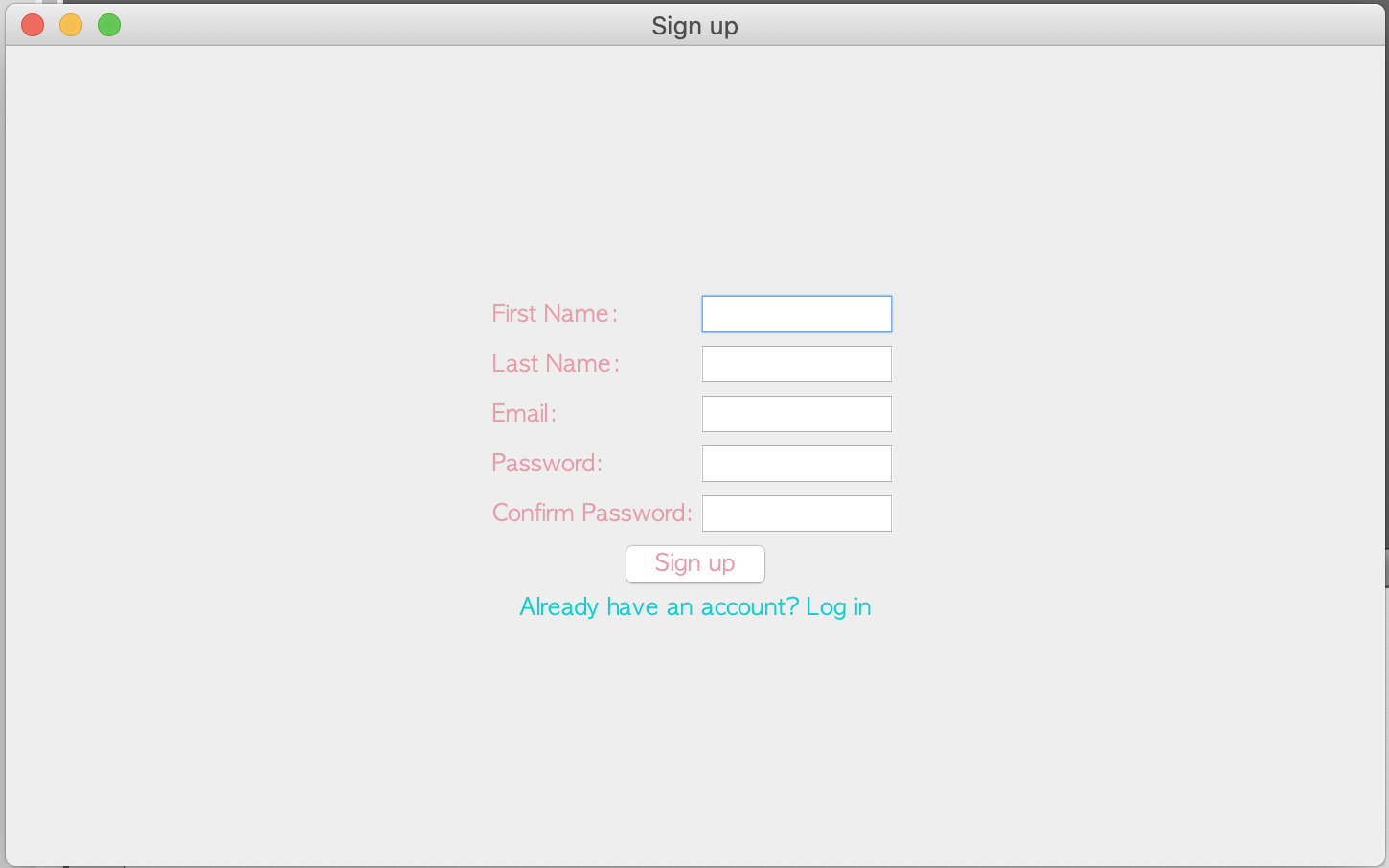
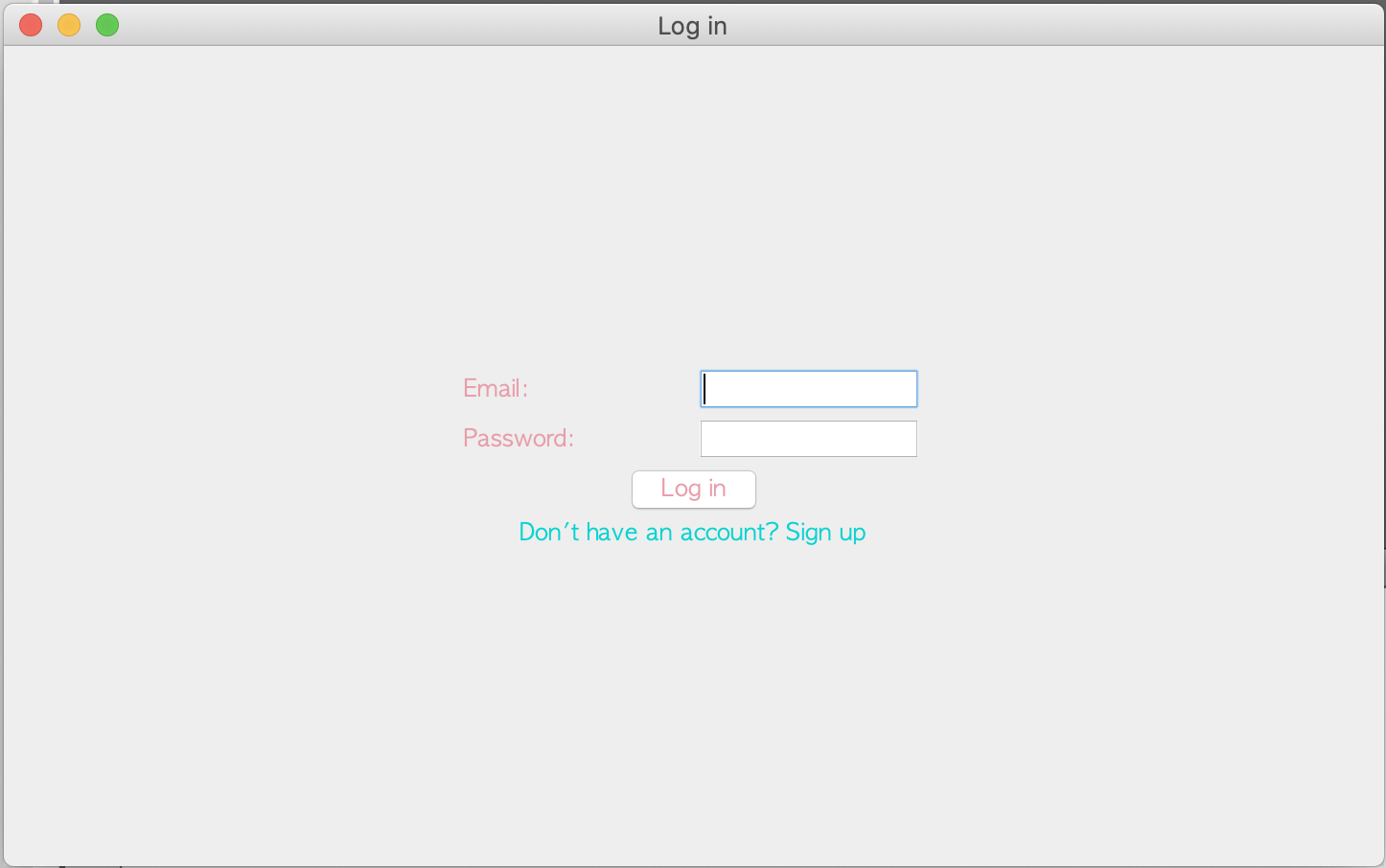
# Book Borrowing System

