmodel.getstr7());

adminview.getlbl8().setText(adminmodel.getstr8());

}else{showNotFound();}}

}

});

}

public void searchConferenceArticle()

{

adminview.getbtnSearch().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e)

{

if(adminview.gettfSearch().getText().isEmpty()) {

showEnterTitleMsg();}

else {

if(adminmodel.searchConference(adminview.gettfSearch().getText())) {

adminview.searchConferencePanel();

adminview.getlblId().setText(adminmodel.getstr1());

adminview.getlbl2().setText(adminmodel.getstr2());

adminview.getlbl3().setText(adminmodel.getstr3());

adminview.getlbl4().setText(adminmodel.getstr4());

adminview.getlbl5().setText(adminmodel.getstr5());

adminview.getlbl6().setText(adminmodel.getstr6());

adminview.getlbl7().setText(adminmodel.getstr7());

}else{showNotFound();}

}

}

});

}

public void issuebook() {

adminview.getbtnIssue().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

boolean issued = false;

if(adminview.gettfID().getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "Select a row first.");}

else {

if(reader.readHireDetails()!=null) {

for(int i=0; i<reader.readHireDetails().size();i++)

{

if(reader.readHireDetails().get(i).getAid().equals(adminview.gettfID().getText())) {

showIssueErrorMessage();

issued = true;

break;

}}}

if(issued==false) adminmodel.enterHireEntry(adminview);

}

}});

}

//return hires

public void returnHire()

{

adminview.getbtnReturn().addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(adminview.gettfID().getText().isEmpty()) {

JOptionPane.showMessageDialog(null, "Select a row to return.");

}else {

adminmodel = new AdminModel();

adminmodel.returnHire(adminview, adminview.getEditIndex());

adminview.displayIssuedPanel();

}

}

});

}

//error messages method

public void showIssueErrorMessage() {JOptionPane.showMessageDialog(null, "Article cannot be issued to more than one resesarcher.");}

public void showNotFound() { JOptionPane.showMessageDialog(null, "NOT FOUND"); };

public void showEnterTitleMsg() { JOptionPane.showMessageDialog(null, "Enter Article's Title to search!!");}

public void showSelectRow() { JOptionPane.showMessageDialog(null, "Select a row to delete."); }

public void showSelectRowtoEdit() { JOptionPane.showMessageDialog(null, "Select a row to edit."); }

}

**package** Controller;

**import** View.\*;

**import** java.awt.event.\*;

**import** javax.swing.JScrollPane;

**import** Login.\*;

**import** Model.\*;

**public** **class** ResearcherController

{

ResearcherView researcherview;

ResearcherModel researchermodel;

LoginView loginview;

String a ;

**public** **void** settrying(String a) {**this**.a = a;}

**public** String gettrying(){ **return** **this**.a; }

**public** ResearcherController(ResearcherView researcherview){**this**.researcherview =researcherview; }

**public** **void** operate()

{

researcherview.getmntmSignOut().addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e)

{

researcherview.dispose();

loginview = **new** LoginView();

loginview.setVisible(**true**);

LoginModel log= **new** LoginModel();

LoginController operator = **new** LoginController(log, loginview);

operator.userSignIn();

}

});

//show

researcherview.getbtnViewBookChapter().addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {researcherview.displayBookChapter();}

});

researcherview.getbtnViewJournalArticle().addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {researcherview.displayJournalArticles();}

});

researcherview.getbtnViewConferenceArticle().addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {researcherview.displayConferenceArticle();}

});

researcherview.getbtnViewHired().addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {researcherview.displayIssuedPanel();}

});

}

**public** **void** viewBookChapters() {

researchermodel= **new** ResearcherModel();

JScrollPane rejsp = researchermodel.getbScrollPane();

researcherview.setBookTable(researchermodel.getBookTable());

researcherview.setBookScrollPane(rejsp); }

**public** **void** viewJournalAricles() {

researchermodel= **new** ResearcherModel();

JScrollPane rejsp = researchermodel.getjScrollPane();

researcherview.setJournalTable(researchermodel.getJournalTable());

researcherview.setJournalScrollPane(rejsp); }

**public** **void** viewConferenceArticles() {

researchermodel= **new** ResearcherModel();

JScrollPane rejsp = researchermodel.getcScrollPane();

researcherview.setConferenceTable(researchermodel.getConferenceTable());

researcherview.setConferenceScrollPane(rejsp); }

**public** **void** viewHires() {

researchermodel= **new** ResearcherModel(gettrying());

JScrollPane rejsp = researchermodel.gethScrollPane();

researcherview.setHireTable(researchermodel.getHireTable());

researcherview.setHireScrollPane(rejsp); }

}

package Login;

import javax.swing.JOptionPane;

import Model.ResearcherModel;

import java.awt.event.\*;

import java.util.\*;

import View.\*;

public class LoginController{

LoginView loginView;

LoginModel loginModel;

public LoginController(LoginModel log, LoginView screen)

{

this.loginView= screen;

this.loginModel=log;

}

public void userSignIn()

{

loginView.signIN(new ActionListener () {

public void actionPerformed(ActionEvent e)

{

char[] c= {'a','d','m','i','n'};

if(loginView.getUserId().equals("admin")) {

if( Arrays.equals(loginView.getUserPass(),c))

{

loginView.dispose();

AdminView newView = new AdminView();

}

else {JOptionPane.showMessageDialog(null, "Password Incorrect!");}

}

else

{

if(loginModel.researcherVerified(loginView))

{

loginView.dispose();

ResearcherView newview = new ResearcherView(loginModel.getUserId());

}

else

{

JOptionPane.showMessageDialog(null, "Username or Password Incorrect!");

}

}

}

});

}

} package Login;

import java.io.\*;

import java.util.\*;

import Serializable.\*;

public class LoginModel {

private String userId, admin;

private char[] password;

ArrayList<Researchers> researchers = new ArrayList<Researchers>();

public boolean passwordComparison(String userId, char[] password) {

String uid= getUserId();

char[] pwd = getPassword();

if(uid.equals(userId)) {

if(Arrays.equals(pwd,password)) {

return true;

}

else

return false;

}

else

return false;

}

@SuppressWarnings("unchecked")

public boolean researcherVerified(LoginView entry)

{

boolean correct= false;

ArrayList<Researchers> researcher = null;

try {

FileInputStream filein = new FileInputStream("src/files/researchers.dat");

ObjectInputStream objectin = new ObjectInputStream(filein);

Object objdata= objectin.readObject();

researcher = (ArrayList<Researchers>)objdata;

for (int i = 0; i < researcher.size(); i++) {

setUserId(researcher.get(i).getUserName());

char[] pass =researcher.get(i).getPassword().toCharArray();

setPassword(pass);

if(correct=passwordComparison(entry.getUserId(),pass))

break;

}

objectin.close();

}catch (Exception e) {

}

return correct;

}

public void setUserId(String userId) {

this.userId= userId;

}

public void setPassword(char[] string) {

this.password = string;

}

public String getUserId() {

return this.userId;

}

public char[] getPassword() {

return this.password;

}

} package Login;

import java.awt.Color;

import java.awt.Font;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.\*;

public class LoginView extends JFrame {

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

private JPanel contentPane;

private JTextField tfUser;

private JPasswordField tfPassword;

JButton btnSignIn = new JButton("Sign In");

public LoginView() {

setTitle("Northampton Research Center - Desktop");

setIconImage(new ImageIcon(LoginView.class.getResource("../imgs/ICON.jpg")).getImage());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(580, 510);

setLocationRelativeTo(null);

setResizable(false);

contentPane = new JPanel();

setContentPane(contentPane);

contentPane.setLayout(null);

JPanel panel = new JPanel();

panel.setBounds(0, 0, 115, 510);

contentPane.add(panel);

panel.setLayout(null);

JLabel lblNewLabel\_1 = new JLabel(new ImageIcon(LoginView.class.getResource("../imgs/logincover.jpg")));

lblNewLabel\_1.setBounds(0, 0, 113, 478);

panel.add(lblNewLabel\_1);

JPanel panel\_1 = new JPanel();

panel\_1.setBackground(Color.WHITE);

panel\_1.setBounds(114, 0, 460, 510);

contentPane.add(panel\_1);

panel\_1.setLayout(null);

JLabel lblNewLabel\_2 = new JLabel("");

lblNewLabel\_2.setIcon(new ImageIcon(LoginView.class.getResource("../imgs/bg.jpg")));

lblNewLabel\_2.setBounds(0, 0, 450, 475);

panel\_1.add(lblNewLabel\_2);

JLabel lblNewLabel = new JLabel("Welcome");

lblNewLabel.setFont(new Font("Calibri Light", Font.PLAIN, 20));

lblNewLabel.setBounds(10, 10, 103, 41);

lblNewLabel\_2.add(lblNewLabel);

JLabel lblUserId = new JLabel("Username");

lblUserId.setFont(new Font("Arial", Font.PLAIN, 10));

lblUserId.setBounds(93, 209, 228, 25);

lblNewLabel\_2.add(lblUserId);

tfUser = new JTextField();

tfUser.setBounds(149, 212, 228, 20);

lblNewLabel\_2.add(tfUser);

tfUser.setColumns(10);

tfPassword = new JPasswordField();

tfPassword.setColumns(10);

tfPassword.setBounds(149, 238, 228, 20);

lblNewLabel\_2.add(tfPassword);

JLabel lblPassword = new JLabel("Password");

lblPassword.setFont(new Font("Arial", Font.PLAIN, 10));

lblPassword.setBounds(93, 236, 46, 25);

lblNewLabel\_2.add(lblPassword);

btnSignIn.addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

btnSignIn.setBackground(Color.WHITE);

}

public void mouseExited(MouseEvent e) {

btnSignIn.setBackground(Color.LIGHT\_GRAY);

}

});

btnSignIn.setForeground(Color.BLACK);

btnSignIn.setFont(new Font("Arial", Font.PLAIN, 10));

btnSignIn.setBackground(Color.LIGHT\_GRAY);

btnSignIn.setBounds(292, 300, 85, 21);

lblNewLabel\_2.add(btnSignIn);

}

//return userid and password from the textfields

public String getUserId() { return this.tfUser.getText(); }

public char[] getUserPass() { return this.tfPassword.getPassword(); }

//add button listener

public void signIN(ActionListener a) {btnSignIn.addActionListener(a);}

}

**package** Model;

**import** java.io.\*;

**import** java.util.ArrayList;

**import** javax.swing.\*;

**import** javax.swing.table.DefaultTableModel;

**import** Serializable.\*;

**import** View.\*;

**public** **class** AdminModel {

//table

**private** DefaultTableModel researchertableView = **new** DefaultTableModel(0, 9);

**private** DefaultTableModel bookTableView = **new** DefaultTableModel(0, 9);

**private** DefaultTableModel journalTableView = **new** DefaultTableModel(0, 9);

**private** DefaultTableModel conferenceTableView = **new** DefaultTableModel(0, 8);

**private** DefaultTableModel hireTableView = **new** DefaultTableModel(0, 3);

**private** JTable researcherTable=**new** JTable(researchertableView);

**private** JTable bookTable=**new** JTable(bookTableView);

**private** JTable journalTable= **new** JTable(journalTableView);

**private** JTable conferencetable= **new** JTable(conferenceTableView);

**private** JTable hiretable= **new** JTable(hireTableView);

**private** JScrollPane rScrollPane = **new** JScrollPane(researcherTable);

**private** JScrollPane bScrollPane = **new** JScrollPane(bookTable);

**private** JScrollPane jScrollPane = **new** JScrollPane(journalTable);

**private** JScrollPane cScrollPane = **new** JScrollPane(conferencetable);

**private** JScrollPane hScrollPane = **new** JScrollPane(hiretable);

**private** String str1,str2,str3,str4,str5,str6,str7,str8;

//array

**private** ArrayList <Researchers> reserchersArray = **new** ArrayList<Researchers>();

**private** ArrayList <BookChapters> bookChapterArray = **new** ArrayList<BookChapters>();

**private** ArrayList <JournalArticles> journalArray = **new** ArrayList<JournalArticles>();

**private** ArrayList <ConferenceArticles> conferenceArray = **new** ArrayList<ConferenceArticles>();

**private** ArrayList <IssueRecords> issueArray = **new** ArrayList<IssueRecords>();

Researchers researcherModel;

BookChapters bookModel;

JournalArticles journalModel;

ConferenceArticles conferenceModel;

IssueRecords issueModel;

DataReader reader = **new** DataReader();

//constructor

**public** AdminModel()

{

showResearcherTable();

showBookChapterTable();

showJournalArticles();

showConferenceArticles();

showHires();

}

/\*\*

\* \*\*\*\*\*creating new records

\*/

//researchers

**public** **void** enterNewReseacher(AdminView adminview)

{

researcherModel = **new** Researchers(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),

adminview.gettfAddress().getText(), adminview.gettfContact().getText(),

adminview.gettfEmail().getText(),

adminview.gettfUser().getText(), adminview.gettfPassword().getText());

**if**(reader.readData()!=**null**)

{

reserchersArray= reader.readData();

reserchersArray.add(researcherModel);

newData();

JOptionPane.*showMessageDialog*(**null**, "Researcher Added");

}

**else**

{

reserchersArray.add(researcherModel);

newData();

JOptionPane.*showMessageDialog*(**null**, "Researcher Added");

}

}

//bookchapters

**public** **void** enterNewBookChapter(AdminView adminview)

{

bookModel = **new** BookChapters(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),

adminview.gettfAuthor().getText(), adminview.gettfYear().getText(),

adminview.gettfEditor().getText(),

adminview.gettfPlace().getText(), adminview.gettfPublisher().getText(),adminview.gettfPage().getText());

**if**(reader.readBookData()!=**null**)

{

bookChapterArray= reader.readBookData();

bookChapterArray.add(bookModel);

newBookData();

JOptionPane.*showMessageDialog*(**null**, "Book Chapter Added");

}

**else**

{

bookChapterArray.add(bookModel);

newBookData();

JOptionPane.*showMessageDialog*(**null**, "Book Chapter Added");

}

}

//journal articles

**public** **void** enterNewJournalArticle(AdminView adminview)

{

journalModel = **new** JournalArticles(adminview.gettfID().getText(),

adminview.gettfArticleTitle().getText(),

adminview.gettfJTitle().getText(), adminview.gettfYear().getText(),

adminview.gettfVolume().getText(),

adminview.gettfIssue().getText(), adminview.gettfAuthor().getText(),adminview.gettfPage().getText());

**if**(reader.readJournalData()!=**null**)

{

journalArray= reader.readJournalData();

journalArray.add(journalModel);

newJournalData();

JOptionPane.*showMessageDialog*(**null**, "Journal Article Added");

}

**else**

{

journalArray.add(journalModel);

newJournalData();

