**PROJECT ON**

Library Management System

**Developed by**

Name:Cynthia Reg.No: 1713051001

Carole 1713051003

Teixeira 1713051010

Dorjane 1713051017

**NIIT**

**Library Management System**

Batch Code :Case Study 1

Start Date :June 24,2019

End Date :June 28,2019

Name of the Coordinator :Cynthia, Carole

Name of the Developer : Teixeira, Dorjane

Date of Submission :June 29,2019

**CERTIFICATE**

This is to certify this report titled Library Management System embodies the original work done by Cynthia, Carole in partial fulfillment of the course requirement at NIIT.

Coordinator:

Cynthia and Carole

**NIIT**

**ACKNOWLEDGEMENT**

In order to develop our programming skills and apply knowledge to practice,we are willing to do the project.We think it is necessary for us to design library management system.

Our team have benefited a lot from the feedback and suggestions given to us by Cynthia and Carole and other faculty members.We contribute a lot to the team.During the project,we overcame a lot of difficulties,we never give up when we have trouble in coding and designing UI.Cynthia and Carole spent a lot of time doing project,they think a lot and practice a lot.Without their help,we can’t finish the project.

At the beginning of designing a library management system,we don’t have a clear direction.According to the project book,we write ER diagram and flow chart.Then we search the Internet to see others how to operate the system.we use mysql-connector-java-8.0.16 to connect our database.There are three parts:membership、book and issue.We face some difficulties,for example,how to force the user to borrow at least 3 books a time and how to calculate the fine.The process is not easy but we finish it.

**Requirement Analysis**

In order to design the library management system,we use mysql.and then connect our database and java through mysql-connector-java-8.0.16.we set that only manager can log in the system.There are three parts of the system:book、membership and issue.

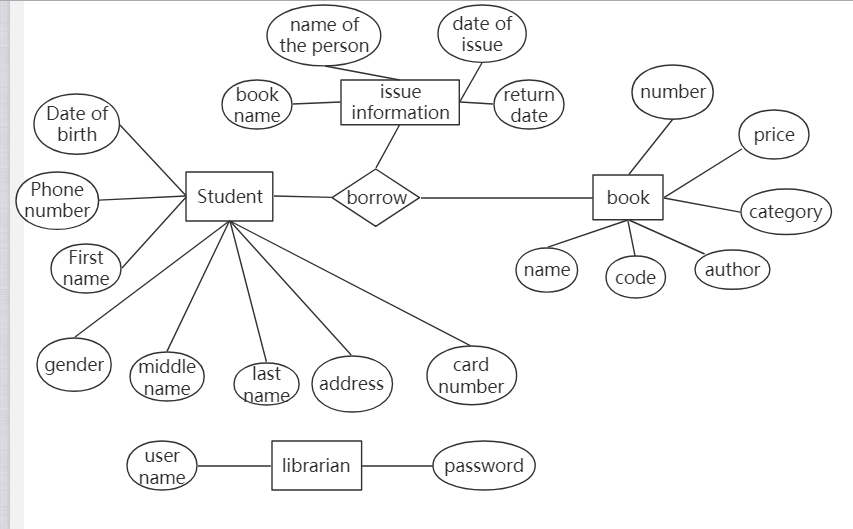
We also pay attention to the design of user interface,we spend a lot of time designing the UI to make it become beautiful.

In the book part,manager can add、delete and update book，we can also see the books stored in the library.In the membership part,we can add、delete and update manager’s information.we can also see personal information.In the issue part,we can borrow book return book and show personal information.There are two notes.First is that a person can only borrow at most 3 books a time.The second is that if the borrower return the book beyond the deadline,he will pay the fine.