Our advanced programming project is a book borrowing system. Participating students are Reem Suliman 437051717 and Marihan Abdulaziz 437051727.

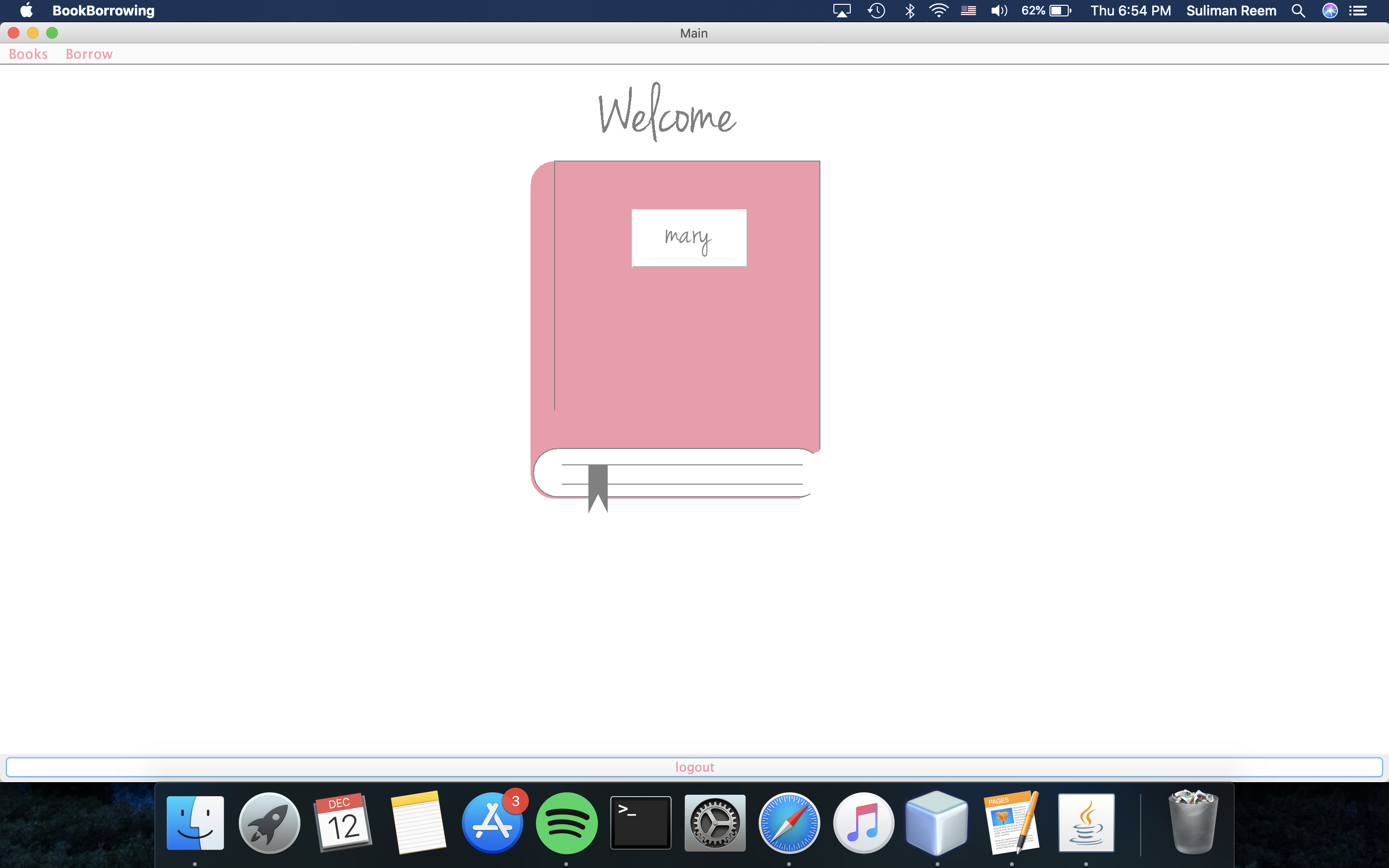
## Sign Up

When the program is run, the sign up page pops up:

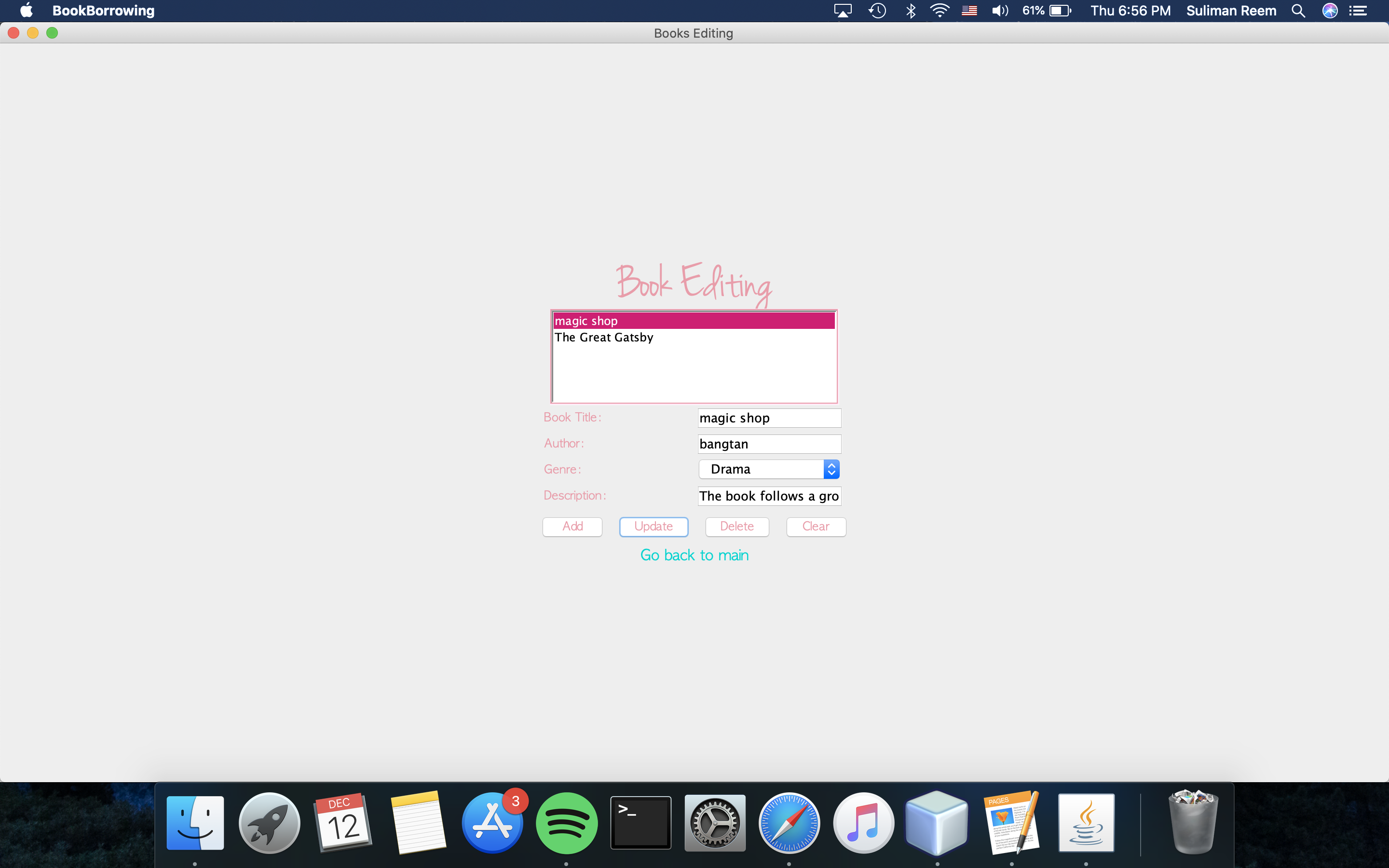
## Log In

The user can be directed via the sign up page to the login page: 

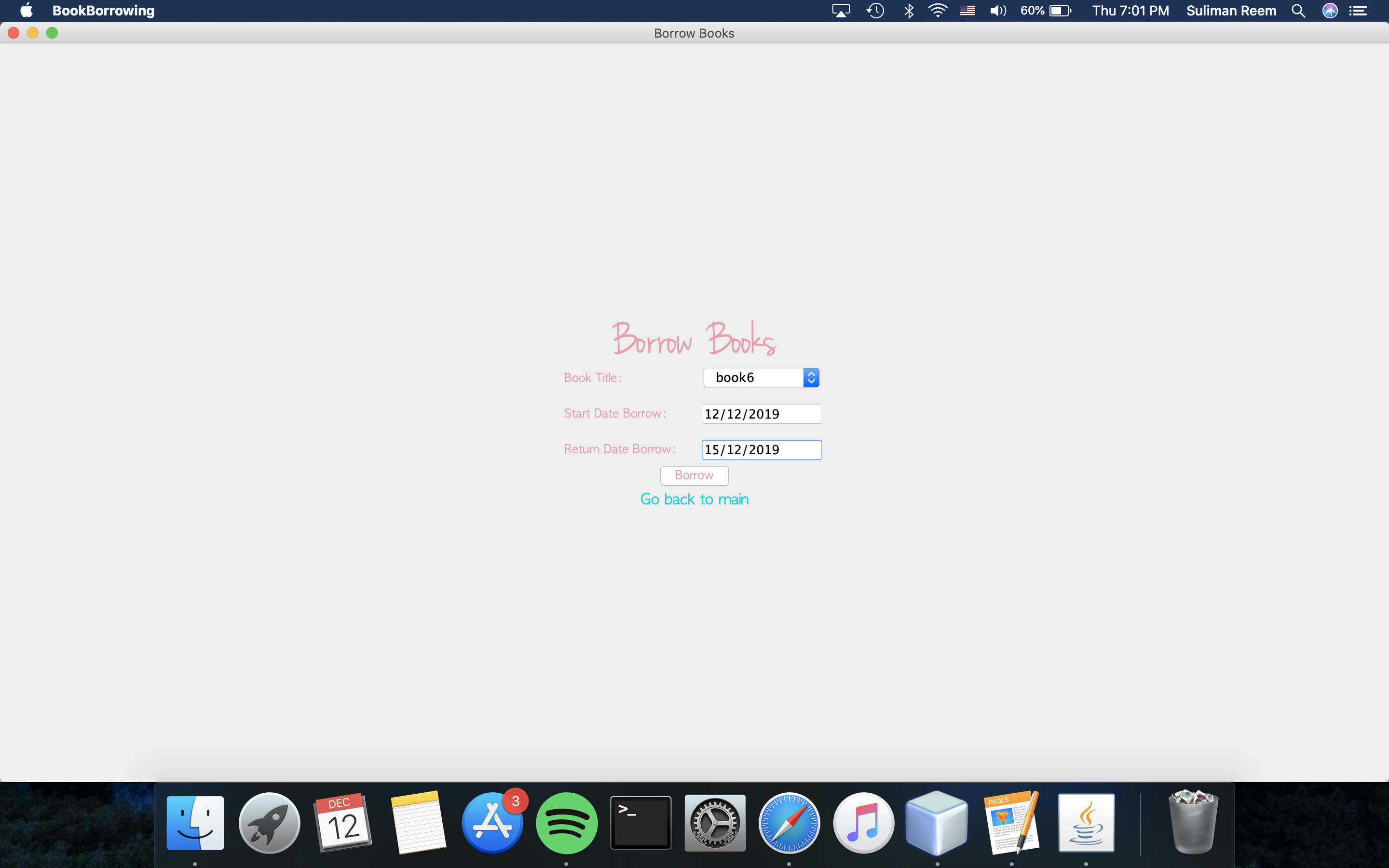
## Main

After logging in, the user will be directed to the main page, which has a personalized welcoming message, the user can navigate the program via the main page using the menu bar:

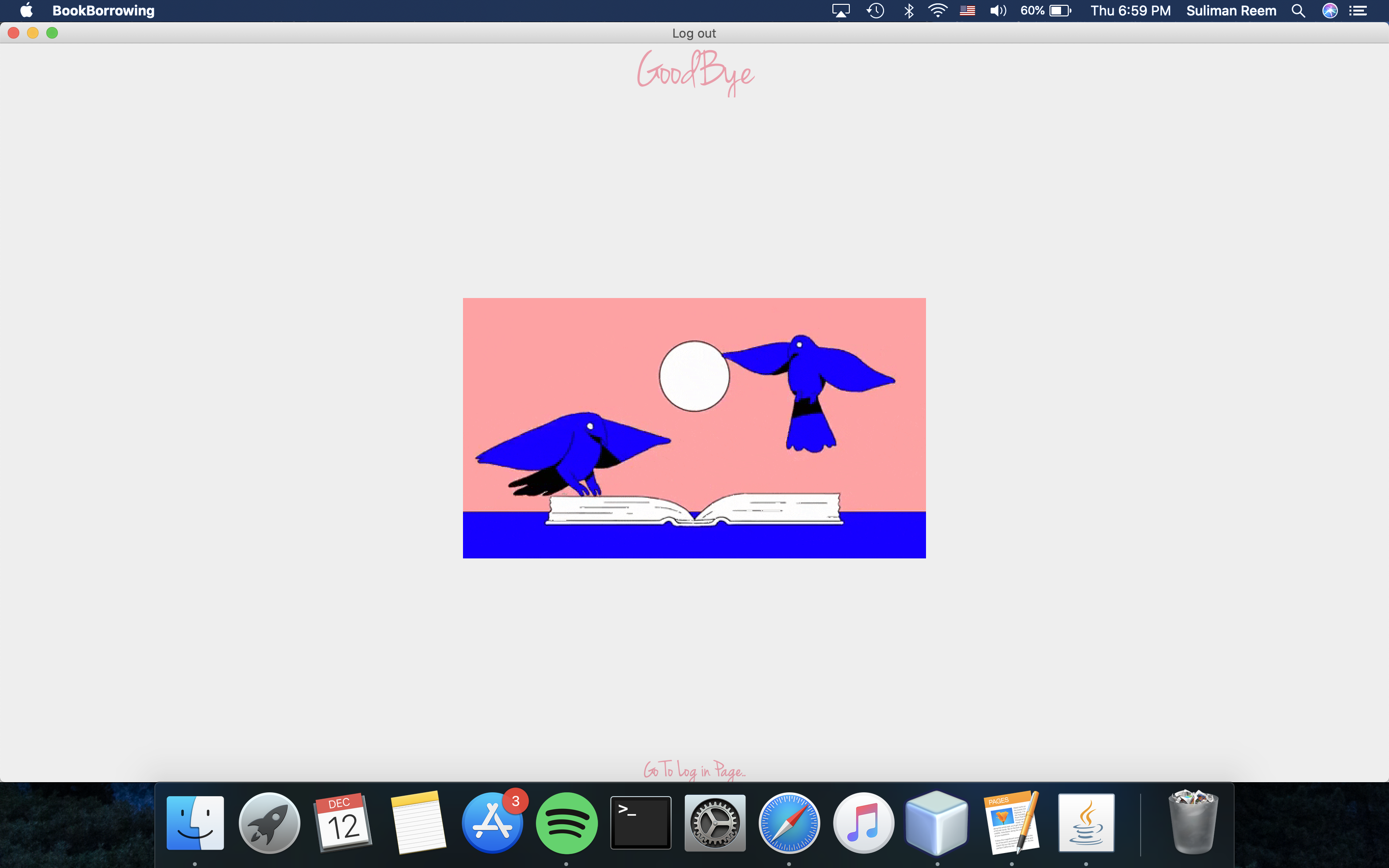
## Books Editing

A user now can add, update, delete their own books for people to borrow via the books edit, once a book is added, its status in the database is changed to available:

## Books Borrowing

A user can also borrow a book by using the borrow page, the users own books don’t show up in the book title combo box, and once a book is borrowed, its status in the database changes to unavailable, and thus, it doesn’t show up either, the user’s ID gets added to the database as the borrowing user:

## Log out

A user can logout after finishing with the program:

## Database

The database is as follows:

CREATE TABLE users (

u\_id int(11) AUTO\_INCREMENT PRIMARY KEY,

f\_name varchar(20)NOT NULL,

l\_name varchar(20)NOT NULL,

password varchar(256) NOT NULL,

email varchar(30) UNIQUE NOT NULL

);

CREATE TABLE books (

b\_id int(11) AUTO\_INCREMENT PRIMARY KEY,

b\_title varchar(50) UNIQUE NOT NULL,

auther varchar(50) NOT NULL,

description varchar(200) NOT NULL,

genre varchar(50) NOT NULL,

status varchar(11) NOT NULL DEFAULT 'Available',

uid int(11)

);

CREATE TABLE bookBorrowing (

bb\_id int(11) AUTO\_INCREMENT PRIMARY KEY,

u\_id int(11) NOT NULL,

b\_title varchar(50) NOT NULL,

date\_b varchar(10) NOT NULL,

date\_r varchar(10) NOT NULL

);

ALTER TABLE bookBorrowing

ADD CONSTRAINT fk\_uid

FOREIGN KEY (u\_id) REFERENCES users (u\_id) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE bookBorrowing

ADD CONSTRAINT fk\_btitle

FOREIGN KEY (b\_title) REFERENCES books (b\_title) ON DELETE CASCADE ON UPDATE CASCADE;

ALTER TABLE books

ADD CONSTRAINT fk\_u\_id

FOREIGN KEY (uid) REFERENCES users (u\_id) ON DELETE CASCADE ON UPDATE CASCADE;

## Code

//main class

package bookborrowing;

import java.awt.\*;

import javax.swing.JFrame;

public class BookBorrowing {

public static void main(String[] args) {

SignUp sp=new SignUp();

sp.setVisible(true);

}

}

////DB Connection

package bookborrowing;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

public class DBConnection {

private static Connection conn=null;

private static final String USERNAME="root";

private static final String passwordDB="";

private static final String conString="jdbc:mysql://localhost:3306/progProj";

public static Connection getConnection(){

try{

conn=DriverManager.getConnection(conString,USERNAME,passwordDB);

} catch (SQLException ex) {

System.out.print(ex.getMessage());

}

return conn;