JOptionPane.*showMessageDialog*(**null**, "Journal Article Added");

}

}

//conference

**public** **void** enterNewConferenceArticle(AdminView adminview)

{

conferenceModel = **new** ConferenceArticles(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),

adminview.gettfAuthor().getText(), adminview.gettfYear().getText(),

adminview.gettfEditor().getText(),

adminview.gettfPlace().getText(),adminview.gettfPage().getText());

**if**(reader.readConferenceData()!=**null**)

{

conferenceArray= reader.readConferenceData();

conferenceArray.add(conferenceModel);

newConferenceData();

JOptionPane.*showMessageDialog*(**null**, "Conference Article Added");

}

**else**

{

conferenceArray.add(conferenceModel);

newConferenceData();

JOptionPane.*showMessageDialog*(**null**, "Conference Article Added");

}

}

/\*\*

\* \*\*\*\*\*\*\*\*\*\*writing to file \*\*\*\*\*\*\*\*\*\*

\*/

**public** **void** newData()

{

**try**

{

File file = **new** File("src/files/researchers.dat");

FileOutputStream fileout = **new** FileOutputStream(file);

ObjectOutputStream objectout = **new** ObjectOutputStream(fileout);

objectout.writeObject(reserchersArray);

objectout.close();

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

}

//adding book chapter

**public** **void** newBookData()

{

**try**

{

File file = **new** File("src/files/bookchapters.dat");

FileOutputStream fileout = **new** FileOutputStream(file);

ObjectOutputStream objectout = **new** ObjectOutputStream(fileout);

objectout.writeObject(bookChapterArray);

objectout.close();

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

}

//journal article

**public** **void** newJournalData()

{

**try**

{

File file = **new** File("src/files/journalArticles.dat");

FileOutputStream fileout = **new** FileOutputStream(file);

ObjectOutputStream objectout = **new** ObjectOutputStream(fileout);

objectout.writeObject(journalArray);

objectout.close();

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

}

//conference

**public** **void** newConferenceData()

{

**try**

{

File file = **new** File("src/files/conferenceArticles.dat");

FileOutputStream fileout = **new** FileOutputStream(file);

ObjectOutputStream objectout = **new** ObjectOutputStream(fileout);

objectout.writeObject(conferenceArray);

objectout.close();

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

}

/\*\*

\* \*\*\*\*\*display methods \*\*\*\*\*\*\*\*\*\*

\*/

//researchers

**public** **void** showResearcherTable() {

**int** number=0;

**if**(reader.readData()!=**null**) {

reserchersArray =reader.readData();

Researchers reObj = **new** Researchers();

**for**(**int** i=0; i<reserchersArray.size(); i++)

{

number++;

reObj=reserchersArray.get(i);

Object[] data= {number,reObj.getId(),reObj.getName(),reObj.getAddress(),reObj.getContact(),reObj.getEmail(),reObj.getUserName(),reObj.getPassword()};

researchertableView.addRow(data);

}}

Object tableHead[]= {"SN","ID","Name","Address","Contact","Email","Username","Password"};

researchertableView.setColumnIdentifiers(tableHead);

researcherTable.setRowHeight(30);

//reference from: https://stackoverflow.com/questions/953972/java-jtable-setting-column-width

researcherTable.getColumnModel().getColumn(0).setPreferredWidth(3);

researcherTable.getColumnModel().getColumn(1).setPreferredWidth(7);

researcherTable.getColumnModel().getColumn(5).setPreferredWidth(100);

researcherTable.setAutoResizeMode(JTable.***AUTO\_RESIZE\_LAST\_COLUMN***);

}

//bookchapters

**public** **void** showBookChapterTable() {

**int** n =0;

**if**(reader.readBookData()!=**null**) {

bookChapterArray = reader.readBookData();

BookChapters reObj = **new** BookChapters();

**for**(**int** i=0; i<bookChapterArray.size(); i++)

{

n++;

reObj=bookChapterArray.get(i);

Object[] data= {n,reObj.getId(), reObj.getTitle(), reObj.getAuthor(), reObj.getYear(), reObj.getBookEditor(), reObj.getPublicationPlace(), reObj.getPublisher(), reObj.getPage()};

bookTableView.addRow(data);

}}

Object tableHead[]= {"SN","ID","Title","Author(s)","Publication Year","Book Editor","Place of Publication","Publisher","Page"};

bookTableView.setColumnIdentifiers(tableHead);

bookTable.setRowHeight(25);

bookTable.getColumnModel().getColumn(0).setPreferredWidth(3);

bookTable.getColumnModel().getColumn(1).setPreferredWidth(5);

bookTable.getColumnModel().getColumn(2).setPreferredWidth(100);

bookTable.setAutoResizeMode(JTable.***AUTO\_RESIZE\_LAST\_COLUMN***);

}

//journalArticles

**public** **void** showJournalArticles() {

**int** n=0;

**if**(reader.readJournalData()!=**null**) {

journalArray = reader.readJournalData();

JournalArticles reObj = **new** JournalArticles();

**for**(**int** i=0; i<journalArray.size(); i++)

{

n++;

reObj=journalArray.get(i);

Object[] data= {n,reObj.getId(), reObj.getTitle(), reObj.getJournalTitle(), reObj.getYear(),reObj.getVolume(), reObj.getIssue(),reObj.getAuthor(), reObj.getPage()};

journalTableView.addRow(data);

}}

Object tableHead[]= {"SN","ID","Title","Journal Title","Publication Year","Volume","Issue No.","Author(s)","Page"};

journalTableView.setColumnIdentifiers(tableHead);

journalTable.setRowHeight(25);

journalTable.getColumnModel().getColumn(0).setPreferredWidth(3);

journalTable.getColumnModel().getColumn(1).setPreferredWidth(5);

journalTable.getColumnModel().getColumn(2).setPreferredWidth(100);

journalTable.getColumnModel().getColumn(3).setPreferredWidth(100);

journalTable.setAutoResizeMode(JTable.***AUTO\_RESIZE\_LAST\_COLUMN***);

}

//conference Article

**public** **void** showConferenceArticles() {

**int** n =0;

**if**(reader.readConferenceData()!=**null**) {

conferenceArray = reader.readConferenceData();

ConferenceArticles reObj = **new** ConferenceArticles();

**for**(**int** i=0; i<conferenceArray.size(); i++)

{

n++;

reObj=conferenceArray.get(i);

Object[] data= {n,reObj.getId(), reObj.getTitle(), reObj.getAuthor(), reObj.getYear(), reObj.getBookEditor(), reObj.getPublicationPlace(), reObj.getPage()};

conferenceTableView.addRow(data);

}}

Object tableHead[]= {"SN","ID","Title","Author(s)","Publication Year","Editor(s)","Conference Location","Page"};

conferenceTableView.setColumnIdentifiers(tableHead);

conferencetable.setRowHeight(25);

conferencetable.getColumnModel().getColumn(0).setPreferredWidth(3);

conferencetable.getColumnModel().getColumn(1).setPreferredWidth(5);

conferencetable.getColumnModel().getColumn(2).setPreferredWidth(100);

conferencetable.setAutoResizeMode(JTable.***AUTO\_RESIZE\_LAST\_COLUMN***);

}

/\*\*

\*\*\*\*\*\*\*\*\*\*editing information\*\*\*\*\*

\*/

//researchers

**public** **void** editResearcherInfo(AdminView adminview, **int** i) **throws** FileNotFoundException, IOException {

reserchersArray= reader.readData();

researcherModel= **new** Researchers(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),

adminview.gettfAddress().getText(), adminview.gettfContact().getText(),

adminview.gettfEmail().getText(),

adminview.gettfUser().getText(), adminview.gettfPassword().getText());

reserchersArray.set(i,researcherModel);

newData();

}

//bookchapter

**public** **void** editBookChapterInfo(AdminView adminview, **int** i) **throws** FileNotFoundException, IOException {

bookChapterArray= reader.readBookData();

bookModel= **new** BookChapters(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),

adminview.gettfAuthor().getText(), adminview.gettfYear().getText(),

adminview.gettfEditor().getText(),

adminview.gettfPlace().getText(), adminview.gettfPublisher().getText(),adminview.gettfPage().getText());

bookChapterArray.set(i,bookModel);

newBookData();

}

//journal

**public** **void** editJournalInfo(AdminView adminview, **int** i) **throws** FileNotFoundException, IOException {

journalArray= reader.readJournalData();

journalModel= **new** JournalArticles(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),adminview.gettfJTitle().getText(),

adminview.gettfYear().getText(),

adminview.gettfVolume().getText(),

adminview.gettfIssue().getText(), adminview.gettfAuthor().getText(),adminview.gettfPage().getText());

journalArray.set(i,journalModel);

newJournalData();

}

//conference article

**public** **void** editConferenceArticle(AdminView adminview, **int** i) **throws** FileNotFoundException, IOException {

conferenceArray= reader.readConferenceData();

conferenceModel= **new** ConferenceArticles(adminview.gettfID().getText(),

adminview.gettfTitle().getText(),

adminview.gettfAuthor().getText(), adminview.gettfYear().getText(),

adminview.gettfEditor().getText(),

adminview.gettfPlace().getText(),adminview.gettfPage().getText());

conferenceArray.set(i,conferenceModel);

newConferenceData();

}

/\*\*

\*\*\*\*\*\*\*\*\*\*delete information\*\*\*\*\*

\*/

//researchers

**public** **void** deleteResearcherInfo(AdminView adminview,**int** i) {

**int** choice = JOptionPane.*showConfirmDialog*(**null**, "Are you sure to delete information?", "Confirm",

JOptionPane.***YES\_NO\_OPTION***, JOptionPane.***QUESTION\_MESSAGE***);

**if** (choice == JOptionPane.***YES\_OPTION***) {

reserchersArray= reader.readData();

reserchersArray.remove(i);

newData();

JOptionPane.*showMessageDialog*(**null**, "Removed");

}

}

//book chapters

**public** **void** deleteBookChapterInfo(AdminView adminview,**int** i) {

**int** choice = JOptionPane.*showConfirmDialog*(**null**, "Are you sure to delete information?", "Confirm",

JOptionPane.***YES\_NO\_OPTION***, JOptionPane.***QUESTION\_MESSAGE***);

**if** (choice == JOptionPane.***YES\_OPTION***) {

bookChapterArray= reader.readBookData();

bookChapterArray.remove(i);

newBookData();

JOptionPane.*showMessageDialog*(**null**, "Removed");

}

}

//journal

**public** **void** deleteJournalArticle(AdminView adminview,**int** i) {

**int** choice = JOptionPane.*showConfirmDialog*(**null**, "Are you sure to delete information?", "Confirm",

JOptionPane.***YES\_NO\_OPTION***, JOptionPane.***QUESTION\_MESSAGE***);

**if** (choice == JOptionPane.***YES\_OPTION***) {

journalArray= reader.readJournalData();

journalArray.remove(i);

newJournalData();

JOptionPane.*showMessageDialog*(**null**, "Removed");

}

}

//conference article

**public** **void** deleteConferenceArticle(AdminView adminview,**int** i) {

**int** choice = JOptionPane.*showConfirmDialog*(**null**, "Are you sure to delete information?", "Confirm",

JOptionPane.***YES\_NO\_OPTION***, JOptionPane.***QUESTION\_MESSAGE***);

**if** (choice == JOptionPane.***YES\_OPTION***) {

conferenceArray= reader.readConferenceData();

conferenceArray.remove(i);

newConferenceData();

JOptionPane.*showMessageDialog*(**null**, "Removed");

}

}

//accessors

**public** JScrollPane getrScrollPane() { **return** rScrollPane; }

**public** JScrollPane getbScrollPane() { **return** bScrollPane; }

**public** JScrollPane getjScrollPane() { **return** jScrollPane; }

**public** JScrollPane getcScrollPane() { **return** cScrollPane; }

**public** JScrollPane gethScrollPane() { **return** hScrollPane; }

**public** JTable getresearcherTable() { **return** researcherTable; }

**public** JTable getBookTable() { **return** bookTable; }

**public** JTable getJournalTable() { **return** journalTable; }

**public** JTable getConferenceTable() { **return** conferencetable; }

**public** JTable getHireTable() { **return** hiretable; }

**public** String getstr1() { **return** str1; }

**public** String getstr2() { **return** str2; }

**public** String getstr3() { **return** str3; }

**public** String getstr4() { **return** str4; }

**public** String getstr5() { **return** str5; }

**public** String getstr6() { **return** str6; }

**public** String getstr7() { **return** str7; }

**public** String getstr8() { **return** str8; }

//search

**public** **boolean** searchBook(String s)

{

bookChapterArray = reader.readBookData();

**boolean** a = **false**;

**for**(**int** i=0; i<bookChapterArray.size(); i++)

{

**if**(bookChapterArray.get(i).getTitle().equalsIgnoreCase(s))

{

str1=bookChapterArray.get(i).getId();

str2= bookChapterArray.get(i).getTitle();

str3= bookChapterArray.get(i).getAuthor();

str4 = bookChapterArray.get(i).getYear();

str5=bookChapterArray.get(i).getBookEditor();

str6=bookChapterArray.get(i).getPublicationPlace();

str7=bookChapterArray.get(i).getPublisher();

str8=bookChapterArray.get(i).getPage();

a= **true**;

**break**;

}

**else** {

a= **false**;

}

}

**return** a;

}

**public** **boolean** searchJournal(String s)

{

journalArray = reader.readJournalData();

**boolean** a = **false**;

**for**(**int** i=0; i<journalArray.size(); i++)

{

**if**(journalArray.get(i).getTitle().equalsIgnoreCase(s))

{

str1= journalArray.get(i).getId();

str2= journalArray.get(i).getTitle();

str3= journalArray.get(i).getJournalTitle();

str4 = journalArray.get(i).getYear();

str5=journalArray.get(i).getVolume();

str6=journalArray.get(i).getIssue();

str7=journalArray.get(i).getAuthor();

str8=journalArray.get(i).getPage();

a= **true**;

**break**;

}

**else** {a= **false**;}

}

**return** a;

}

**public** **boolean** searchConference(String s)

{

conferenceArray = reader.readConferenceData();

**boolean** a = **false**;

**for**(**int** i=0; i<conferenceArray.size(); i++)

{

**if**(conferenceArray.get(i).getTitle().equalsIgnoreCase(s))

{

str1=conferenceArray.get(i).getId();

str2= conferenceArray.get(i).getTitle();

str3= conferenceArray.get(i).getAuthor();

str4 = conferenceArray.get(i).getYear();

str5=conferenceArray.get(i).getBookEditor();

str6=conferenceArray.get(i).getPublicationPlace();

str7=conferenceArray.get(i).getPage();

a= **true**;