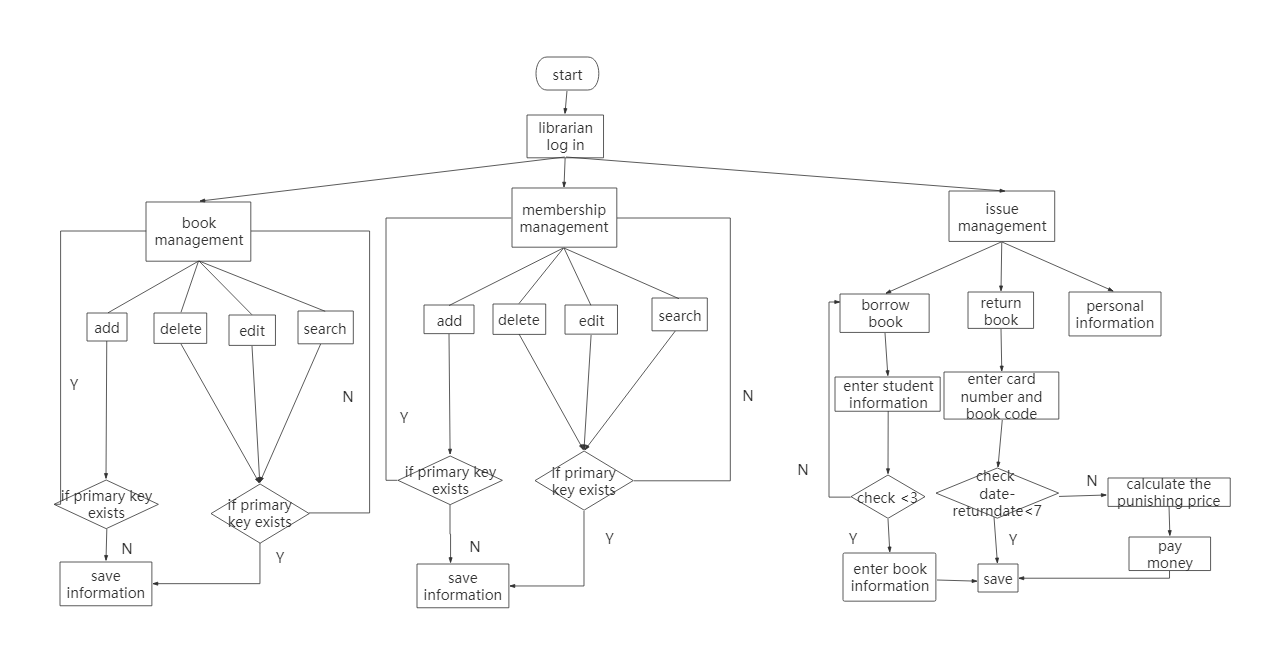
The book in the library is stayed in mysql database.when we add、delete or update,data in the library will be changed by our operation.We think the most difficult part in the design is the fine set,because we should consider the date and display the fine on the screen.

Finally we finish the project successfully.We solve lots of problems and understand the project better.