}

}

//Login class

package bookborrowing;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.SQLIntegrityConstraintViolationException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

public class Login extends JFrame {

private JLabel email, password, msg;

private JPasswordField passwordF;

private JTextField emailF;

private JButton login;

PreparedStatement ps;

ResultSet rs;

private static String sqlQ;

String username="";

int userid=0;

public Login(){

super("Log in");

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Toolkit kit = Toolkit.getDefaultToolkit();

setLocation(kit.getScreenSize().width/4, kit.getScreenSize().height/4);

setSize(kit.getScreenSize().width/2, kit.getScreenSize().height/2);

Color lyellow = new Color(232,157,170);

Color aqua = new Color(11,211,208);

setLayout(new GridBagLayout());

JPanel logInPanel=new JPanel();

logInPanel.setLayout(new GridLayout(2,2));

GridBagConstraints gbc = new GridBagConstraints();

email=new JLabel("Email:");

password=new JLabel("Password:");

email.setForeground(lyellow);

password.setForeground(lyellow);

email.setFont(new Font("AppleGothic", Font.BOLD,12));

password.setFont(new Font("AppleGothic", Font.BOLD,12));

emailF=new JTextField();

passwordF=new JPasswordField();

passwordF.setPreferredSize(new Dimension(120,20));

logInPanel.add(email);

logInPanel.add(emailF);

logInPanel.add(password);

logInPanel.add(passwordF);

login=new JButton("Log in");

login.setForeground(lyellow);

msg=new JLabel("Don't have an account? Sign up");

msg.setForeground(aqua);

msg.setFont(new Font("AppleGothic", Font.BOLD,12));

login.setFont(new Font("AppleGothic", Font.BOLD,12));

gbc.gridx = 200;

gbc.gridy = 50;

add(logInPanel, gbc);

gbc.gridx = 200;

gbc.gridy = 170;

add(login, gbc);

gbc.gridx = 200;

gbc.gridy = 200;

add(msg, gbc);

login.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(emailF.getText().equals("") || String.valueOf(passwordF.getPassword()).equals("") ){

JOptionPane.showMessageDialog(rootPane, "All fields are required", "error message", JOptionPane.ERROR\_MESSAGE);

} else{

try {

String email=emailF.getText();

String password=String.valueOf(passwordF.getPassword());

sqlQ="SELECT \* FROM `users` WHERE `email`=? AND `password`=? ";

ps= DBConnection.getConnection().prepareStatement(sqlQ);

ps.setString(1, email);

ps.setString(2, password);

rs=ps.executeQuery();

if(rs.next()){

username=rs.getString("f\_name");

userid=rs.getInt("u\_id");

setVisible(false);

Main m=new Main(username, userid);

m.setVisible(true);

} else{

JOptionPane.showMessageDialog(rootPane, "Error loggin in, please try again", "error message", JOptionPane.ERROR\_MESSAGE);

}

} catch (SQLException ex) {

if (ex instanceof SQLIntegrityConstraintViolationException) {

JOptionPane.showMessageDialog(rootPane, "The email entered has already registerd", "error message", JOptionPane.ERROR\_MESSAGE);

} else{

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}

}}

});

msg.addMouseListener(new MouseAdapter(){

public void mouseClicked(MouseEvent e){

SignUp sp=new SignUp();

setVisible(false);

sp.setVisible(true);

}

});

}

}

//Sign up class

package bookborrowing;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.SQLIntegrityConstraintViolationException;

import java.sql.Statement;

import javax.swing.\*;

public class SignUp extends JFrame {

private JLabel firstName, lastName, email, password, conPassword, msg;

private JPasswordField passwordF, conPasswordF;

private JTextField firstNameF, lastNameF, emailF;

private JButton SignUp;

private static String sqlQ;

PreparedStatement ps;

public SignUp(){

super("Sign up");

Toolkit kit = Toolkit.getDefaultToolkit();

setLocation(kit.getScreenSize().width/4, kit.getScreenSize().height/4);

setSize(kit.getScreenSize().width/2, kit.getScreenSize().height/2); //1440, 900

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new GridBagLayout());

JPanel signUpPanel=new JPanel();

signUpPanel.setLayout(new GridLayout(5,2));

GridBagConstraints gbc = new GridBagConstraints();

Color lyellow = new Color(232,157,170);

Color aqua = new Color(11,211,208);

firstName=new JLabel("First Name:");

lastName=new JLabel("Last Name:");

email=new JLabel("Email:");

password=new JLabel("Password:");

conPassword=new JLabel("Confirm Password:");

firstName.setForeground(lyellow);

lastName.setForeground(lyellow);

email.setForeground(lyellow);

password.setForeground(lyellow);

conPassword.setForeground(lyellow);

firstName.setFont(new Font("AppleGothic", Font.BOLD,12));

lastName.setFont(new Font("AppleGothic", Font.BOLD,12));

email.setFont(new Font("AppleGothic", Font.BOLD,12));

password.setFont(new Font("AppleGothic", Font.BOLD,12));

conPassword.setFont(new Font("AppleGothic", Font.BOLD,12));

passwordF=new JPasswordField();

conPasswordF=new JPasswordField();

firstNameF=new JTextField();

lastNameF=new JTextField();

emailF=new JTextField();

signUpPanel.add(firstName);

signUpPanel.add(firstNameF);

signUpPanel.add(lastName);

signUpPanel.add(lastNameF);

signUpPanel.add(email);

signUpPanel.add(emailF);

signUpPanel.add(password);

signUpPanel.add(passwordF);

signUpPanel.add(conPassword);

signUpPanel.add(conPasswordF);

gbc.gridx = 200;

gbc.gridy = 50;

add(signUpPanel, gbc);

SignUp=new JButton("Sign up");

SignUp.setForeground(lyellow);

msg=new JLabel("Already have an account? Log in");

msg.setForeground(aqua);

msg.setFont(new Font("AppleGothic", Font.BOLD,12));

SignUp.setFont(new Font("AppleGothic", Font.BOLD,12));

gbc.gridx = 200;

gbc.gridy = 170;

add(SignUp, gbc);

gbc.gridx = 200;

gbc.gridy = 200;

add(msg, gbc);

SignUp.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(firstNameF.getText().equals("") || lastNameF.getText().equals("") || emailF.getText().equals("") || String.valueOf(passwordF.getPassword()).equals("") || String.valueOf(conPasswordF.getPassword()).equals("") ){

JOptionPane.showMessageDialog(rootPane, "All fields are required", "error message", JOptionPane.ERROR\_MESSAGE);

}

else {

if(!String.valueOf(passwordF.getPassword()).equals(String.valueOf(conPasswordF.getPassword()))){