**break**;

}

**else** {a= **false**;}

}

**return** a;

}

**public** **boolean** searchResearcher(String s)

{

reserchersArray = reader.readData();

**boolean** a = **false**;

**for**(**int** i=0; i<reserchersArray.size(); i++)

{

**if**(reserchersArray.get(i).getId().equalsIgnoreCase(s))

{

str1=reserchersArray.get(i).getId();

str2= reserchersArray.get(i).getName();

str3= reserchersArray.get(i).getAddress();

str4 = reserchersArray.get(i).getContact();

str5=reserchersArray.get(i).getEmail();

str6=reserchersArray.get(i).getUserName();

str7=reserchersArray.get(i).getPassword();

str8="";

a= **true**;

//hire checks

**if**(reader.readHireDetails()!=**null**) {

**for**(**int** i1=0; i1<reader.readHireDetails().size();i1++) {

**if**(s.equals(reader.readHireDetails().get(i1).getRid()))

{

str8 = str8+", "+ reader.readHireDetails().get(i1).getAid();

}

}}

**break**;

}

**else** {a= **false**;}

}

**return** a;

}

/\*

\* \*\*\*\*hire articles

\*/

@SuppressWarnings("unchecked")

**public** **void** enterHireEntry(AdminView adminview)

{

issueModel = **new** IssueRecords(adminview.getReID().getSelectedItem().toString(), adminview.gettfID().getText(),adminview.gettfTitle().getText());

**if**(reader.readHireDetails()!=**null**){issueArray= reader.readHireDetails();}

issueArray.add(issueModel);

newIssue();

JOptionPane.*showMessageDialog*(**null**, "Article Issued");

}

**public** **void** newIssue()

{

**try**

{

File file = **new** File("src/files/issueRecords.dat");

FileOutputStream fileout = **new** FileOutputStream(file);

ObjectOutputStream objectout = **new** ObjectOutputStream(fileout);

objectout.writeObject(issueArray);

objectout.close();

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

}

//view hires

**public** **void** showHires() {

**if**(reader.readHireDetails()!=**null**) {

issueArray = reader.readHireDetails();

IssueRecords reObj = **new** IssueRecords();

**for**(**int** i=0; i<issueArray.size(); i++)

{

reObj=issueArray.get(i);

Object[] data= {reObj.getRid(), reObj.getAid(), reObj.getArticleName()};

hireTableView.addRow(data);

}

Object tableHead[]= {"Researcher ID","Article No.","Article Title"};

hireTableView.setColumnIdentifiers(tableHead);

}

}

//return

**public** **void** returnHire(AdminView adminview,**int** i) {

**int** opt = JOptionPane.*showConfirmDialog*(**null**, "Confirm Return ?", "Confirm",

JOptionPane.***YES\_NO\_OPTION***, JOptionPane.***QUESTION\_MESSAGE***);

**if** (opt == JOptionPane.***YES\_OPTION***) {

issueArray= reader.readHireDetails();

issueArray.remove(i);

newIssue();

JOptionPane.*showMessageDialog*(**null**, "Article Returned");

}

}

}

package View;

import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import Controller.\*;

import Login.\*;

import Model.\*;

public class AdminView extends JFrame {

private static final long serialVersionUID = 1L;

AdminController c= new AdminController(this);

DataReader mod= new DataReader();

//panel

JPanel bodyPanel, inbodyPanel;

//menus

private JMenu mnFile;

private JMenuItem mntmSignOut;

//buttons

private JButton btnAddResearcher, btnViewResearcher, btnAddBookChapter, btnAddJournalArticle, btnAddConferenceArticle, btnViewBookChapter, btnViewJournalArticle, btnViewConferenceArticle ,btnViewIssuedArticles;

//add data

private JButton btnAddJArticle, btnAddCArticle, btnAddBook ,btnSave, btnDelete;

//text fields

private JTextField tfID,tfPublisher,tfTitle,tfAddress,tfContact,tfEmail,tfUser,tfPassword, tfArticleTitle, tfJTitle, tfYear, tfVolume, tfIssue, tfAuthor, tfPage, tfEditor, tfPlace;

private JFormattedTextField tfEmailf;

//search fields

private JTextField tfSearch;

private JButton btnSearch, btnIssue, btnReturn;

//issue

private JComboBox researcherID;

//for table view

private JTable resTable, bookTable, journalTable, conferenceTable, hiresTable;

private JScrollPane reScrollPane, bookScrollPane, journalScrollPane, conferenceScrollPane, hireScrollPane;

private int editIndex;

//for search results

private JLabel lblID, lbl2, lbl3, lbl4,lbl5, lbl6, lbl7, lbl8;

private JTextArea lbl9;

//constructor

public AdminView()

{

setTitle("Northampton Research Center - Admin");

setIconImage(new ImageIcon(LoginView.class.getResource("../imgs/ICON.jpg")).getImage());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 1080, 700);

getContentPane().setBackground(Color.WHITE);

getContentPane().setLayout(null);

//body

bodyPanel = new JPanel();

bodyPanel.setBackground(Color.WHITE);

bodyPanel.setBounds(177, 20, 890, 645);

getContentPane().add(bodyPanel);

bodyPanel.setLayout(null);

inbodyPanel= new JPanel();

JMenuBar menuBar = new JMenuBar();

menuBar.setBackground(Color.LIGHT\_GRAY);

menuBar.setBounds(0, 0, 1066, 20);

getContentPane().add(menuBar);

mnFile = new JMenu("File");

menuBar.add(mnFile);

mntmSignOut = new JMenuItem("Sign Out");

mnFile.add(mntmSignOut);

JPanel menuPanel = new JPanel();

menuPanel.setBackground(new Color(216, 191, 216));

menuPanel.setBounds(0, 20, 180, 642);

getContentPane().add(menuPanel);

menuPanel.setLayout(null);

JLabel sidelabel = new JLabel(new ImageIcon(AdminView.class.getResource("../imgs/menu.jpg")));

sidelabel.setBounds(0, -15, 180, 669);

menuPanel.add(sidelabel);

btnAddResearcher = new JButton("Add Researcher");

sidelabel.add(btnAddResearcher);

btnAddResearcher.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnAddResearcher.setBorderPainted(false);

btnAddResearcher.setBackground(new Color(0, 191, 255));

btnAddResearcher.setBounds(22, 45, 131, 34);

btnViewResearcher = new JButton("View Researcher");

sidelabel.add(btnViewResearcher);

btnViewResearcher.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewResearcher.setBorderPainted(false);

btnViewResearcher.setBackground(new Color(0, 191, 255));

btnViewResearcher.setBounds(22, 89, 131, 34);

btnAddBookChapter = new JButton("Add Book Chapter");

sidelabel.add(btnAddBookChapter);

btnAddBookChapter.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnAddBookChapter.setBorderPainted(false);

btnAddBookChapter.setBackground(new Color(0, 191, 255));

btnAddBookChapter.setBounds(22, 133, 131, 34);

btnAddJournalArticle = new JButton("Add Journal Article");

sidelabel.add(btnAddJournalArticle);

btnAddJournalArticle.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnAddJournalArticle.setBorderPainted(false);

btnAddJournalArticle.setBackground(new Color(0, 191, 255));

btnAddJournalArticle.setBounds(22, 177, 131, 34);

btnAddConferenceArticle = new JButton("Add Conference Article");

sidelabel.add(btnAddConferenceArticle);

btnAddConferenceArticle.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnAddConferenceArticle.setBorderPainted(false);

btnAddConferenceArticle.setBackground(new Color(0, 191, 255));

btnAddConferenceArticle.setBounds(22, 221, 131, 34);

btnViewBookChapter = new JButton("View Book Chapter");

sidelabel.add(btnViewBookChapter);

btnViewBookChapter.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewBookChapter.setBorderPainted(false);

btnViewBookChapter.setBackground(new Color(0, 191, 255));

btnViewBookChapter.setBounds(22, 265, 131, 34);

btnViewJournalArticle = new JButton("View Journal Article");

sidelabel.add(btnViewJournalArticle);

btnViewJournalArticle.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewJournalArticle.setBorderPainted(false);

btnViewJournalArticle.setBackground(new Color(0, 191, 255));

btnViewJournalArticle.setBounds(22, 309, 131, 34);

btnViewConferenceArticle = new JButton("View Conference Article");

sidelabel.add(btnViewConferenceArticle);

btnViewConferenceArticle.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewConferenceArticle.setBorderPainted(false);

btnViewConferenceArticle.setBackground(new Color(0, 191, 255));

btnViewConferenceArticle.setBounds(22, 353, 131, 34);

btnViewIssuedArticles= new JButton("View Hired Articles");

sidelabel.add(btnViewIssuedArticles);

btnViewIssuedArticles.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewIssuedArticles.setBorderPainted(false);

btnViewIssuedArticles.setBackground(new Color(0, 191, 255));

btnViewIssuedArticles.setBounds(22, 397, 131, 34);

JLabel lblwelcom= new JLabel("Welcome Admin");

lblwelcom.setFont(new Font("Tahoma", Font.PLAIN, 50));

lblwelcom.setBounds(10,50,800,500);

bodyPanel.add(lblwelcom);

setVisible(true);

c.activateButtons();

}

//to add inbody panel to body

public void addPanel()

{

bodyPanel.removeAll();

bodyPanel.add(inbodyPanel);

bodyPanel.revalidate();

inbodyPanel.repaint();

bodyPanel.repaint();

}

//inbody panel property

public void inBodyProperty()

{

inbodyPanel.setBackground(Color.WHITE);

inbodyPanel.setBounds(10, 20, 890, 645);

inbodyPanel.setLayout(null);

inbodyPanel.removeAll();

}

//body contents

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Add Researcher

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void addResearcher()

{

inBodyProperty();

JLabel lblAddResearcher = new JLabel("Add New Researcher");

lblAddResearcher.setFont(new Font("Calibri Light", Font.PLAIN, 25));

lblAddResearcher.setBounds(352, 26, 224, 37);

inbodyPanel.add(lblAddResearcher);

JLabel lblID = new JLabel("ID No.");

lblID.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblID.setBounds(180, 96, 45, 15);

inbodyPanel.add(lblID);

tfID = new JTextField();

tfID.setBounds(304, 92, 364, 19);

tfID.setEditable(false);

inbodyPanel.add(tfID);

tfID.setColumns(10);

if(mod.readData()==null) {

tfID.setText("100001");

}

else {

int b= mod.readData().size();

int n = Integer.parseInt(mod.readData().get(b-1).getId());

String id = String.valueOf(n+1);

tfID.setText(id);

}

JLabel lblName = new JLabel("Name");

lblName.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblName.setBounds(180, 138, 45, 15);

inbodyPanel.add(lblName);

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(304, 134, 364, 19);

inbodyPanel.add(tfTitle);

JLabel lblAddress = new JLabel("Address");

lblAddress.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblAddress.setBounds(180, 181, 69, 15);

inbodyPanel.add(lblAddress);

tfAddress = new JTextField();

tfAddress.setColumns(10);

tfAddress.setBounds(304, 177, 364, 19);

inbodyPanel.add(tfAddress);

JLabel lblContact = new JLabel("Contact");

lblContact.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblContact.setBounds(180, 221, 82, 15);

inbodyPanel.add(lblContact);

tfContact = new JTextField();

tfContact.setColumns(10);

tfContact.setBounds(304, 217, 364, 19);

inbodyPanel.add(tfContact);

JLabel lblEmail = new JLabel("Email");

lblEmail.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblEmail.setBounds(180, 265, 45, 15);

inbodyPanel.add(lblEmail);

tfEmail = new JTextField();

tfEmail.setColumns(10);

tfEmail.setBounds(304, 261, 364, 19);

inbodyPanel.add(tfEmail);

JLabel lblUsername = new JLabel("Username");

lblUsername.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblUsername.setBounds(180, 351, 69, 15);

inbodyPanel.add(lblUsername);

tfUser = new JTextField();

tfUser.setColumns(10);

tfUser.setBounds(304, 347, 364, 19);

inbodyPanel.add(tfUser);

JLabel lblP = new JLabel("Password");

lblP.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblP.setBounds(180, 403, 82, 15);

inbodyPanel.add(lblP);

tfPassword = new JTextField();

tfPassword.setColumns(10);

tfPassword.setBounds(304, 399, 364, 19);

inbodyPanel.add(tfPassword);

JLabel lblcreateUsernameAnd = new JLabel("(Create username and password)");

lblcreateUsernameAnd.setForeground(new Color(0, 102, 204));

lblcreateUsernameAnd.setFont(new Font("Tahoma", Font.PLAIN, 13));

lblcreateUsernameAnd.setBounds(180, 317, 218, 13);

inbodyPanel.add(lblcreateUsernameAnd);

btnAddResearcher = new JButton("Add Researcher");

btnAddResearcher.setFont(new Font("Calibri Light", Font.PLAIN, 15));

btnAddResearcher.setBounds(387, 491, 142, 29);

inbodyPanel.add(btnAddResearcher);

addPanel();

c.addResearcher();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Add Book Chapter

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void addBook()

{

inBodyProperty();

JLabel lblBookchapter = new JLabel("Add a Book Chapter");

lblBookchapter.setFont(new Font("Calibri Light", Font.PLAIN, 25));

lblBookchapter.setBounds(352, 26, 224, 37);

inbodyPanel.add(lblBookchapter);

JLabel lblID = new JLabel("ID ");

lblID.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblID.setBounds(188, 96, 45, 15);

inbodyPanel.add(lblID);

btnAddBook = new JButton("Add Chapter");

btnAddBook.setFont(new Font("Calibri Light", Font.PLAIN, 15));

btnAddBook.setBounds(387, 491, 142, 29);

inbodyPanel.add(btnAddBook);

tfID = new JTextField();

tfID.setBounds(358, 92, 369, 19);

tfID.setEditable(false);

inbodyPanel.add(tfID);

tfID.setColumns(10);

if(mod.readBookData()==null) {

tfID.setText("200001");

}

else {

int b= mod.readBookData().size();

int n = Integer.parseInt(mod.readBookData().get(b-1).getId());

String id = String.valueOf(n+1);

tfID.setText(id);

}

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(358, 135, 369, 19);

inbodyPanel.add(tfTitle);

JLabel lblTitle = new JLabel("Title");

lblTitle.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblTitle.setBounds(188, 139, 45, 15);

inbodyPanel.add(lblTitle);

JLabel lblAuthors = new JLabel("Author(s)");

lblAuthors.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblAuthors.setBounds(188, 185, 111, 15);

inbodyPanel.add(lblAuthors);

tfAuthor = new JTextField();

tfAuthor.setColumns(10);

tfAuthor.setBounds(358, 181, 369, 19);

inbodyPanel.add(tfAuthor);