ER diagram

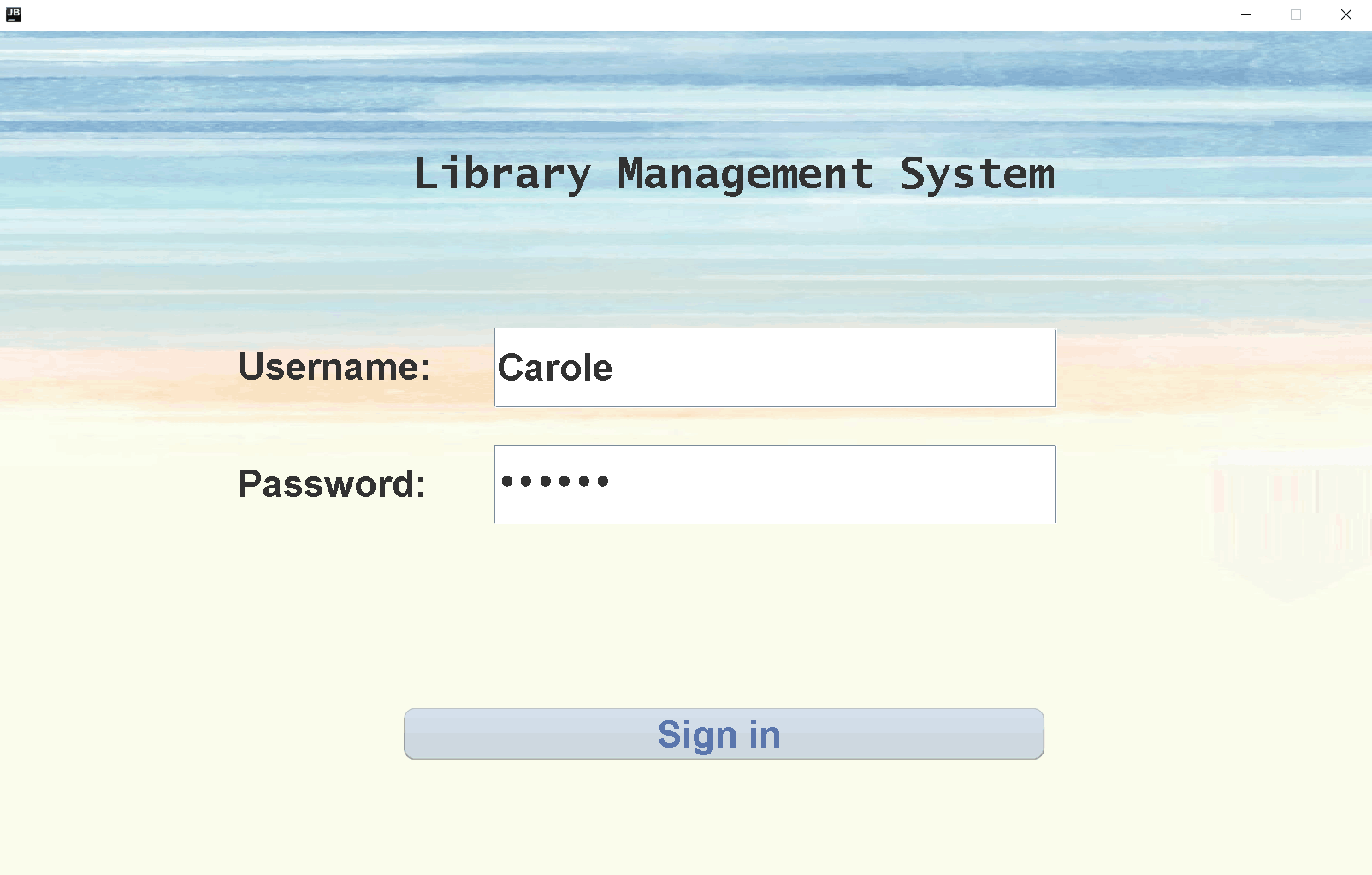


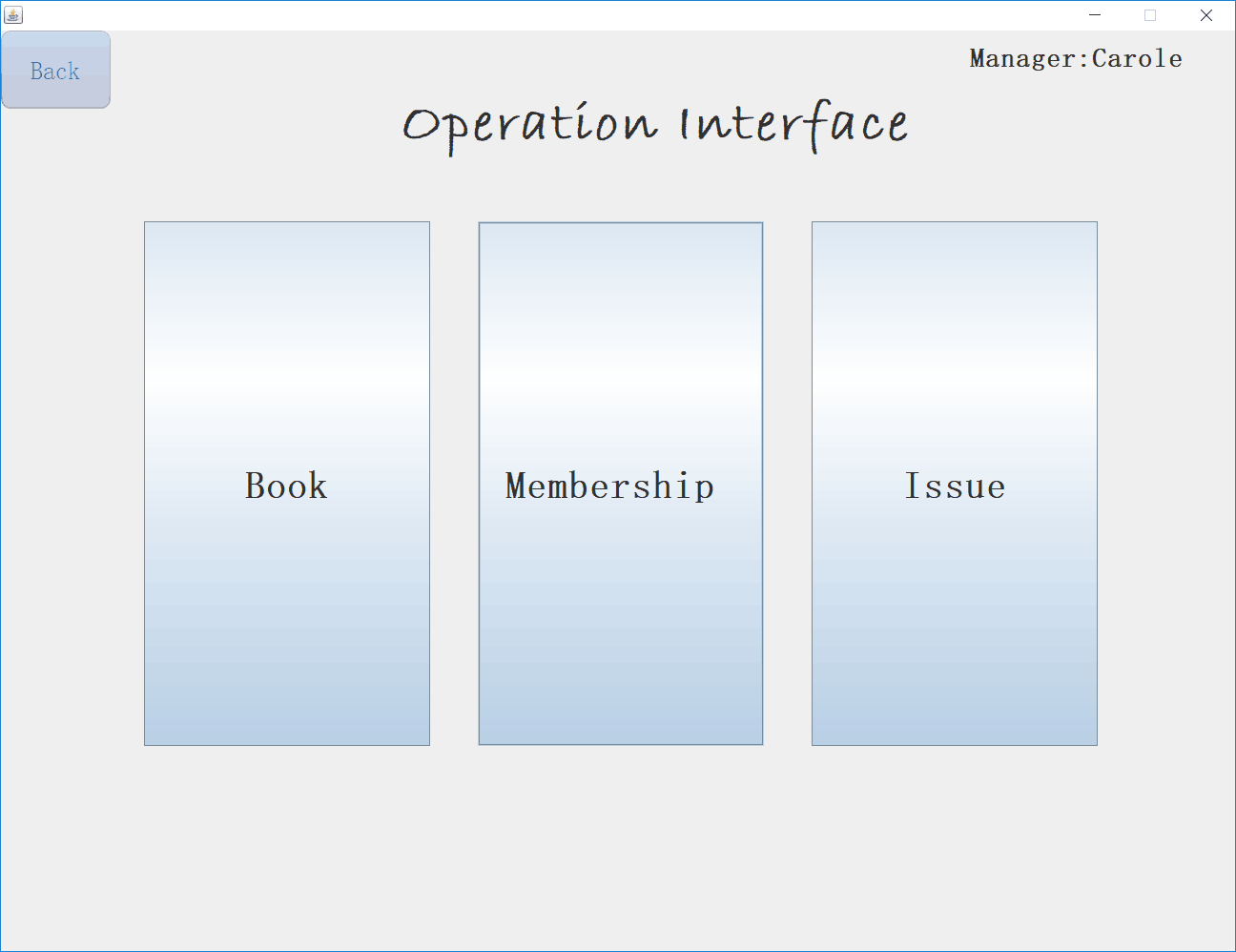
Flow Chart

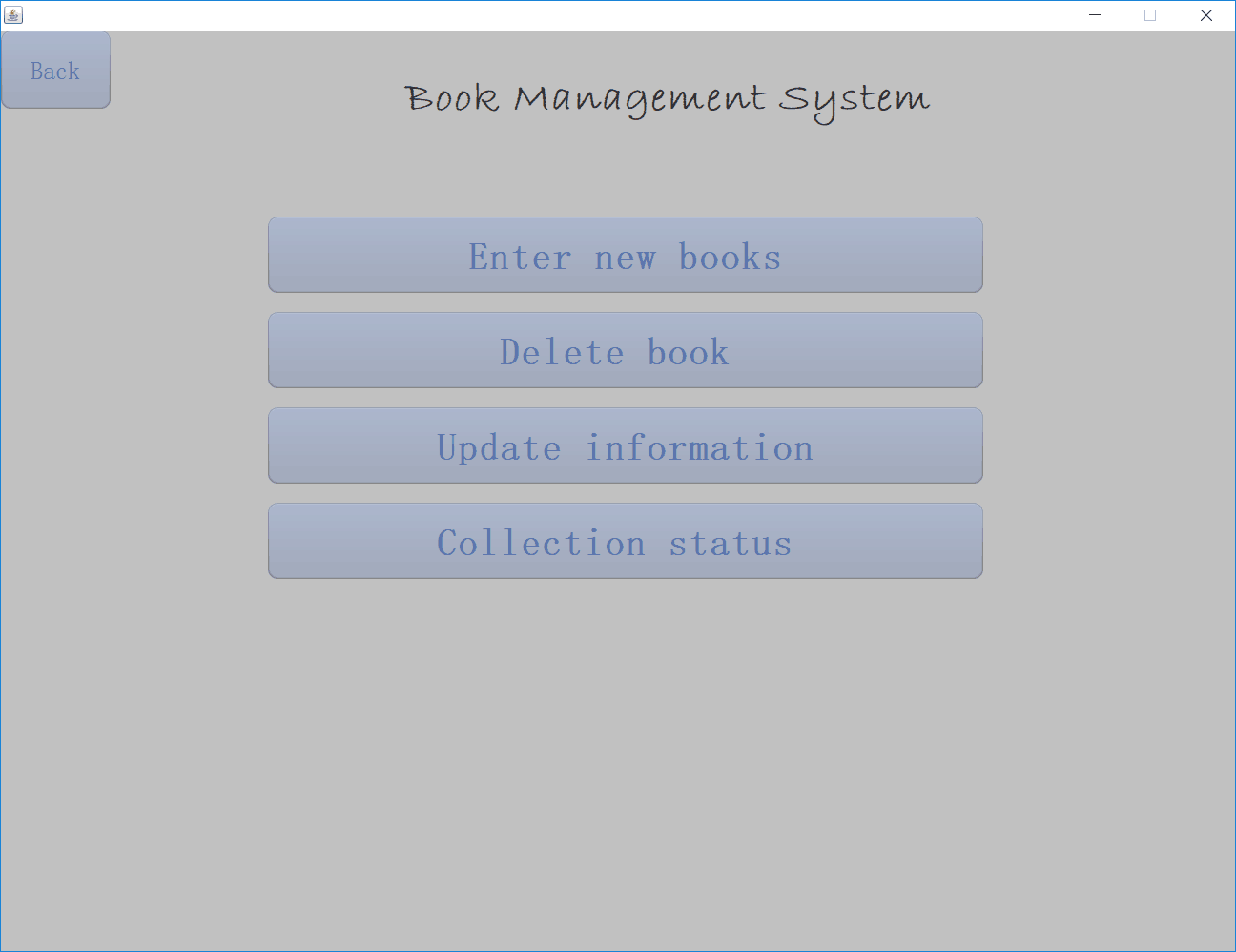


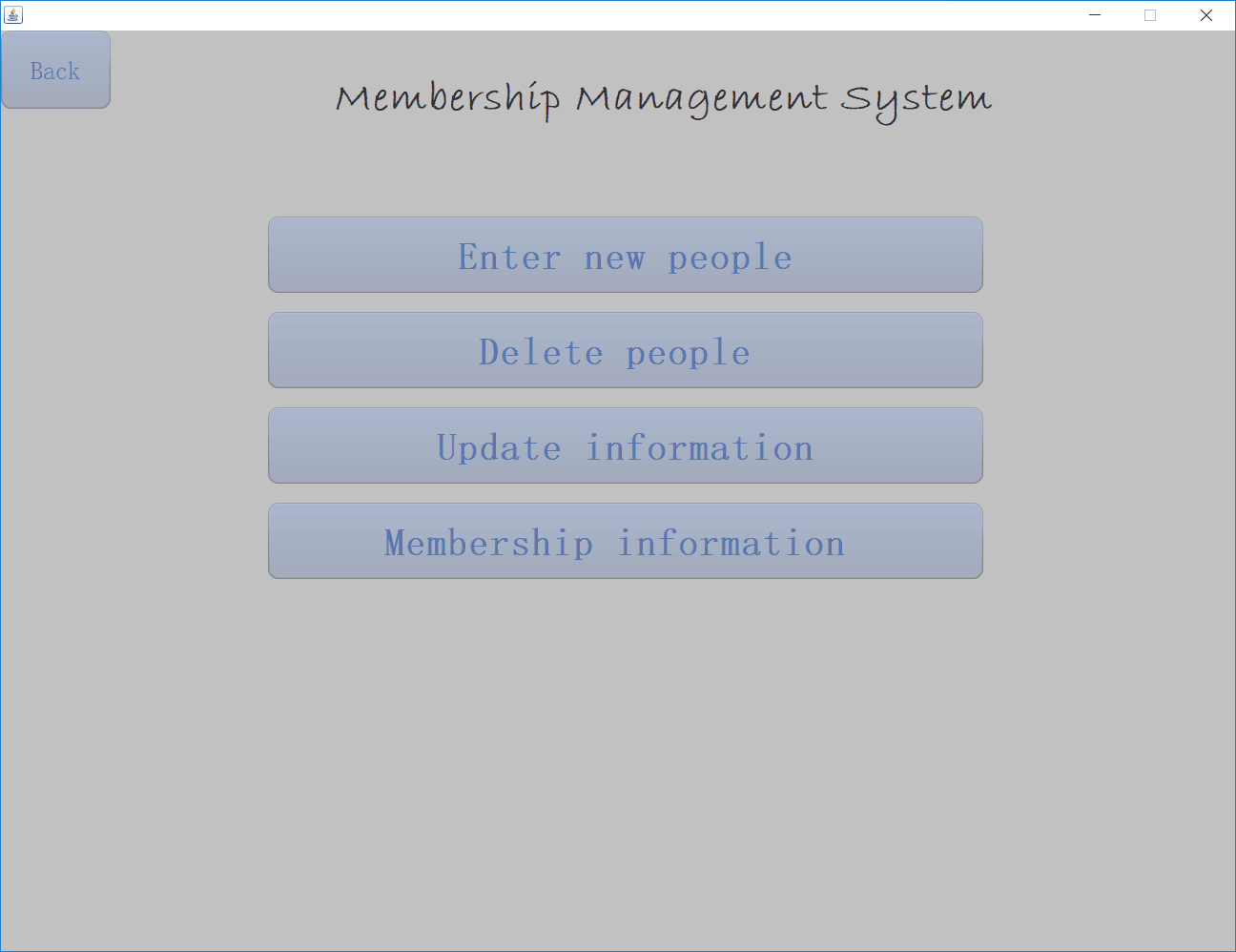
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task Description | Planned Date | Actual Date | Responsibility | Remarks |
| UI | 25th -27th June | 25th -27th June | Design user interface and implement function |  |
| DB | 25th -27th June | 25th -27th June | According to project book and teacher’s requirement to finish the document |  |
| FC | 24th June | 24th June | Draw flow chart and class diagram properly |  |
| Coding will have many rows | 27th June | 27th June | To confirm all the programs are true |  |