JOptionPane.showMessageDialog(rootPane, "The passwords entered do not match up, please enter password again", "error message", JOptionPane.ERROR\_MESSAGE);

} else{

try{

String email=emailF.getText();

String password= String.valueOf(passwordF.getPassword());

String fName=firstNameF.getText();

String lName=lastNameF.getText();

sqlQ="INSERT INTO `users`(`f\_name`, `l\_name`, `password`, `email`) VALUES (?,?,?,?)";

ps= DBConnection.getConnection().prepareStatement(sqlQ);

ps.setString(1, fName);

ps.setString(2, lName);

ps.setString(3, password);

ps.setString(4, email);

if(ps.executeUpdate()>0){

JOptionPane.showMessageDialog(rootPane, "Sign up sucessfully!");

}

} catch (SQLException ex) {

if (ex instanceof SQLIntegrityConstraintViolationException) {

JOptionPane.showMessageDialog(rootPane, "The email entered has already registerd", "error message", JOptionPane.ERROR\_MESSAGE);

} else{

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}

}

}

}

});

msg.addMouseListener(new MouseAdapter(){

public void mouseClicked(MouseEvent e){

Login li=new Login();

setVisible(false);

li.setVisible(true);

}

});

}

}

//main class

package bookborrowing;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

public class Main extends JFrame {

draw d;

int userid;

String username;

JMenuBar bar;

JMenu Menu1,Menu2;

JMenuItem i2, i3;

JButton b1;

public Main(){

super("Main");

}

public Main(final String username, final int userid){

super("Main");

this.username=username;

setExtendedState(JFrame.MAXIMIZED\_BOTH);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

d=new draw();

d.setString(""+username);

Menu1=new JMenu("Books");

Menu2=new JMenu("Borrow");

Color c=new Color(232,157,170);

i2=new JMenuItem("Edit Books");

i3=new JMenuItem("Borrow Books");

i2.setForeground(c);

i3.setForeground(c);

bar=new JMenuBar();

bar.setForeground(c);

bar.setOpaque(true);

Menu1.add(i2);

Menu2.add(i3);

setJMenuBar(bar);

bar.add(Menu1);

bar.add(Menu2);

add(d,BorderLayout.CENTER);

b1=new JButton("logout");

b1.setForeground(c);

add(b1,BorderLayout.SOUTH);

i2.addActionListener( new ActionListener() {

public void actionPerformed(ActionEvent e) {

//EditPage

booksEditing be=new booksEditing(userid,username);

setVisible(false);

be.setVisible(true);

}

} );

i3.addActionListener( new ActionListener() {

public void actionPerformed(ActionEvent e) {

borrowpage b=new borrowpage(userid,username);

setVisible(false);

b.setVisible(true);

}

} );

b1.addActionListener( new ActionListener() {

public void actionPerformed(ActionEvent e) {

//logout

setVisible(false);

logoutpage l=new logoutpage();

l.setVisible(true);

}

} );

}

}

//Books Editing class

package bookborrowing;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.SQLIntegrityConstraintViolationException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

import javax.swing.border.Border;

import javax.swing.event.ListSelectionEvent;

import javax.swing.event.ListSelectionListener;

public class booksEditing extends JFrame {

private JLabel bookTitle, author, description, genre;

private JTextField bookTitleF, autherF, descriptionF;

private JComboBox genreB;

private JButton insert, update, delete, clear;

private JLabel main, title;

JList<String> list;

DefaultListModel<String> DLM;

private static String sqlQ;

PreparedStatement ps;

ResultSet rs;

Border raisedbevel,loweredbevel;

int userid;

String usrename;

Main m;

Login log;

public booksEditing(final int userid, final String username){

super("Books Editing");

this.userid=userid;

this.usrename=username;

setLayout(new GridBagLayout());

setExtendedState(JFrame.MAXIMIZED\_BOTH);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setBackground(Color.white);

JPanel booksEditingPanel=new JPanel();

booksEditingPanel.setLayout(new GridLayout(4,2));

GridBagConstraints gbc = new GridBagConstraints();

Color lyellow = new Color(232,157,170);

Color aqua = new Color(11,211,208);

list = new JList<>();

Border compound;

raisedbevel = BorderFactory.createBevelBorder(1, lyellow, lyellow);

loweredbevel = BorderFactory.createLoweredBevelBorder();

compound = BorderFactory.createCompoundBorder(raisedbevel, loweredbevel);

list.setBorder(compound);

DLM = new DefaultListModel<>();

list.setPreferredSize(new Dimension(300, 100));

fillList();

title=new JLabel("Book Editing");

bookTitle=new JLabel("Book Title: ");

author=new JLabel("Author: ");

genre=new JLabel("Genre: ");

description=new JLabel("Description: ");

title.setForeground(lyellow);

title.setFont(new Font("Jenna Sue", Font.PLAIN,50));

bookTitleF=new JTextField();

autherF=new JTextField();

descriptionF=new JTextField();

genreB=new JComboBox();

genreB.addItem("Drama");

genreB.addItem("Action");

genreB.addItem("adventure");

genreB.addItem("Fantasy");

genreB.addItem("Mystery");

genreB.addItem("Horror");

genreB.addItem("Coming-of-age");

genreB.addItem("Comic book");

genreB.addItem("Romance");

genreB.addItem("Poetry");

genreB.addItem("Science fiction");

genreB.addItem("Crime");

bookTitle.setForeground(lyellow);

author.setForeground(lyellow);

genre.setForeground(lyellow);

description.setForeground(lyellow);

bookTitle.setFont(new Font("AppleGothic", Font.BOLD,12));

author.setFont(new Font("AppleGothic", Font.BOLD,12));

genre.setFont(new Font("AppleGothic", Font.BOLD,12));

description.setFont(new Font("AppleGothic", Font.BOLD,12));

booksEditingPanel.add(bookTitle);

booksEditingPanel.add(bookTitleF);

booksEditingPanel.add(author);

booksEditingPanel.add(autherF);

booksEditingPanel.add(genre);

booksEditingPanel.add(genreB);

booksEditingPanel.add(description);

booksEditingPanel.add(descriptionF);

insert=new JButton("Add");

update=new JButton("Update");

delete=new JButton("Delete");

clear=new JButton("Clear");

insert.setForeground(lyellow);

update.setForeground(lyellow);

delete.setForeground(lyellow);

clear.setForeground(lyellow);

insert.setFont(new Font("AppleGothic", Font.PLAIN,12));

update.setFont(new Font("AppleGothic", Font.PLAIN,12));

delete.setFont(new Font("AppleGothic", Font.PLAIN,12));

clear.setFont(new Font("AppleGothic", Font.PLAIN,12));