JLabel lblPublicationYear = new JLabel("Publication Year");

lblPublicationYear.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPublicationYear.setBounds(188, 229, 111, 15);

inbodyPanel.add(lblPublicationYear);

tfYear = new JTextField();

tfYear.setColumns(10);

tfYear.setBounds(358, 225, 369, 19);

inbodyPanel.add(tfYear);

JLabel lblBookEditor = new JLabel("Book Editor");

lblBookEditor.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblBookEditor.setBounds(188, 275, 111, 15);

inbodyPanel.add(lblBookEditor);

tfEditor = new JTextField();

tfEditor.setColumns(10);

tfEditor.setBounds(358, 271, 369, 19);

inbodyPanel.add(tfEditor);

JLabel lblPlaceOfPublication = new JLabel("Place of Publication");

lblPlaceOfPublication.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPlaceOfPublication.setBounds(188, 321, 119, 15);

inbodyPanel.add(lblPlaceOfPublication);

tfPlace = new JTextField();

tfPlace.setColumns(10);

tfPlace.setBounds(358, 317, 369, 19);

inbodyPanel.add(tfPlace);

tfPublisher = new JTextField();

tfPublisher.setColumns(10);

tfPublisher.setBounds(358, 366, 369, 19);

inbodyPanel.add(tfPublisher);

JLabel lblPublisher = new JLabel("Publisher");

lblPublisher.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPublisher.setBounds(188, 370, 119, 15);

inbodyPanel.add(lblPublisher);

tfPage = new JTextField();

tfPage.setColumns(10);

tfPage.setBounds(358, 415, 369, 19);

inbodyPanel.add(tfPage);

JLabel lblPageNo = new JLabel("Page No.");

lblPageNo.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPageNo.setBounds(188, 419, 119, 15);

inbodyPanel.add(lblPageNo);

addPanel();

c.addBookChapter();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Add Journal Article

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void addJournalArticle()

{

inBodyProperty();

JLabel lblAddResearcher = new JLabel("Add Journal Article");

lblAddResearcher.setFont(new Font("Calibri Light", Font.PLAIN, 25));

lblAddResearcher.setBounds(387, 23, 224, 37);

inbodyPanel.add(lblAddResearcher);

JLabel lblID = new JLabel("ID ");

lblID.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblID.setBounds(188, 96, 45, 15);

inbodyPanel.add(lblID);

btnAddJArticle = new JButton("Add Article");

btnAddJArticle.setFont(new Font("Calibri Light", Font.PLAIN, 15));

btnAddJArticle.setBounds(387, 491, 142, 29);

inbodyPanel.add(btnAddJArticle);

tfID = new JTextField();

tfID.setBounds(358, 92, 369, 19);

tfID.setEditable(false);

inbodyPanel.add(tfID);

tfID.setColumns(10);

if(mod.readJournalData()==null) {

tfID.setText("300001");

}

else {

int b= mod.readJournalData().size();

int n = Integer.parseInt(mod.readJournalData().get(b-1).getId());

String id = String.valueOf(n+1);

tfID.setText(id);

}

tfArticleTitle = new JTextField();

tfArticleTitle.setColumns(10);

tfArticleTitle.setBounds(358, 135, 369, 19);

inbodyPanel.add(tfArticleTitle);

JLabel lblTitle = new JLabel("Article Title");

lblTitle.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblTitle.setBounds(188, 139, 136, 15);

inbodyPanel.add(lblTitle);

JLabel lblAuthors = new JLabel("Author(s)");

lblAuthors.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblAuthors.setBounds(188, 370, 111, 15);

inbodyPanel.add(lblAuthors);

tfJTitle = new JTextField();

tfJTitle.setColumns(10);

tfJTitle.setBounds(358, 181, 369, 19);

inbodyPanel.add(tfJTitle);

JLabel lblPublicationYear = new JLabel("Publication Year");

lblPublicationYear.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPublicationYear.setBounds(188, 229, 111, 15);

inbodyPanel.add(lblPublicationYear);

tfYear = new JTextField();

tfYear.setColumns(10);

tfYear.setBounds(358, 225, 369, 19);

inbodyPanel.add(tfYear);

JLabel lblBookEditor = new JLabel("Volume");

lblBookEditor.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblBookEditor.setBounds(188, 275, 111, 15);

inbodyPanel.add(lblBookEditor);

tfVolume = new JTextField();

tfVolume.setColumns(10);

tfVolume.setBounds(358, 271, 369, 19);

inbodyPanel.add(tfVolume);

JLabel lblPlaceOfPublication = new JLabel("Issue no.");

lblPlaceOfPublication.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPlaceOfPublication.setBounds(188, 321, 119, 15);

inbodyPanel.add(lblPlaceOfPublication);

tfIssue = new JTextField();

tfIssue.setColumns(10);

tfIssue.setBounds(358, 317, 369, 19);

inbodyPanel.add(tfIssue);

tfAuthor = new JTextField();

tfAuthor.setColumns(10);

tfAuthor.setBounds(358, 366, 369, 19);

inbodyPanel.add(tfAuthor);

tfPage = new JTextField();

tfPage.setColumns(10);

tfPage.setBounds(358, 415, 369, 19);

inbodyPanel.add(tfPage);

JLabel lblPageNo = new JLabel("Page No.");

lblPageNo.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPageNo.setBounds(188, 419, 119, 15);

inbodyPanel.add(lblPageNo);

JLabel lblJournalTitle = new JLabel("Journal Title");

lblJournalTitle.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblJournalTitle.setBounds(188, 184, 136, 15);

inbodyPanel.add(lblJournalTitle);

addPanel();

c.addJournalArticle();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Add Conference Article

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void addConference()

{

inBodyProperty();

JLabel lblAddResearcher = new JLabel("Add Conference Article");

lblAddResearcher.setFont(new Font("Calibri Light", Font.PLAIN, 25));

lblAddResearcher.setBounds(318, 26, 311, 37);

inbodyPanel.add(lblAddResearcher);

JLabel lblID = new JLabel("ID ");

lblID.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblID.setBounds(188, 96, 45, 15);

inbodyPanel.add(lblID);

btnAddCArticle = new JButton("Add Article");

btnAddCArticle.setFont(new Font("Calibri Light", Font.PLAIN, 15));

btnAddCArticle.setBounds(403, 467, 142, 29);

inbodyPanel.add(btnAddCArticle);

tfID = new JTextField();

tfID.setBounds(358, 92, 369, 19);

tfID.setEditable(false);

inbodyPanel.add(tfID);

tfID.setColumns(10);

if(mod.readConferenceData()==null) {

tfID.setText("400001");

}

else {

int b= mod.readConferenceData().size();

int n = Integer.parseInt(mod.readConferenceData().get(b-1).getId());

String id = String.valueOf(n+1);

tfID.setText(id);

}

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(358, 135, 369, 19);

inbodyPanel.add(tfTitle);

JLabel lblTitle = new JLabel("Title");

lblTitle.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblTitle.setBounds(188, 139, 45, 15);

inbodyPanel.add(lblTitle);

JLabel lblAuthors = new JLabel("Author(s)");

lblAuthors.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblAuthors.setBounds(188, 185, 111, 15);

inbodyPanel.add(lblAuthors);

tfAuthor = new JTextField();

tfAuthor.setColumns(10);

tfAuthor.setBounds(358, 181, 369, 19);

inbodyPanel.add(tfAuthor);

JLabel lblPublicationYear = new JLabel("Publication Year");

lblPublicationYear.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPublicationYear.setBounds(188, 227, 111, 19);

inbodyPanel.add(lblPublicationYear);

tfYear = new JTextField();

tfYear.setColumns(10);

tfYear.setBounds(358, 225, 369, 19);

inbodyPanel.add(tfYear);

JLabel lblBookEditor = new JLabel("Editor(s)");

lblBookEditor.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblBookEditor.setBounds(188, 273, 111, 19);

inbodyPanel.add(lblBookEditor);

tfEditor = new JTextField();

tfEditor.setColumns(10);

tfEditor.setBounds(358, 271, 369, 19);

inbodyPanel.add(tfEditor);

JLabel lblPlaceOfPublication = new JLabel("Conference Location");

lblPlaceOfPublication.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPlaceOfPublication.setBounds(188, 319, 135, 19);

inbodyPanel.add(lblPlaceOfPublication);

tfPlace = new JTextField();

tfPlace.setColumns(10);

tfPlace.setBounds(358, 317, 369, 19);

inbodyPanel.add(tfPlace);

JLabel lblPageNo = new JLabel("Page No.");

lblPageNo.setFont(new Font("Calibri Light", Font.PLAIN, 15));

lblPageNo.setBounds(188, 368, 119, 15);

inbodyPanel.add(lblPageNo);

tfPage = new JTextField();

tfPage.setColumns(10);

tfPage.setBounds(358, 364, 369, 19);

inbodyPanel.add(tfPage);

addPanel();

c.addConferenceArticle();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* View Researchers

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void setScrollPane(JScrollPane jsp) { this.reScrollPane = jsp;}

public void setTable(JTable tab) { this.resTable = tab;}

public JTable getResearchTable(){ return resTable;}

public JScrollPane getResearchScrollPane(){ return reScrollPane;}

public void displayResearchers()

{

inBodyProperty();

//search box

tfSearch = new JTextField();

tfSearch.setToolTipText("Enter the ID");

tfSearch.setBounds(650, 0, 150, 20);

inbodyPanel.add(tfSearch);

btnSearch= new JButton();

btnSearch.setIcon(new ImageIcon(LoginView.class.getResource("../imgs/find1.png")));

btnSearch.setBounds(800, -5, 50, 30);

btnSearch.setBorderPainted(false);

btnSearch.setContentAreaFilled(false);

inbodyPanel.add(btnSearch);

JLabel l1 = new JLabel("Researchers");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

AdminController c= new AdminController(this);

c.viewResearchers();

inbodyPanel.add(getResearchScrollPane());

getResearchScrollPane().setBounds(10, 30, 850, 200);

//modify section

JLabel lblDetails = new JLabel("Details");

lblDetails.setForeground(new Color(0, 0, 139));

lblDetails.setFont(new Font("Corbel Light", Font.PLAIN, 20));

lblDetails.setBounds(413, 240, 62, 37);

inbodyPanel.add(lblDetails);

JLabel label = new JLabel("ID");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(10, 303, 45, 15);

inbodyPanel.add(label);

JLabel label\_1 = new JLabel("Name");

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_1.setBounds(10, 345, 45, 15);

inbodyPanel.add(label\_1);

JLabel label\_2 = new JLabel("Address");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(10, 388, 69, 15);

inbodyPanel.add(label\_2);

JLabel label\_3 = new JLabel("Contact");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(10, 428, 82, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Email");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(10, 472, 45, 15);

inbodyPanel.add(label\_4);

JLabel label\_5 = new JLabel("Username");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(10, 515, 69, 15);

inbodyPanel.add(label\_5);

JLabel label\_6 = new JLabel("Password");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(10, 555, 82, 15);

inbodyPanel.add(label\_6);

tfID = new JTextField();

tfID.setEditable(false);

tfID.setColumns(10);

tfID.setBounds(134, 299, 364, 19);

inbodyPanel.add(tfID);

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(134, 341, 364, 19);

inbodyPanel.add(tfTitle);

tfAddress = new JTextField();

tfAddress.setColumns(10);

tfAddress.setBounds(134, 384, 364, 19);

inbodyPanel.add(tfAddress);

tfContact = new JTextField();

tfContact.setColumns(10);

tfContact.setBounds(134, 424, 364, 19);

inbodyPanel.add(tfContact);

tfEmail = new JTextField();

tfEmail.setColumns(10);

tfEmail.setBounds(134, 468, 364, 19);

inbodyPanel.add(tfEmail);

tfUser = new JTextField();

tfUser.setColumns(10);

tfUser.setBounds(134, 511, 364, 19);

inbodyPanel.add(tfUser);

tfPassword = new JTextField();

tfPassword.setColumns(10);

tfPassword.setBounds(134, 555, 364, 19);

inbodyPanel.add(tfPassword);

btnSave = new JButton("Save");

btnSave.setBounds(301, 590, 85, 21);

inbodyPanel.add(btnSave);

btnDelete = new JButton("Delete");

btnDelete.setBounds(413, 590, 85, 21);

inbodyPanel.add(btnDelete);

addPanel();

showResearcherRowDetails();

c.editResearcher();

c.deleteResearcher();

c.searchResearcher();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* View Book Chapters

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void setBookScrollPane(JScrollPane jsp) { this.bookScrollPane = jsp;}

public void setBookTable(JTable tab) { this.bookTable = tab;}

public JTable getBookTable(){ return bookTable;}

public JScrollPane getBookScrollPane(){ return bookScrollPane;}

public void displayBookChapter()

{

inBodyProperty();

//search box

tfSearch = new JTextField();

tfSearch.setBounds(650, 0, 150, 20);

tfSearch.setToolTipText("Enter chapter title to search");

inbodyPanel.add(tfSearch);

btnSearch= new JButton();

btnSearch.setIcon(new ImageIcon(LoginView.class.getResource("../imgs/find1.png")));

btnSearch.setBounds(800, -5, 50, 30);

btnSearch.setBorderPainted(false);

btnSearch.setContentAreaFilled(false);

inbodyPanel.add(btnSearch);

JLabel l1 = new JLabel("Book Chapters");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

AdminController c= new AdminController(this);

c.viewBookChapters();

inbodyPanel.add(getBookScrollPane());

getBookScrollPane().setBounds(10, 30, 850, 200);

btnDelete = new JButton("Delete");

btnDelete.setBounds(413, 580, 85, 21);

inbodyPanel.add(btnDelete);

btnSave = new JButton("Save");

btnSave.setBounds(301, 580, 85, 21);

inbodyPanel.add(btnSave);

JLabel label\_7 = new JLabel("Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(412, 231, 62, 30);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_8.setBounds(22, 260, 45, 15);

inbodyPanel.add(label\_8);

JLabel label\_6 = new JLabel("Title");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(22, 300, 45, 15);

inbodyPanel.add(label\_6);

JLabel label\_5 = new JLabel("Author(s)");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(22, 340, 111, 15);

inbodyPanel.add(label\_5);

JLabel label\_3 = new JLabel("Publication Year");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(22, 380, 111, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Book Editor");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(22, 420, 111, 15);

inbodyPanel.add(label\_4);

JLabel label\_2 = new JLabel("Place of Publication");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(22, 460, 119, 15);

inbodyPanel.add(label\_2);

JLabel label\_1 = new JLabel("Publisher");

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_1.setBounds(22, 500, 119, 15);

inbodyPanel.add(label\_1);

JLabel label = new JLabel("Page No.");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(22, 540, 119, 15);

inbodyPanel.add(label);

tfID = new JTextField();

tfID.setEditable(false);

tfID.setColumns(10);

tfID.setBounds(192, 255, 369, 19);

inbodyPanel.add(tfID);

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(192, 295, 369, 19);

inbodyPanel.add(tfTitle);

tfAuthor = new JTextField();

tfAuthor.setColumns(10);

tfAuthor.setBounds(192, 335, 369, 19);

inbodyPanel.add(tfAuthor);

tfYear = new JTextField();

tfYear.setColumns(10);

tfYear.setBounds(192, 375, 369, 19);

inbodyPanel.add(tfYear);

tfEditor = new JTextField();

tfEditor.setColumns(10);

tfEditor.setBounds(192, 415, 369, 19);

inbodyPanel.add(tfEditor);

tfPlace = new JTextField();

tfPlace.setColumns(10);

tfPlace.setBounds(192, 455, 369, 19);

inbodyPanel.add(tfPlace);

tfPublisher = new JTextField();

tfPublisher.setColumns(10);

tfPublisher.setBounds(192, 495, 369, 19);

inbodyPanel.add(tfPublisher);

tfPage = new JTextField();

tfPage.setColumns(10);

tfPage.setBounds(192, 535, 369, 19);

inbodyPanel.add(tfPage);

JLabel l = new JLabel("Select Research ID to issue:");

l.setBounds(693, 255, 200, 19);

l.setFont(new Font("Calibri Light", Font.PLAIN, 15));

inbodyPanel.add(l);

researcherID = new JComboBox();

String[] ids =getReasearcherIds().toArray(new String[] {});

researcherID.setModel(new DefaultComboBoxModel(ids));

researcherID.setBounds(700, 295, 150, 19);

inbodyPanel.add(researcherID);

btnIssue = new JButton("Issue");

btnIssue.setBounds(750, 335, 100, 25);

inbodyPanel.add(btnIssue);

addPanel();

showBookChapterRow();

c.editBookChapters();

c.deleteBookChapters();

c.searchBook();

c.issuebook();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* View Journal Articles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void setJournalScrollPane(JScrollPane jsp) { this.journalScrollPane = jsp;}

public void setJournalTable(JTable tab) { this.journalTable = tab;}

public JTable getJournalTable(){ return journalTable;}

public JScrollPane getJournalScrollPane(){ return journalScrollPane;}

public void displayJournalArticles()

{

inBodyProperty();

//search box

tfSearch = new JTextField();

tfSearch.setBounds(650, 0, 150, 20);

tfSearch.setToolTipText("Enter article title to search");

inbodyPanel.add(tfSearch);

btnSearch= new JButton();

btnSearch.setIcon(new ImageIcon(LoginView.class.getResource("../imgs/find1.png")));

btnSearch.setBounds(800, -5, 50, 30);

btnSearch.setBorderPainted(false);

btnSearch.setContentAreaFilled(false);

inbodyPanel.add(btnSearch);

AdminController c= new AdminController(this);

c.viewJournalAricles();

inbodyPanel.add(getJournalScrollPane());

getJournalScrollPane().setBounds(10, 30, 850, 200);

btnDelete = new JButton("Delete");

btnDelete.setBounds(413, 580, 85, 21);

inbodyPanel.add(btnDelete);

btnSave = new JButton("Save");

btnSave.setBounds(301, 580, 85, 21);

inbodyPanel.add(btnSave);

JLabel l1 = new JLabel("Journal Articles");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

JLabel label\_7 = new JLabel("Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(412, 231, 62, 30);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_8.setBounds(22, 260, 45, 15);

inbodyPanel.add(label\_8);

JLabel label\_6 = new JLabel("Article Title");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(22, 300, 80, 15);

inbodyPanel.add(label\_6);

JLabel label\_5 = new JLabel("Journal Title");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(22, 340, 111, 15);

inbodyPanel.add(label\_5);

JLabel label\_3 = new JLabel("Publication Year");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(22, 380, 111, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Volume");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(22, 420, 111, 15);

inbodyPanel.add(label\_4);

JLabel label\_2 = new JLabel("Issue no.");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(22, 460, 119, 15);

inbodyPanel.add(label\_2);

JLabel label\_1 = new JLabel("Author(s)");

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_1.setBounds(22, 500, 119, 15);

inbodyPanel.add(label\_1);

JLabel label = new JLabel("Page No.");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(22, 540, 119, 15);

inbodyPanel.add(label);

tfID = new JTextField();

tfID.setEditable(false);

tfID.setColumns(10);

tfID.setBounds(192, 255, 369, 19);

inbodyPanel.add(tfID);

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(192, 295, 369, 19);

inbodyPanel.add(tfTitle);

tfJTitle = new JTextField();

tfJTitle.setColumns(10);

tfJTitle.setBounds(192, 335, 369, 19);

inbodyPanel.add(tfJTitle);

tfYear = new JTextField();

tfYear.setColumns(10);

tfYear.setBounds(192, 375, 369, 19);

inbodyPanel.add(tfYear);

tfVolume = new JTextField();

tfVolume.setColumns(10);

tfVolume.setBounds(192, 415, 369, 19);

inbodyPanel.add(tfVolume);

tfIssue = new JTextField();

tfIssue.setColumns(10);

tfIssue.setBounds(192, 455, 369, 19);

inbodyPanel.add(tfIssue);

tfAuthor = new JTextField();

tfAuthor.setColumns(10);

tfAuthor.setBounds(192, 495, 369, 19);

inbodyPanel.add(tfAuthor);

tfPage = new JTextField();

tfPage.setColumns(10);

tfPage.setBounds(192, 535, 369, 19);

inbodyPanel.add(tfPage);

JLabel l = new JLabel("Select Research ID to issue:");

l.setBounds(693, 255, 200, 19);

l.setFont(new Font("Calibri Light", Font.PLAIN, 15));

inbodyPanel.add(l);

researcherID = new JComboBox();

String[] ids =getReasearcherIds().toArray(new String[] {});

researcherID.setModel(new DefaultComboBoxModel(ids));

researcherID.setBounds(700, 295, 150, 19);

inbodyPanel.add(researcherID);

btnIssue = new JButton("Issue");

btnIssue.setBounds(750, 335, 100, 25);

inbodyPanel.add(btnIssue);

addPanel();

showJournalRow();

c.editJournalArticles();

c.deleteJournalArticles();

c.searchJournalArticle();

c.issuebook();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* View Conference Articles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void setConferenceScrollPane(JScrollPane jsp) { this.conferenceScrollPane = jsp;}

public void setConferenceTable(JTable tab) { this.conferenceTable = tab;}

public JTable getconferenceTable(){ return conferenceTable;}

public JScrollPane getconferenceScrollPane(){ return conferenceScrollPane;}

@SuppressWarnings({ "rawtypes", "unchecked" })

public void displayConferenceArticle()

{

inBodyProperty();

//search box

tfSearch = new JTextField();

tfSearch.setBounds(650, 0, 150, 20);

tfSearch.setToolTipText("Enter article title to search");

inbodyPanel.add(tfSearch);

btnSearch= new JButton();

btnSearch.setIcon(new ImageIcon(LoginView.class.getResource("../imgs/find1.png")));

btnSearch.setBounds(800, -5, 50, 30);

btnSearch.setBorderPainted(false);

btnSearch.setContentAreaFilled(false);

inbodyPanel.add(btnSearch);

JLabel l1 = new JLabel("Coference Articles");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

AdminController c= new AdminController(this);

c.viewConferenceArticles();

inbodyPanel.add(getconferenceScrollPane());

getconferenceScrollPane().setBounds(10, 30, 850, 200);

btnDelete = new JButton("Delete");

btnDelete.setBounds(413, 580, 85, 21);

inbodyPanel.add(btnDelete);

btnSave = new JButton("Save");

btnSave.setBounds(301, 580, 85, 21);

inbodyPanel.add(btnSave);

JLabel label\_7 = new JLabel("Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(412, 231, 62, 30);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_8.setBounds(22, 260, 45, 15);

inbodyPanel.add(label\_8);

JLabel label\_6 = new JLabel("Title");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(22, 300, 45, 15);

inbodyPanel.add(label\_6);

JLabel label\_5 = new JLabel("Author(s)");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(22, 340, 111, 15);

inbodyPanel.add(label\_5);

JLabel label\_3 = new JLabel("Publication Year");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(22, 380, 111, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Book Editor");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(22, 420, 111, 15);

inbodyPanel.add(label\_4);

JLabel label\_2 = new JLabel("Conference Location");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(22, 460, 140, 15);

inbodyPanel.add(label\_2);

JLabel label = new JLabel("Page No.");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(22, 500, 119, 15);

inbodyPanel.add(label);

tfID = new JTextField();

tfID.setEditable(false);

tfID.setColumns(10);

tfID.setBounds(192, 255, 369, 19);

inbodyPanel.add(tfID);

tfTitle = new JTextField();

tfTitle.setColumns(10);

tfTitle.setBounds(192, 295, 369, 19);

inbodyPanel.add(tfTitle);

tfAuthor = new JTextField();

tfAuthor.setColumns(10);

tfAuthor.setBounds(192, 335, 369, 19);

inbodyPanel.add(tfAuthor);

tfYear = new JTextField();

tfYear.setColumns(10);

tfYear.setBounds(192, 375, 369, 19);

inbodyPanel.add(tfYear);

tfEditor = new JTextField();

tfEditor.setColumns(10);

tfEditor.setBounds(192, 415, 369, 19);

inbodyPanel.add(tfEditor);

tfPlace = new JTextField();

tfPlace.setColumns(10);

tfPlace.setBounds(192, 455, 369, 19);

inbodyPanel.add(tfPlace);

tfPage = new JTextField();

tfPage.setColumns(10);

tfPage.setBounds(192, 495, 369, 19);

inbodyPanel.add(tfPage);

//issue section

JLabel l = new JLabel("Select Research ID to issue:");

l.setBounds(693, 255, 200, 19);

l.setFont(new Font("Calibri Light", Font.PLAIN, 15));

inbodyPanel.add(l);

researcherID = new JComboBox();

String[] ids =getReasearcherIds().toArray(new String[] {});

researcherID.setModel(new DefaultComboBoxModel(ids));

researcherID.setBounds(700, 295, 150, 19);

inbodyPanel.add(researcherID);

btnIssue= new JButton("Issue");

btnIssue.setBounds(750, 335, 100, 25);

inbodyPanel.add(btnIssue);

addPanel();

showConferenceRow();

c.editConfernceArticles();

c.deleteConferenceArticles();

c.searchConferenceArticle();

c.issuebook();

}

/\*\*

\* \*\*\* Accessors \*\*\*\*\*\*\*

\*/

//menu

public JMenuItem getmntmSignOut() { return mntmSignOut; }

//buttons

public JButton getbtnAddResearcher(){ return btnAddResearcher; }

public JButton getbtnViewResearcher(){ return btnViewResearcher;}

public JButton getbtnAddBookChapter(){ return btnAddBookChapter;}

public JButton getbtnAddJournalArticle(){ return btnAddJournalArticle;}

public JButton getbtnAddConferenceArticle(){ return btnAddConferenceArticle;}

public JButton getbtnViewBookChapter(){ return btnViewBookChapter;}

public JButton getbtnViewJournalArticle(){ return btnViewJournalArticle;}

public JButton getbtnViewConferenceArticle() { return btnViewConferenceArticle; }

public JButton getbtnViewIssuedArticles() { return btnViewIssuedArticles; }

public JButton getbtnAddJArticle(){ return btnAddJArticle;}

public JButton getbtnAddCArticle(){ return btnAddCArticle;}

public JButton getbtnAddBook(){ return btnAddBook;}

public JButton getbtnSave() { return btnSave; }

public JButton getbtnDelete() { return btnDelete; }

public JButton getbtnSearch() { return btnSearch; }

public JButton getbtnIssue() { return btnIssue; }

public JButton getbtnReturn() { return btnReturn; }

//textfields

public JTextField gettfID() { return tfID; }

public JTextField gettfPublisher() { return tfPublisher; }

public JTextField gettfTitle() { return tfTitle; }

public JTextField gettfAddress() { return tfAddress; }

public JTextField gettfContact() { return tfContact; }

public JTextField gettfEmail() { return tfEmail; }

public JTextField gettfUser() { return tfUser; }

public JTextField gettfPassword() { return tfPassword; }

public JTextField gettfArticleTitle() { return tfArticleTitle; }

public JTextField gettfJTitle() { return tfJTitle; }

public JTextField gettfYear() { return tfYear; }

public JTextField gettfVolume() { return tfVolume; }

public JTextField gettfIssue() { return tfIssue; }

public JTextField gettfAuthor() { return tfAuthor; }

public JTextField gettfPage() { return tfPage; }

public JTextField gettfEditor() { return tfEditor; }

public JTextField gettfPlace() { return tfPlace; }

public JTextField gettfSearch() { return tfSearch; }

public JComboBox getReID() { return researcherID; }

//table row index

public int getEditIndex() { return editIndex; }

public void setEditIndex(int i) {editIndex= i;}

/\*

\* row setters

\*/

//researchers

public void showResearcherCell(String id, String name, String address, String contact, String email, String username, String password)

{ this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAddress.setText(address);

this.tfContact.setText(contact);

this.tfEmail.setText(email);

this.tfUser.setText(username);

this.tfPassword.setText(password);

}

//book chapter

public void showBookcell(String id, String name, String author, String year, String editor, String place, String publisher, String page)

{

this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAuthor.setText(author);

this.tfYear.setText(year);

this.tfEditor.setText(editor);

this.tfPlace.setText(place);

this.tfPublisher.setText(publisher);

this.tfPage.setText(page);

}

//journal Article

public void showJournalCell(String id, String name, String journalTitle, String year, String volume, String issue, String author, String page)

{

this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAuthor.setText(author);

this.tfYear.setText(year);

this.tfVolume.setText(volume);

this.tfIssue.setText(issue);

this.tfJTitle.setText(journalTitle);

this.tfPage.setText(page);

}

//conference article

public void showConferenceCell(String id, String name, String author, String year, String editor, String place, String page)

{

this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAuthor.setText(author);

this.tfYear.setText(year);

this.tfEditor.setText(editor);

this.tfPlace.setText(place);

this.tfPage.setText(page);

}

/\*

\* row selector of table

\*/

//researcher table

public void showResearcherRowDetails()

{

resTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAddress,valContact,valEmail,valUser, valPassword;

DefaultTableModel resModel = (DefaultTableModel)resTable.getModel();

indexRow=resTable.getSelectedRow();

valID= resModel.getValueAt(indexRow, 0).toString();

valName= resModel.getValueAt(indexRow,1).toString();

valAddress= resModel.getValueAt(indexRow,2).toString();

valContact= resModel.getValueAt(indexRow,3).toString();

valEmail= resModel.getValueAt(indexRow,4).toString();

valUser= resModel.getValueAt(indexRow,5).toString();

valPassword= resModel.getValueAt(indexRow,6).toString();

showResearcherCell(valID, valName, valAddress, valContact, valEmail, valUser, valPassword);

setEditIndex(resTable.getSelectedRow());