JPanel buttonsPanel=new JPanel();

buttonsPanel.setLayout(new FlowLayout());

buttonsPanel.add(insert);

buttonsPanel.add(update);

buttonsPanel.add(delete);

buttonsPanel.add(clear);

main=new JLabel("Go back to main");

main.setForeground(aqua);

main.setFont(new Font("AppleGothic", Font.PLAIN,15));

gbc.gridx = 200;

gbc.gridy = 50;

add(title,gbc);

gbc.gridx = 200;

gbc.gridy = 100;

add(list , gbc);

gbc.gridx = 200;

gbc.gridy = 200;

add(booksEditingPanel,gbc);

gbc.gridx = 200;

gbc.gridy = 250;

add(buttonsPanel,gbc);

gbc.gridx = 200;

gbc.gridy = 300;

add(main,gbc);

main.addMouseListener(new MouseAdapter(){

public void mouseClicked(MouseEvent e){

m=new Main(username, userid);

setVisible(false);

m.setVisible(true);

}

});

list.addMouseListener(new MouseAdapter(){

@Override

public void mouseClicked(MouseEvent e){

try{

String tmp=(String) list.getSelectedValue();

String sqlQ1="SELECT `auther`,`description`,`genre` FROM `books` WHERE uid=? AND status=? AND `b\_title`=? ";

ps= DBConnection.getConnection().prepareStatement(sqlQ1);

ps.setInt(1, userid);

ps.setString(2, "Available");

ps.setString(3, tmp);

rs=ps.executeQuery();

if(rs.next()){

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

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

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

bookTitleF.setText(tmp);

autherF.setText(auther);

genreB.setSelectedItem(genre);

descriptionF.setText(description);

}

}catch(Exception ex){

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}

});

insert.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(bookTitleF.getText().equals("") || autherF.getText().equals("") || descriptionF.getText().equals("")){

JOptionPane.showMessageDialog(rootPane, "All fields are required", "error message", JOptionPane.ERROR\_MESSAGE);

}

else {

try{

String bookTitle=bookTitleF.getText();

String auther=autherF.getText();

String description=descriptionF.getText();

String genre=genreB.getSelectedItem().toString();

sqlQ="INSERT INTO books(b\_title, auther, description, genre, status, uid) VALUES (?,?,?,?,?,?)";

ps= DBConnection.getConnection().prepareStatement(sqlQ);

ps.setString(1, bookTitle);

ps.setString(2, auther);

ps.setString(3, description);

ps.setString(4, genre);

ps.setString(5, "Available");

ps.setInt(6, userid);

if(ps.executeUpdate()>0){

DLM.addElement(bookTitle);

JOptionPane.showMessageDialog(rootPane, "Book added sucessfully!");

}

} catch(SQLException ex){

if (ex instanceof SQLIntegrityConstraintViolationException) {

JOptionPane.showMessageDialog(rootPane,ex , "error message", JOptionPane.ERROR\_MESSAGE); //"The book entered has already been added"

} else{

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}finally{

try {

rs.close();

ps.close();

} catch (SQLException ex) {

Logger.getLogger(booksEditing.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

}});

clear.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

bookTitleF.setText("");

autherF.setText("");

genreB.setSelectedItem( genreB.getItemAt(0));

descriptionF.setText("");

list.clearSelection();

}

});

delete.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try {

int i=list.getSelectedIndex();

list.clearSelection();

String sqlQ1="DELETE FROM books WHERE b\_title=? ";

ps= DBConnection.getConnection().prepareStatement(sqlQ1);

ps.setString(1, bookTitleF.getText());

if(ps.executeUpdate()>0){

DLM.removeElementAt(i);

bookTitleF.setText("");

autherF.setText("");

genreB.setSelectedItem( genreB.getItemAt(0));

descriptionF.setText("");

JOptionPane.showMessageDialog(rootPane, "Record deleted sucessfully");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}

});

update.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

try {

String i=list.getSelectedValue(); //book choosen

int ii=list.getSelectedIndex();

String sqlQ1="SELECT b\_id FROM books WHERE b\_title=? ";

ps= DBConnection.getConnection().prepareStatement(sqlQ1);

ps.setString(1, i);

rs=ps.executeQuery();

if(rs.next()){

int b\_id=rs.getInt("b\_id");

String bookTitle=bookTitleF.getText();

String author=autherF.getText();

String genre= genreB.getSelectedItem().toString();

String description=descriptionF.getText();

String sqlQ12="UPDATE books SET b\_title=? ,auther=?,description=?,genre=? WHERE b\_id=? ";

ps= DBConnection.getConnection().prepareStatement(sqlQ12);

ps.setString(1, bookTitle);

ps.setString(2, author);

ps.setString(3, description);

ps.setString(4, genre);

ps.setInt(5, b\_id);

if(ps.executeUpdate()>0){

if(!DLM.getElementAt(ii).equals(bookTitle)){

DLM.removeElementAt(ii);

DLM.addElement(bookTitle);

}

JOptionPane.showMessageDialog(rootPane, "Record updated sucessfully");

}

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}

});

}

private void fillList(){

try{

String sqlQ1="SELECT b\_title, auther,description,genre FROM books WHERE uid=? AND status=? ";

ps= DBConnection.getConnection().prepareStatement(sqlQ1);

ps.setInt(1, userid);

ps.setString(2, "Available");

rs=ps.executeQuery();

while(rs.next()){

String bookTitle=rs.getString("b\_title");

DLM.addElement(bookTitle);

} list.setModel(DLM);

}catch(Exception e){

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

}

}

}

//Borrow Class

package bookborrowing;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

public class borrowpage extends JFrame{

JLabel title,l0,l1,l2,l3;

JTextField text1,text2,text3;

JButton b1;

JPanel p;

JComboBox box;

private static String sql0;

PreparedStatement ps;

ResultSet rs;

int userid;

String usrename;

Main m;

public borrowpage(final int userid, final String username) {

super("Borrow Books");

this.userid=userid;

this.usrename=username;

setLayout(new GridBagLayout());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setExtendedState(JFrame.MAXIMIZED\_BOTH);

setVisible(true);

GridBagConstraints gbc=new GridBagConstraints();

Color lyellow = new Color(232,157,170);

Color aqua = new Color(11,211,208);

title=new JLabel("Borrow Books", SwingConstants.CENTER);

title.setForeground(lyellow);

title.setFont(new Font("Jenna Sue", Font.PLAIN,50));

gbc.gridx=200;

gbc.gridy=0;

add(title,gbc);

l1=new JLabel("Book Title: ");

l2=new JLabel("Start Date Borrow: ");

l3=new JLabel("Return Date Borrow: ");

b1=new JButton("Borrow");

p=new JPanel();

l1.setFont(new Font("AppleGothic", Font.PLAIN,12));

l2.setFont(new Font("AppleGothic", Font.PLAIN,12));

l3.setFont(new Font("AppleGothic", Font.PLAIN,12));

l1.setForeground(lyellow);

l2.setForeground(lyellow);

l3.setForeground(lyellow);

b1.setForeground(lyellow);

b1.setFont(new Font("AppleGothic", Font.PLAIN,12));

box=new JComboBox();

text2=new JTextField(10);

text3=new JTextField(10);

p.setLayout(new GridLayout(3,2,10,10));

p.add(l1);

p.add(box);

p.add(l2);

p.add(text2);

p.add(l3);

p.add(text3);

gbc.gridx=200;

gbc.gridy=50;

add(p,gbc);

gbc.gridx=200;

gbc.gridy=200;

add(b1,gbc);

gbc.gridx=200;

gbc.gridy=550;

l0=new JLabel("Go back to main", SwingConstants.CENTER);

l0.setForeground(aqua);

l0.setFont(new Font("AppleGothic", Font.PLAIN,15));

add(l0,gbc);

try{

String sql1="select b\_title from books WHERE status=? AND uid <> ? ";

ps = DBConnection.getConnection().prepareStatement(sql1);

ps.setString(1, "Available");

ps.setInt(2, userid);

rs=ps.executeQuery();

while (rs.next()) {

box.addItem(rs.getString("b\_title"));

}

}catch(Exception e) {

JOptionPane.showMessageDialog(null,"Failed to Connect to Database","Error Connection", JOptionPane.WARNING\_MESSAGE);

}

l0.addMouseListener( new MouseAdapter()

{

public void mouseClicked( MouseEvent e )

{

int userid = 0;

setVisible(false);

m=new Main(username, userid);

m.setVisible(true);

}

});

b1.addActionListener( new ActionListener() {

public void actionPerformed(ActionEvent e) {

try {

String btitle=box.getSelectedItem().toString();

int selectedI=box.getSelectedIndex();

String sdate=text2.getText();

String rdate=text3.getText();

String sql01="INSERT INTO `bookBorrowing`( `u\_id`, `b\_title`, `date\_b`, `date\_r`) VALUES (?,?,?,?)";

ps=DBConnection.getConnection().prepareStatement(sql01);

ps.setInt(1, userid);

ps.setString(2, btitle);

ps.setString(3, sdate);

ps.setString(4, rdate);

if(ps.executeUpdate()>0){

String sql012="UPDATE books SET status=? WHERE b\_title=?";

ps=DBConnection.getConnection().prepareStatement(sql012);

ps.setString(1, "Unavailable");

ps.setString(2, btitle);

if(ps.executeUpdate()>0){

JOptionPane.showMessageDialog(null,"Borrow Sucessfully");

box.removeItemAt(selectedI);

box.setSelectedItem( box.getItemAt(0));

text2.setText("");

text3.setText("");

}

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(rootPane, ex.getMessage());

}

}} );

}

}

//draw class

package bookborrowing;

import javax.swing.\*;

import java.awt.\*;

public class draw extends JPanel{

String a;

Login log;

public draw(){

log=new Login();

}

public void paintComponent(Graphics g){

super.paintComponent(g);

setBackground(Color.WHITE);

Color c=new Color(232,157,170);

g.setColor(c);

//حجم الكتاب

g.fillRoundRect(550, 100,300, 350, 50, 50);

g.fillRoundRect(760, 100, 90, 55, 0, 0);

g.setColor(Color.WHITE);

g.fillRoundRect(553, 398,300, 50, 50, 50);

g.fillRoundRect(830, 400, 50, 50, 55, 55);

g.setColor(Color.GRAY);

//الخطين حق الصفحات

g.drawLine(583, 435, 850, 435);

g.drawLine(583, 415, 850, 415);

//حدود احافة الصفحات

g.drawRoundRect(553, 398,300, 50, 50, 50);

g.drawRoundRect(553, 398,300, 50, 50, 50);

g.setColor(Color.WHITE);

g.fillRoundRect(830, 400, 50, 50, 55, 55);

g.setColor(Color.MAGENTA);

//g.fillRoundRect(360, 50, 90, 55, 0, 0);

//بوكس النص

g.setColor(Color.WHITE);

g.fill3DRect(655,150, 120, 60, true);

g.setColor(Color.GRAY);

Font ff=new Font("Jenna Sue",Font.BOLD,75);

g.setFont(ff);

g.drawString("Welcome ",620,70);

Font f=new Font("Jenna Sue",Font.BOLD,40);

g.setFont(f);

g.drawString(a,690,185);

g.setColor(Color.GRAY);

//خطوط لتحديد الكتاب

g.drawLine(850, 100, 850, 398);

g.drawLine(575, 100, 850, 100);

g.drawLine(575, 100, 575, 358);

//رسم مؤشر الكتاب

g.fillRoundRect(610, 415,20, 50, 0, 0);

//رسم مثلث لتشكيل مؤشر الكتاب

g.setColor(Color.WHITE);

g.fillPolygon(new int[] {620, 610, 630}, new int[] {445, 465, 465}, 3);

}

public void setString(String a){

this.a=a;

}

}

//logout class

package bookborrowing;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import javax.swing.\*;

public class logoutpage extends JFrame{

JLabel l1,l2,l3;

public logoutpage(){

super("Log out");

setLayout(new BorderLayout());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setExtendedState(JFrame.MAXIMIZED\_BOTH);

setVisible(true);

Color lyellow = new Color(232,157,170);

l1=new JLabel("GoodBye", SwingConstants.CENTER);

l1.setFont(new Font("Jenna Sue",Font.BOLD,55));

l1.setForeground(lyellow);

ImageIcon pic=new ImageIcon("/Users/sulimanreem/NetBeansProjects/BookBorrowing/giphy2.gif");

l2=new JLabel(pic);

l3=new JLabel("Go To Log in Page..", SwingConstants.CENTER);

l3.setFont(new Font("Jenna Sue",Font.BOLD,24));

l3.setForeground(lyellow);

add(l1,BorderLayout.NORTH);

add(l2,BorderLayout.CENTER);

add(l3,BorderLayout.SOUTH);

l3.addMouseListener( new MouseAdapter()

{

public void mouseClicked( MouseEvent e )

{

setVisible(false);

Login l=new Login();

l.setVisible(true);

}

});

}

}