}

});

}

//book chapter table

public void showBookChapterRow()

{

bookTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAuthor,valYear,valEditor,valPlace, valPublisher, valPage;

DefaultTableModel bookMdl = (DefaultTableModel)bookTable.getModel();

indexRow=bookTable.getSelectedRow();

valID= bookMdl.getValueAt(indexRow, 1).toString();

valName= bookMdl.getValueAt(indexRow,2).toString();

valAuthor= bookMdl.getValueAt(indexRow,3).toString();

valYear= bookMdl.getValueAt(indexRow,4).toString();

valEditor= bookMdl.getValueAt(indexRow,5).toString();

valPlace= bookMdl.getValueAt(indexRow,6).toString();

valPublisher= bookMdl.getValueAt(indexRow,7).toString();

valPage= bookMdl.getValueAt(indexRow,8).toString();

showBookcell(valID, valName, valAuthor, valYear, valEditor, valPlace, valPublisher, valPage);

setEditIndex(bookTable.getSelectedRow());

}

});

}

//journal articles table

public void showJournalRow()

{

journalTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAuthor,valYear,valVolume,valIssue, valJournalTitle, valPage;

DefaultTableModel jmodel = (DefaultTableModel)journalTable.getModel();

indexRow=journalTable.getSelectedRow();

valID= jmodel.getValueAt(indexRow, 1).toString();

valName= jmodel.getValueAt(indexRow,2).toString();

valJournalTitle= jmodel.getValueAt(indexRow,3).toString();

valYear= jmodel.getValueAt(indexRow,4).toString();

valVolume= jmodel.getValueAt(indexRow,5).toString();

valIssue= jmodel.getValueAt(indexRow,6).toString();

valAuthor= jmodel.getValueAt(indexRow,7).toString();

valPage= jmodel.getValueAt(indexRow,8).toString();

showJournalCell(valID, valName, valJournalTitle, valYear, valVolume, valIssue, valAuthor, valPage);

setEditIndex(journalTable.getSelectedRow());

}

});

}

//conference article

public void showConferenceRow()

{

conferenceTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAuthor,valYear,valEditor,valPlace, valPage;

DefaultTableModel bookMdl = (DefaultTableModel)conferenceTable.getModel();

indexRow=conferenceTable.getSelectedRow();

valID= bookMdl.getValueAt(indexRow, 1).toString();

valName= bookMdl.getValueAt(indexRow,2).toString();

valAuthor= bookMdl.getValueAt(indexRow,3).toString();

valYear= bookMdl.getValueAt(indexRow,4).toString();

valEditor= bookMdl.getValueAt(indexRow,5).toString();

valPlace= bookMdl.getValueAt(indexRow,6).toString();

valPage= bookMdl.getValueAt(indexRow,7).toString();

showConferenceCell(valID, valName, valAuthor, valYear, valEditor, valPlace, valPage);

setEditIndex(conferenceTable.getSelectedRow());

}

});

}

//methods

public void lblProperty()

{

lblID = new JLabel();

lbl2 = new JLabel();

lbl3 = new JLabel();

lbl4 = new JLabel();

lbl5 = new JLabel();

lbl6 = new JLabel();

lbl7 = new JLabel();

lbl8 = new JLabel();

inbodyPanel.add(lblID);

inbodyPanel.add(lbl2);

inbodyPanel.add(lbl3);

inbodyPanel.add(lbl4);

inbodyPanel.add(lbl5);

inbodyPanel.add(lbl6);

inbodyPanel.add(lbl7);

inbodyPanel.add(lbl8);

lblID.setBounds(358, 92, 369, 19);

lbl2.setBounds(358, 135, 369, 19);

lbl3.setBounds(358, 181, 369, 19);

lbl4.setBounds(358, 225, 369, 19);

lbl5.setBounds(358, 271, 369, 19);

lbl6.setBounds(358, 317, 369, 19);

lbl7.setBounds(358, 366, 369, 19);

lbl8.setBounds(358, 415, 500, 19);

}

//search results deatails panel

public void searchResearcherPanel()

{

inBodyProperty();

JLabel label\_7 = new JLabel("Researcher Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(352, 26, 224, 37);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

JLabel label\_6 = new JLabel("Name");

JLabel label\_5 = new JLabel("Address");

JLabel label\_3 = new JLabel("Contact");

JLabel label\_4 = new JLabel("Email");

JLabel label\_2 = new JLabel("Username");

JLabel label\_1 = new JLabel("Password");

JLabel articles = new JLabel("Articles / Chapters Hired:");

JButton btnBack = new JButton("Back");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_8.setBounds(188, 96, 150, 20);

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_6.setBounds(188, 139, 150, 20);

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_5.setBounds(188, 185, 150, 20);

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_3.setBounds(188, 229, 150, 20);

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_4.setBounds(188, 275, 150, 20);

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_2.setBounds(188, 321, 150, 20);

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_1.setBounds(188, 370, 150, 20);

articles.setFont(new Font("Calibri Light", Font.PLAIN, 16));

articles.setBounds(188, 415, 200, 20);

btnBack.setBounds(600, 500, 100, 25);

lbl9 = new JTextArea();

lbl9.setBounds(358, 415, 495, 50);

lbl9.setLineWrap(true);

inbodyPanel.add(label\_8);

inbodyPanel.add(label\_6);

inbodyPanel.add(label\_5);

inbodyPanel.add(label\_3);

inbodyPanel.add(label\_4);

inbodyPanel.add(label\_2);

inbodyPanel.add(label\_1);

inbodyPanel.add(articles);

inbodyPanel.add(lbl9);

inbodyPanel.add(btnBack);

lblProperty();

addPanel();

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {displayResearchers();}

});

}

public void bookDetailPanel()

{

inBodyProperty();

JLabel label\_7 = new JLabel("Book Chapter Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(352, 26, 224, 37);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

JLabel label\_6 = new JLabel("Title");

JLabel label\_5 = new JLabel("Author(s)");

JLabel label\_3 = new JLabel("Publication Year");

JLabel label\_4 = new JLabel("Book Editor");

JLabel label\_2 = new JLabel("Place of Publication");

JLabel label\_1 = new JLabel("Publisher");

JLabel label = new JLabel("Page No.");

JButton btnBack = new JButton("Back");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_8.setBounds(188, 96, 150, 20);

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_6.setBounds(188, 139, 150, 20);

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_5.setBounds(188, 185, 150, 20);

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_3.setBounds(188, 229, 150, 20);

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_4.setBounds(188, 275, 150, 20);

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_2.setBounds(188, 321, 150, 20);

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_1.setBounds(188, 370, 150, 20);

label.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label.setBounds(188, 419, 150, 20);

btnBack.setBounds(600, 500, 100, 25);

inbodyPanel.add(label\_8);

inbodyPanel.add(label\_6);

inbodyPanel.add(label\_5);

inbodyPanel.add(label\_3);

inbodyPanel.add(label\_4);

inbodyPanel.add(label\_2);

inbodyPanel.add(label\_1);

inbodyPanel.add(label);

inbodyPanel.add(btnBack);

lblProperty();

addPanel();

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {displayBookChapter();}

});

}

public void searchJournalPanel()

{

inBodyProperty();

JLabel label\_7 = new JLabel("Article Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(352, 26, 224, 37);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

JLabel label\_6 = new JLabel("Article Title");

JLabel label\_5 = new JLabel("Journal Title");

JLabel label\_3 = new JLabel("Publication Year");

JLabel label\_4 = new JLabel("Volume");

JLabel label\_2 = new JLabel("Issue No.");

JLabel label\_1 = new JLabel("Author(s)");

JLabel label = new JLabel("Page No.");

JButton btnBack = new JButton("Back");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_8.setBounds(188, 96, 150, 20);

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_6.setBounds(188, 139, 150, 20);

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_5.setBounds(188, 185, 150, 20);

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_3.setBounds(188, 229, 150, 20);

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_4.setBounds(188, 275, 150, 20);

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_2.setBounds(188, 321, 150, 20);

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_1.setBounds(188, 370, 150, 20);

label.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label.setBounds(188, 419, 150, 20);

btnBack.setBounds(600, 500, 100, 25);

inbodyPanel.add(label\_8);

inbodyPanel.add(label\_6);

inbodyPanel.add(label\_5);

inbodyPanel.add(label\_3);

inbodyPanel.add(label\_4);

inbodyPanel.add(label\_2);

inbodyPanel.add(label\_1);

inbodyPanel.add(label);

inbodyPanel.add(btnBack);

lblProperty();

addPanel();

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {displayJournalArticles();}

});

}

public void searchConferencePanel()

{

inBodyProperty();

JLabel label\_7 = new JLabel("Article Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(352, 26, 224, 37);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

JLabel label\_6 = new JLabel("Title");

JLabel label\_5 = new JLabel("Author(s)");

JLabel label\_3 = new JLabel("Publication Year");

JLabel label\_4 = new JLabel("Editor(s)");

JLabel label\_2 = new JLabel("Conference Location");

JLabel label = new JLabel("Page No.");

JButton btnBack = new JButton("Back");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_8.setBounds(188, 96, 150, 20);

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_6.setBounds(188, 139, 150, 20);

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_5.setBounds(188, 185, 150, 20);

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_3.setBounds(188, 229, 150, 20);

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_4.setBounds(188, 275, 150, 20);

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label\_2.setBounds(188, 321, 150,20);

label.setFont(new Font("Calibri Light", Font.PLAIN, 16));

label.setBounds(188, 370, 150, 20);

btnBack.setBounds(600, 500, 100, 25);

inbodyPanel.add(label\_8);

inbodyPanel.add(label\_6);

inbodyPanel.add(label\_5);

inbodyPanel.add(label\_3);

inbodyPanel.add(label\_4);

inbodyPanel.add(label\_2);

inbodyPanel.add(label);

inbodyPanel.add(btnBack);

lblProperty();

addPanel();

btnBack.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {displayConferenceArticle();}

});}

public JLabel getlblId() {return lblID; }

public JLabel getlbl2() {return lbl2; }

public JLabel getlbl3() {return lbl3; }

public JLabel getlbl4() {return lbl4; }

public JLabel getlbl5() {return lbl5; }

public JLabel getlbl6() {return lbl6; }

public JLabel getlbl7() {return lbl7; }

public JLabel getlbl8() {return lbl8; }

public JTextArea getlbl9() {return lbl9; }

// arraylist of researchers id required for issue of articles

public ArrayList<String> getReasearcherIds() {

ArrayList<String> ids= new ArrayList<String>();

if(mod.readData()!=null) {

for(int i=0;i<mod.readData().size();i++)

{ids.add(mod.readData().get(i).getId());}}

return ids; }

// view articles issued and researchers id

public void setHireScrollPane(JScrollPane jsp) { this.hireScrollPane = jsp;}

public void setHireTable(JTable tab) { this.hiresTable = tab;}

public JTable getHireTable(){ return hiresTable;}

public JScrollPane getHireScrollPane(){ return hireScrollPane;}

public void displayIssuedPanel()

{

inBodyProperty();

JLabel l1 = new JLabel("Hires");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

AdminController c= new AdminController(this);

c.viewHires();

inbodyPanel.add(getHireScrollPane());

getHireScrollPane().setBounds(10, 30, 850, 500);

btnReturn= new JButton("Return");

btnReturn.setBounds(750, 555, 100, 25);

inbodyPanel.add(btnReturn);

tfID =new JTextField();

tfID.setEditable(false);

tfID.setBounds(10, 555, 150, 25);

inbodyPanel.add(tfID);

addPanel();

issueRowSelect();

c.returnHire();

}

public void issueRowSelect()

{

hiresTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String rid ,aid;

DefaultTableModel jmodel = (DefaultTableModel)hiresTable.getModel();

indexRow=hiresTable.getSelectedRow();

rid= jmodel.getValueAt(indexRow, 0).toString();

aid= jmodel.getValueAt(indexRow,1).toString();

tfID.setText(aid);

setEditIndex(hiresTable.getSelectedRow());

}

});

}

}

package View;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import Controller.\*;

import Login.\*;

import Model.\*;

public class ResearcherView extends JFrame{

private static final long serialVersionUID = 1L;

ResearcherController rec = new ResearcherController(this);

//panel

JPanel bodyPanel, inbodyPanel;

//menus

private JMenu mnFile;

private JMenuItem mntmSignOut;

//buttons

private JButton btnViewHired, btnViewBookChapter, btnViewJournalArticle, btnViewConferenceArticle;

private JLabel tfID,tfPublisher,tfTitle, tfJTitle, tfYear, tfVolume, tfIssue, tfAuthor, tfPage, tfEditor, tfPlace;

private JTable bookTable, journalTable, conferenceTable, hiresTable;

private JScrollPane bookScrollPane, journalScrollPane, conferenceScrollPane, hireScrollPane;

private String a;

//public void trying(String a) { this.a= a;}

public String getTrying() {return this.a;}

public ResearcherView(String a)

{

this.a = a;

setTitle("Northampton Research Center - Researcher");

setIconImage(new ImageIcon(LoginView.class.getResource("../imgs/ICON.jpg")).getImage());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBounds(100, 100, 1080, 700);

getContentPane().setBackground(Color.WHITE);

getContentPane().setLayout(null);

//body

bodyPanel = new JPanel();

bodyPanel.setBackground(Color.WHITE);

bodyPanel.setBounds(177, 20, 890, 645);

getContentPane().add(bodyPanel);

bodyPanel.setLayout(null);

inbodyPanel= new JPanel();

JMenuBar menuBar = new JMenuBar();

menuBar.setBackground(Color.LIGHT\_GRAY);

menuBar.setBounds(0, 0, 1066, 20);

getContentPane().add(menuBar);

mnFile = new JMenu("File");

menuBar.add(mnFile);

mntmSignOut = new JMenuItem("Sign Out");

mnFile.add(mntmSignOut);

JPanel menuPanel = new JPanel();

menuPanel.setBackground(new Color(216, 191, 216));

menuPanel.setBounds(0, 20, 180, 642);

getContentPane().add(menuPanel);

menuPanel.setLayout(null);

JLabel sidelabel = new JLabel(new ImageIcon(AdminView.class.getResource("../imgs/menu.jpg")));

sidelabel.setBounds(0, -15, 180, 669);

menuPanel.add(sidelabel);

btnViewHired = new JButton("View Hired Articles");

sidelabel.add(btnViewHired);

btnViewHired.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewHired.setBorderPainted(false);

btnViewHired.setBackground(new Color(0, 191, 255));

btnViewHired.setBounds(22, 89, 135, 34);

btnViewBookChapter = new JButton("View Book Chapters");

sidelabel.add(btnViewBookChapter);

btnViewBookChapter.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewBookChapter.setBorderPainted(false);

btnViewBookChapter.setBackground(new Color(0, 191, 255));

btnViewBookChapter.setBounds(22, 133, 135, 34);

btnViewJournalArticle = new JButton("View Journal Articles");

sidelabel.add(btnViewJournalArticle);

btnViewJournalArticle.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewJournalArticle.setBorderPainted(false);

btnViewJournalArticle.setBackground(new Color(0, 191, 255));

btnViewJournalArticle.setBounds(22, 177, 135, 34);

btnViewConferenceArticle = new JButton("View Conference Articles");

sidelabel.add(btnViewConferenceArticle);

btnViewConferenceArticle.setFont(new Font("Tahoma", Font.PLAIN, 9));

btnViewConferenceArticle.setBorderPainted(false);

btnViewConferenceArticle.setBackground(new Color(0, 191, 255));

btnViewConferenceArticle.setBounds(22, 221, 135, 34);

JLabel lblwelcom= new JLabel("Welcome to the Northampton Research Center Library");

lblwelcom.setFont(new Font("Tahoma", Font.PLAIN, 30));

lblwelcom.setBounds(10,50,900,500);

bodyPanel.add(lblwelcom);

setVisible(true);

rec.settrying(getTrying());

rec.operate();

}

public void addPanel()

{

bodyPanel.removeAll();

bodyPanel.add(inbodyPanel);

bodyPanel.revalidate();

inbodyPanel.repaint();

bodyPanel.repaint();

}

public void inBodyProperty()

{

inbodyPanel.setBackground(Color.WHITE);

inbodyPanel.setBounds(10, 20, 890, 645);

inbodyPanel.setLayout(null);

inbodyPanel.removeAll();

}

/\*

\*View Book Chapters

\* \*/

public void setBookScrollPane(JScrollPane jsp) { this.bookScrollPane = jsp;}

public void setBookTable(JTable tab) { this.bookTable = tab;}

public JTable getBookTable(){ return bookTable;}

public JScrollPane getBookScrollPane(){ return bookScrollPane;}

public void displayBookChapter()

{

inBodyProperty();

JLabel l1 = new JLabel("Book Chapters");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

rec.viewBookChapters();

inbodyPanel.add(getBookScrollPane());

getBookScrollPane().setBounds(10, 30, 850, 200);

JLabel label\_7 = new JLabel("Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(412, 231, 62, 30);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_8.setBounds(22, 260, 45, 15);

inbodyPanel.add(label\_8);

JLabel label\_6 = new JLabel("Title");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(22, 300, 45, 15);

inbodyPanel.add(label\_6);

JLabel label\_5 = new JLabel("Author(s)");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(22, 340, 111, 15);

inbodyPanel.add(label\_5);

JLabel label\_3 = new JLabel("Publication Year");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(22, 380, 111, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Book Editor");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(22, 420, 111, 15);

inbodyPanel.add(label\_4);

JLabel label\_2 = new JLabel("Place of Publication");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(22, 460, 119, 15);

inbodyPanel.add(label\_2);

JLabel label\_1 = new JLabel("Publisher");

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_1.setBounds(22, 500, 119, 15);

inbodyPanel.add(label\_1);

JLabel label = new JLabel("Page No.");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(22, 540, 119, 15);

inbodyPanel.add(label);

tfID = new JLabel();

tfID.setBounds(192, 255, 369, 19);

inbodyPanel.add(tfID);

tfTitle = new JLabel();

tfTitle.setBounds(192, 295, 369, 19);

inbodyPanel.add(tfTitle);

tfAuthor = new JLabel();

tfAuthor.setBounds(192, 335, 369, 19);

inbodyPanel.add(tfAuthor);

tfYear = new JLabel();

tfYear.setBounds(192, 375, 369, 19);

inbodyPanel.add(tfYear);

tfEditor = new JLabel();

tfEditor.setBounds(192, 415, 369, 19);

inbodyPanel.add(tfEditor);

tfPlace = new JLabel();

tfPlace.setBounds(192, 455, 369, 19);

inbodyPanel.add(tfPlace);

tfPublisher = new JLabel();

tfPublisher.setBounds(192, 495, 369, 19);

inbodyPanel.add(tfPublisher);

tfPage = new JLabel();

tfPage.setBounds(192, 535, 369, 19);

inbodyPanel.add(tfPage);

addPanel();

showBookChapterRow();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* View Journal Articles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void setJournalScrollPane(JScrollPane jsp) { this.journalScrollPane = jsp;}

public void setJournalTable(JTable tab) { this.journalTable = tab;}

public JTable getJournalTable(){ return journalTable;}

public JScrollPane getJournalScrollPane(){ return journalScrollPane;}

public void displayJournalArticles()

{

inBodyProperty();

rec.viewJournalAricles();

inbodyPanel.add(getJournalScrollPane());

getJournalScrollPane().setBounds(10, 30, 850, 200);

JLabel l1 = new JLabel("Journal Articles");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

JLabel label\_7 = new JLabel("Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(412, 231, 62, 30);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_8.setBounds(22, 260, 45, 15);

inbodyPanel.add(label\_8);

JLabel label\_6 = new JLabel("Article Title");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(22, 300, 80, 15);

inbodyPanel.add(label\_6);

JLabel label\_5 = new JLabel("Journal Title");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(22, 340, 111, 15);

inbodyPanel.add(label\_5);

JLabel label\_3 = new JLabel("Publication Year");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(22, 380, 111, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Volume");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(22, 420, 111, 15);

inbodyPanel.add(label\_4);

JLabel label\_2 = new JLabel("Issue no.");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(22, 460, 119, 15);

inbodyPanel.add(label\_2);

JLabel label\_1 = new JLabel("Author(s)");

label\_1.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_1.setBounds(22, 500, 119, 15);

inbodyPanel.add(label\_1);

JLabel label = new JLabel("Page No.");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(22, 540, 119, 15);

inbodyPanel.add(label);

tfID = new JLabel();

tfID.setBounds(192, 255, 369, 19);

inbodyPanel.add(tfID);

tfTitle = new JLabel();

tfTitle.setBounds(192, 295, 369, 19);

inbodyPanel.add(tfTitle);

tfJTitle = new JLabel();

tfJTitle.setBounds(192, 335, 369, 19);

inbodyPanel.add(tfJTitle);

tfYear = new JLabel();

tfYear.setBounds(192, 375, 369, 19);

inbodyPanel.add(tfYear);

tfVolume = new JLabel();

tfVolume.setBounds(192, 415, 369, 19);

inbodyPanel.add(tfVolume);

tfIssue = new JLabel();

tfIssue.setBounds(192, 455, 369, 19);

inbodyPanel.add(tfIssue);

tfAuthor = new JLabel();

tfAuthor.setBounds(192, 495, 369, 19);

inbodyPanel.add(tfAuthor);

tfPage = new JLabel();

tfPage.setBounds(192, 535, 369, 19);

inbodyPanel.add(tfPage);

addPanel();

showJournalRow();

}

/\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* View Conference Articles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/

public void setConferenceScrollPane(JScrollPane jsp) { this.conferenceScrollPane = jsp;}

public void setConferenceTable(JTable tab) { this.conferenceTable = tab;}

public JTable getconferenceTable(){ return conferenceTable;}

public JScrollPane getconferenceScrollPane(){ return conferenceScrollPane;}

@SuppressWarnings({ "rawtypes", "unchecked" })

public void displayConferenceArticle()

{

inBodyProperty();

JLabel l1 = new JLabel("Coference Articles");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

rec.viewConferenceArticles();

inbodyPanel.add(getconferenceScrollPane());

getconferenceScrollPane().setBounds(10, 30, 850, 200);

JLabel label\_7 = new JLabel("Details");

label\_7.setForeground(new Color(0, 0, 139));

label\_7.setFont(new Font("Corbel Light", Font.PLAIN, 20));

label\_7.setBounds(412, 231, 62, 30);

inbodyPanel.add(label\_7);

JLabel label\_8 = new JLabel("ID ");

label\_8.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_8.setBounds(22, 260, 45, 15);

inbodyPanel.add(label\_8);

JLabel label\_6 = new JLabel("Title");

label\_6.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_6.setBounds(22, 300, 45, 15);

inbodyPanel.add(label\_6);

JLabel label\_5 = new JLabel("Author(s)");

label\_5.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_5.setBounds(22, 340, 111, 15);

inbodyPanel.add(label\_5);

JLabel label\_3 = new JLabel("Publication Year");

label\_3.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_3.setBounds(22, 380, 111, 15);

inbodyPanel.add(label\_3);

JLabel label\_4 = new JLabel("Book Editor");

label\_4.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_4.setBounds(22, 420, 111, 15);

inbodyPanel.add(label\_4);

JLabel label\_2 = new JLabel("Conference Location");

label\_2.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label\_2.setBounds(22, 460, 140, 15);

inbodyPanel.add(label\_2);

JLabel label = new JLabel("Page No.");

label.setFont(new Font("Calibri Light", Font.PLAIN, 15));

label.setBounds(22, 500, 119, 15);

inbodyPanel.add(label);

tfID = new JLabel();

tfID.setBounds(192, 255, 369, 19);

inbodyPanel.add(tfID);

tfTitle = new JLabel();

tfTitle.setBounds(192, 295, 369, 19);

inbodyPanel.add(tfTitle);

tfAuthor = new JLabel();

tfAuthor.setBounds(192, 335, 369, 19);

inbodyPanel.add(tfAuthor);

tfYear = new JLabel();

tfYear.setBounds(192, 375, 369, 19);

inbodyPanel.add(tfYear);

tfEditor = new JLabel();

tfEditor.setBounds(192, 415, 369, 19);

inbodyPanel.add(tfEditor);

tfPlace = new JLabel();

tfPlace.setBounds(192, 455, 369, 19);

inbodyPanel.add(tfPlace);

tfPage = new JLabel();

tfPage.setBounds(192, 495, 369, 19);

inbodyPanel.add(tfPage);

addPanel();

showConferenceRow();

}

//book chapter

public void showBookcell(String id, String name, String author, String year, String editor, String place, String publisher, String page)

{

this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAuthor.setText(author);

this.tfYear.setText(year);

this.tfEditor.setText(editor);

this.tfPlace.setText(place);

this.tfPublisher.setText(publisher);

this.tfPage.setText(page);

}

//journal Article

public void showJournalCell(String id, String name, String journalTitle, String year, String volume, String issue, String author, String page)

{

this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAuthor.setText(author);

this.tfYear.setText(year);

this.tfVolume.setText(volume);

this.tfIssue.setText(issue);

this.tfJTitle.setText(journalTitle);

this.tfPage.setText(page);

}

//conference article

public void showConferenceCell(String id, String name, String author, String year, String editor, String place, String page)

{

this.tfID.setText(id);

this.tfTitle.setText(name);

this.tfAuthor.setText(author);

this.tfYear.setText(year);

this.tfEditor.setText(editor);

this.tfPlace.setText(place);

this.tfPage.setText(page);

}

//details

public void showBookChapterRow()

{

bookTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAuthor,valYear,valEditor,valPlace, valPublisher, valPage;

DefaultTableModel bookMdl = (DefaultTableModel)bookTable.getModel();

indexRow=bookTable.getSelectedRow();

valID= bookMdl.getValueAt(indexRow, 0).toString();

valName= bookMdl.getValueAt(indexRow,1).toString();

valAuthor= bookMdl.getValueAt(indexRow,2).toString();

valYear= bookMdl.getValueAt(indexRow,3).toString();

valEditor= bookMdl.getValueAt(indexRow,4).toString();

valPlace= bookMdl.getValueAt(indexRow,5).toString();

valPublisher= bookMdl.getValueAt(indexRow,6).toString();

valPage= bookMdl.getValueAt(indexRow,7).toString();

showBookcell(valID, valName, valAuthor, valYear, valEditor, valPlace, valPublisher, valPage);

}

});

}

//journal articles table

public void showJournalRow()

{

journalTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAuthor,valYear,valVolume,valIssue, valJournalTitle, valPage;

DefaultTableModel jmodel = (DefaultTableModel)journalTable.getModel();

indexRow=journalTable.getSelectedRow();

valID= jmodel.getValueAt(indexRow, 0).toString();

valName= jmodel.getValueAt(indexRow,1).toString();

valJournalTitle= jmodel.getValueAt(indexRow,2).toString();

valYear= jmodel.getValueAt(indexRow,3).toString();

valVolume= jmodel.getValueAt(indexRow,4).toString();

valIssue= jmodel.getValueAt(indexRow,5).toString();

valAuthor= jmodel.getValueAt(indexRow,6).toString();

valPage= jmodel.getValueAt(indexRow,7).toString();

showJournalCell(valID, valName, valJournalTitle, valYear, valVolume, valIssue, valAuthor, valPage);

}

});

}

//conference article

public void showConferenceRow()

{

conferenceTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String valID ,valName,valAuthor,valYear,valEditor,valPlace, valPage;

DefaultTableModel bookMdl = (DefaultTableModel)conferenceTable.getModel();

indexRow=conferenceTable.getSelectedRow();

valID= bookMdl.getValueAt(indexRow, 0).toString();

valName= bookMdl.getValueAt(indexRow,1).toString();

valAuthor= bookMdl.getValueAt(indexRow,2).toString();

valYear= bookMdl.getValueAt(indexRow,3).toString();

valEditor= bookMdl.getValueAt(indexRow,4).toString();

valPlace= bookMdl.getValueAt(indexRow,5).toString();

valPage= bookMdl.getValueAt(indexRow,6).toString();

showConferenceCell(valID, valName, valAuthor, valYear, valEditor, valPlace, valPage);

}

});

}

public void setHireScrollPane(JScrollPane jsp) { this.hireScrollPane = jsp;}

public void setHireTable(JTable tab) { this.hiresTable = tab;}

public JTable getHireTable(){ return hiresTable;}

public JScrollPane getHireScrollPane(){ return hireScrollPane;}

public void displayIssuedPanel()

{

inBodyProperty();

JLabel l1 = new JLabel("Hires");

inbodyPanel.add(l1);

l1.setBounds(8, 0, 150, 20);

l1.setFont(new Font("Corbel Light", Font.PLAIN, 20));

rec.viewHires();

inbodyPanel.add(getHireScrollPane());

getHireScrollPane().setBounds(10, 30, 850, 500);

addPanel();

issueRowSelect();

}

public void issueRowSelect()

{

hiresTable.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e)

{

int indexRow;

String rid ,aid;

DefaultTableModel jmodel = (DefaultTableModel)hiresTable.getModel();

indexRow=hiresTable.getSelectedRow();

rid= jmodel.getValueAt(indexRow, 0).toString();

aid= jmodel.getValueAt(indexRow,1).toString();

}

});

}

//getters

public JMenuItem getmntmSignOut() { return mntmSignOut; }

public JButton getbtnViewHired() { return btnViewHired; }

public JButton getbtnViewBookChapter(){ return btnViewBookChapter;}

public JButton getbtnViewJournalArticle(){ return btnViewJournalArticle;}

public JButton getbtnViewConferenceArticle() { return btnViewConferenceArticle; }

}

package Model;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.util.ArrayList;

import Serializable.BookChapters;

import Serializable.ConferenceArticles;

import Serializable.IssueRecords;

import Serializable.JournalArticles;

import Serializable.Researchers;

public class DataReader {

/\*\*

\* \*\*\*\*\*Reading files \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

//researchers

@SuppressWarnings("unchecked")

public ArrayList<Researchers> readData()

{

ArrayList<Researchers> researchersList = null;

try{

FileInputStream fileinput = new FileInputStream("src/files/researchers.dat");

ObjectInputStream objectinput = new ObjectInputStream(fileinput);

Object objR= objectinput.readObject();

researchersList = (ArrayList<Researchers>)objR;

objectinput.close();

}

catch (FileNotFoundException e){e.printStackTrace();}

catch (IOException e){}

catch (ClassNotFoundException e){e.printStackTrace();}

return researchersList;

}

//book chapters

@SuppressWarnings("unchecked")

public ArrayList<BookChapters> readBookData()

{

ArrayList<BookChapters> bookChapterLists = null;

try{

FileInputStream fileinput = new FileInputStream("src/files/bookchapters.dat");

ObjectInputStream objectinput = new ObjectInputStream(fileinput);

Object objR= objectinput.readObject();

bookChapterLists = (ArrayList<BookChapters>)objR;

objectinput.close();

}

catch (FileNotFoundException e){e.printStackTrace();}

catch (IOException e){}

catch (ClassNotFoundException e){e.printStackTrace();}

return bookChapterLists;

}

//journal Articles

@SuppressWarnings("unchecked")

public ArrayList<JournalArticles> readJournalData()

{

ArrayList<JournalArticles> journalLists = null;

try{

FileInputStream fileinput = new FileInputStream("src/files/JournalArticles.dat");

ObjectInputStream objectinput = new ObjectInputStream(fileinput);

Object objR= objectinput.readObject();

journalLists = (ArrayList<JournalArticles>)objR;

objectinput.close();

}

catch (FileNotFoundException e){e.printStackTrace();}

catch (IOException e){}

catch (ClassNotFoundException e){e.printStackTrace();}

return journalLists;

}

//conference articles

@SuppressWarnings("unchecked")

public ArrayList<ConferenceArticles> readConferenceData()

{

ArrayList<ConferenceArticles> conferenceLists = null;

try{

FileInputStream fileinput = new FileInputStream("src/files/conferenceArticles.dat");

ObjectInputStream objectinput = new ObjectInputStream(fileinput);

Object objR= objectinput.readObject();

conferenceLists = (ArrayList<ConferenceArticles>)objR;

objectinput.close();

}

catch (FileNotFoundException e){e.printStackTrace();}

catch (IOException e){}

catch (ClassNotFoundException e){e.printStackTrace();}

return conferenceLists;

}

public ArrayList<IssueRecords> readHireDetails()

{

ArrayList<IssueRecords> hirelists = null;

try{

FileInputStream fileinput = new FileInputStream("src/files/issueRecords.dat");

ObjectInputStream objectinput = new ObjectInputStream(fileinput);

Object objR= objectinput.readObject();

hirelists = (ArrayList<IssueRecords>)objR;

objectinput.close();

}

catch (FileNotFoundException e){e.printStackTrace();}

catch (IOException e){}

catch (ClassNotFoundException e){e.printStackTrace();}

return hirelists;

}

}

package Model;

import java.util.ArrayList;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import Serializable.BookChapters;

import Serializable.ConferenceArticles;

import Serializable.IssueRecords;

import Serializable.JournalArticles;

import Serializable.Researchers;

import Login.\*;

public class ResearcherModel {

private DefaultTableModel bookTableView = new DefaultTableModel(0, 9);

private DefaultTableModel journalTableView = new DefaultTableModel(0, 8);

private DefaultTableModel conferenceTableView = new DefaultTableModel(0, 7);

private DefaultTableModel hireTableView = new DefaultTableModel(0, 2);

private JTable bookTable=new JTable(bookTableView);

private JTable journalTable= new JTable(journalTableView);

private JTable conferencetable= new JTable(conferenceTableView);

private JTable hiretable= new JTable(hireTableView);

private JScrollPane bScrollPane = new JScrollPane(bookTable);

private JScrollPane jScrollPane = new JScrollPane(journalTable);

private JScrollPane cScrollPane = new JScrollPane(conferencetable);

private JScrollPane hScrollPane = new JScrollPane(hiretable);

//array

private ArrayList <Researchers> researcherArray = new ArrayList<Researchers>();

private ArrayList <BookChapters> bookChapterArray = new ArrayList<BookChapters>();

private ArrayList <JournalArticles> journalArray = new ArrayList<JournalArticles>();

private ArrayList <ConferenceArticles> conferenceArray = new ArrayList<ConferenceArticles>();

private ArrayList <IssueRecords> issueArray = new ArrayList<IssueRecords>();

BookChapters bookModel;

JournalArticles journalModel;

ConferenceArticles conferenceModel;

IssueRecords issueModel;

DataReader reader = new DataReader();

public ResearcherModel()

{

showBookChapterTable();

showJournalArticles();

showConferenceArticles();

}

//for viewing specific reseracher's hires

public ResearcherModel(String a) { this.a = a; showHires(); }

//bookchapters

public void showBookChapterTable() {

if(reader.readBookData()!=null) {

boolean found =false;

bookChapterArray = reader.readBookData();

BookChapters reObj = new BookChapters();

for(int i1=0; i1<bookChapterArray.size(); i1++)

{

if(reader.readHireDetails()!=null)

{

issueArray = reader.readHireDetails();

IssueRecords check = new IssueRecords();

for(int i=0; i<issueArray.size(); i++)

{

reObj=bookChapterArray.get(i1);

check=issueArray.get(i);

if(reObj.getId().equals(check.getAid()))

{ found = true; break;}

else {found = false;}

}}

if(found==false) {

reObj=bookChapterArray.get(i1);

Object[] data= {reObj.getId(), reObj.getTitle(), reObj.getAuthor(), reObj.getYear(), reObj.getBookEditor(), reObj.getPublicationPlace(), reObj.getPublisher(), reObj.getPage()};

bookTableView.addRow(data);

}}}

Object tableHead[]= {"ID","Title","Author(s)","Publication Year","Book Editor","Place of Publication","Publisher","Page"};

bookTableView.setColumnIdentifiers(tableHead);

}

// journalArticles

public void showJournalArticles() {

if(reader.readJournalData()!=null) {

boolean found =false;

journalArray = reader.readJournalData();

JournalArticles reObj = new JournalArticles();

for(int i1=0; i1<journalArray.size(); i1++)

{

if(reader.readHireDetails()!=null)

{

issueArray = reader.readHireDetails();

IssueRecords check = new IssueRecords();

for(int i=0; i<issueArray.size(); i++)

{

reObj=journalArray.get(i1);

check=issueArray.get(i);

if(!reObj.getId().equals(check.getAid()))

{ found = false;

} else {found = true; break;}

}

}

if(found==false)

{

reObj=journalArray.get(i1);

Object[] data= {reObj.getId(), reObj.getTitle(), reObj.getJournalTitle(), reObj.getYear(),reObj.getVolume(),reObj.getIssue(),reObj.getAuthor(),reObj.getPage()};

journalTableView.addRow(data);

}}}

Object tableHead[]= {"ID","Title","Journal Title","Publication Year","Volume","Issue No.","Author(s)","Page"};

journalTableView.setColumnIdentifiers(tableHead);

}

//conference Article

public void showConferenceArticles()

{

if(reader.readConferenceData()!=null)

{

boolean found = false;

conferenceArray = reader.readConferenceData();

ConferenceArticles reObj = new ConferenceArticles();

for(int i1=0; i1<conferenceArray.size(); i1++)

{

if(reader.readHireDetails()!=null)

{

issueArray = reader.readHireDetails();

IssueRecords check = new IssueRecords();

for(int i=0; i<issueArray.size(); i++)

{

reObj=conferenceArray.get(i1);

check=issueArray.get(i);

if(!reObj.getId().equals(check.getAid()))

{ found = false;}

else

{found = true;

break;}

}}

if(found==false)

{

reObj=conferenceArray.get(i1);

Object[] data= {reObj.getId(), reObj.getTitle(), reObj.getAuthor(), reObj.getYear(), reObj.getBookEditor(), reObj.getPublicationPlace(), reObj.getPage()};

conferenceTableView.addRow(data);

}}

}

Object tableHead[]= {"ID","Title","Author(s)","Publication Year","Editor(s)","Conference Location","Page"};

conferenceTableView.setColumnIdentifiers(tableHead);}

//articles

public void showHires() {

if(reader.readHireDetails()!=null)

{

issueArray = reader.readHireDetails();

IssueRecords reObj = new IssueRecords();

for(int i=0; i<issueArray.size(); i++)

{

reObj=issueArray.get(i);

if(reObj.getRid().equals(getResearcherUsernamefromHire())) {

Object[] data= {reObj.getAid(), reObj.getArticleName()};

hireTableView.addRow(data);

}

}}

Object tableHead[]= {"Article No.","Title"};

hireTableView.setColumnIdentifiers(tableHead);

}

//accessors

public JScrollPane getbScrollPane() { return bScrollPane; }

public JScrollPane getjScrollPane() { return jScrollPane; }

public JScrollPane getcScrollPane() { return cScrollPane; }

public JScrollPane gethScrollPane() { return hScrollPane; }

public JTable getBookTable() { return bookTable; }

public JTable getJournalTable() { return journalTable; }

public JTable getConferenceTable() { return conferencetable; }

public JTable getHireTable() { return hiretable; }

public String getResearcherUsernamefromHire() {

String userid="";

if(reader.readData()!=null) {

researcherArray= reader.readData();

Researchers reObj = new Researchers();

for(int i=0; i<researcherArray.size(); i++)

{

reObj=researcherArray.get(i);

if(reObj.getUserName().equals(getTrying())) {

userid= reObj.getId();

break;

}}}

return userid;

}

private String a;

public void trying(String a) { a= this.a;}

public String getTrying() {return this.a;}

}

package Serializable;

import java.io.Serializable;

import java.util.ArrayList;

public class BookChapters implements Serializable{

private static final long serialVersionUID = 1L;

private String ID, title, author,publicationYear, bookEditor, place, publisher, page;

//constructor

public BookChapters() {}

//parameterized

public BookChapters(String ID, String title, String author, String publicationYear, String bookEditor, String place, String publisher, String page)

{

this.ID= ID;

this.title=title;

this.author= author;

this.publicationYear= publicationYear;

this.bookEditor= bookEditor;

this.place= place;

this.publisher = publisher;

this.page = page;

}

//accessors

public String getId() { return this.ID; }

public String getTitle() { return this.title; }

public String getAuthor() { return this.author; }

public String getBookEditor() { return this.bookEditor; }

public String getPublicationPlace() { return this.place; }

public String getYear() { return this.publicationYear; }

public String getPublisher() { return this.publisher; }

public String getPage() { return this.page; }

} **package** Serializable;

**import** java.io.Serializable;

**public** **class** ConferenceArticles **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** String ID, title, author,publicationYear, bookEditor, place, page;

//constructor

**public** ConferenceArticles() {}

//parameterized

**public** ConferenceArticles(String ID, String title, String author, String publicationYear, String bookEditor, String place, String page)

{

**this**.ID= ID;

**this**.title=title;

**this**.author= author;

**this**.publicationYear= publicationYear;

**this**.bookEditor= bookEditor;

**this**.place= place;

**this**.page = page;

}

//accessors

**public** String getId() { **return** **this**.ID; }

**public** String getTitle() { **return** **this**.title; }

**public** String getAuthor() { **return** **this**.author; }

**public** String getBookEditor() { **return** **this**.bookEditor; }

**public** String getPublicationPlace() { **return** **this**.place; }

**public** String getYear() { **return** **this**.publicationYear; }

**public** String getPage() { **return** **this**.page; }

}

**package** Serializable;

**import** java.io.Serializable;

/\*

\* This records the hire out article's id and name and researcher's id

\*/

**public** **class** IssueRecords **implements** Serializable {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** String rid, aid, articlename;

**public** IssueRecords(String r, String a, String article)

{

rid= r;

aid= a;

articlename= article;

}

**public** IssueRecords() {}

**public** String getRid() { **return** rid;}

**public** String getAid() { **return** aid;}

**public** String getArticleName() { **return** articlename;}

}

**package** Serializable;

**import** java.io.Serializable;

**public** **class** JournalArticles **implements** Serializable{

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**private** String ID, title, journalTitle,publicationYear, volume, issue, author, page;

//constructor

**public** JournalArticles() {}

//parameterized

**public** JournalArticles(String ID, String title, String jtitle, String publicationYear, String volume, String issue, String author, String page)

{

**this**.ID= ID;

**this**.title=title;

**this**.author= author;

**this**.publicationYear= publicationYear;

**this**.volume= volume;

**this**.issue= issue;

**this**.journalTitle= jtitle;

**this**.page = page;

}

//accessors

**public** String getId() { **return** **this**.ID; }

**public** String getTitle() { **return** **this**.title; }

**public** String getAuthor() { **return** **this**.author; }

**public** String getVolume() { **return** **this**.volume; }

**public** String getIssue() { **return** **this**.issue; }

**public** String getYear() { **return** **this**.publicationYear; }

**public** String getJournalTitle() { **return** **this**.journalTitle; }

**public** String getPage() { **return** **this**.page; }

}

package Serializable;

import java.io.Serializable;

import java.util.ArrayList;

public class Researchers implements Serializable {

private static final long serialVersionUID = 1L;

private String ID, name, address,contact, email, username, password;

//constructor

public Researchers() {}

//parameterized

public Researchers(String ID, String name, String address, String contact, String email, String username, String password)

{

this.ID= ID;

this.name=name;

this.address= address;

this.contact= contact;

this.email= email;

this.username= username;

this.password = password;

}

//accessors

public String getId() { return this.ID; }

public String getName() { return this.name; }

public String getAddress() { return this.address; }

public String getEmail() { return this.email; }

public String getContact() { return this.contact; }

public String getUserName() { return this.username; }

public String getPassword() { return this.password; }

}