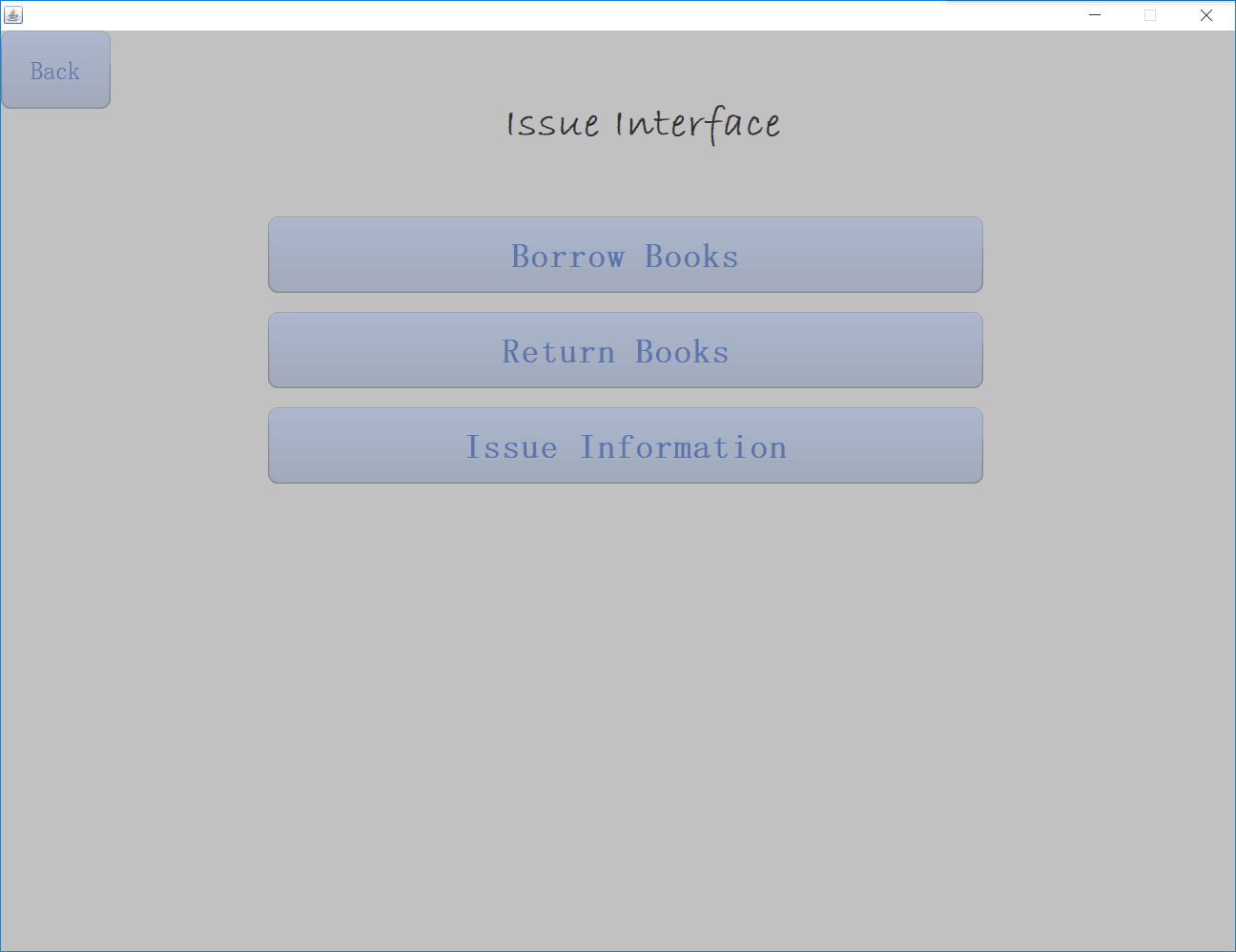
User interface design

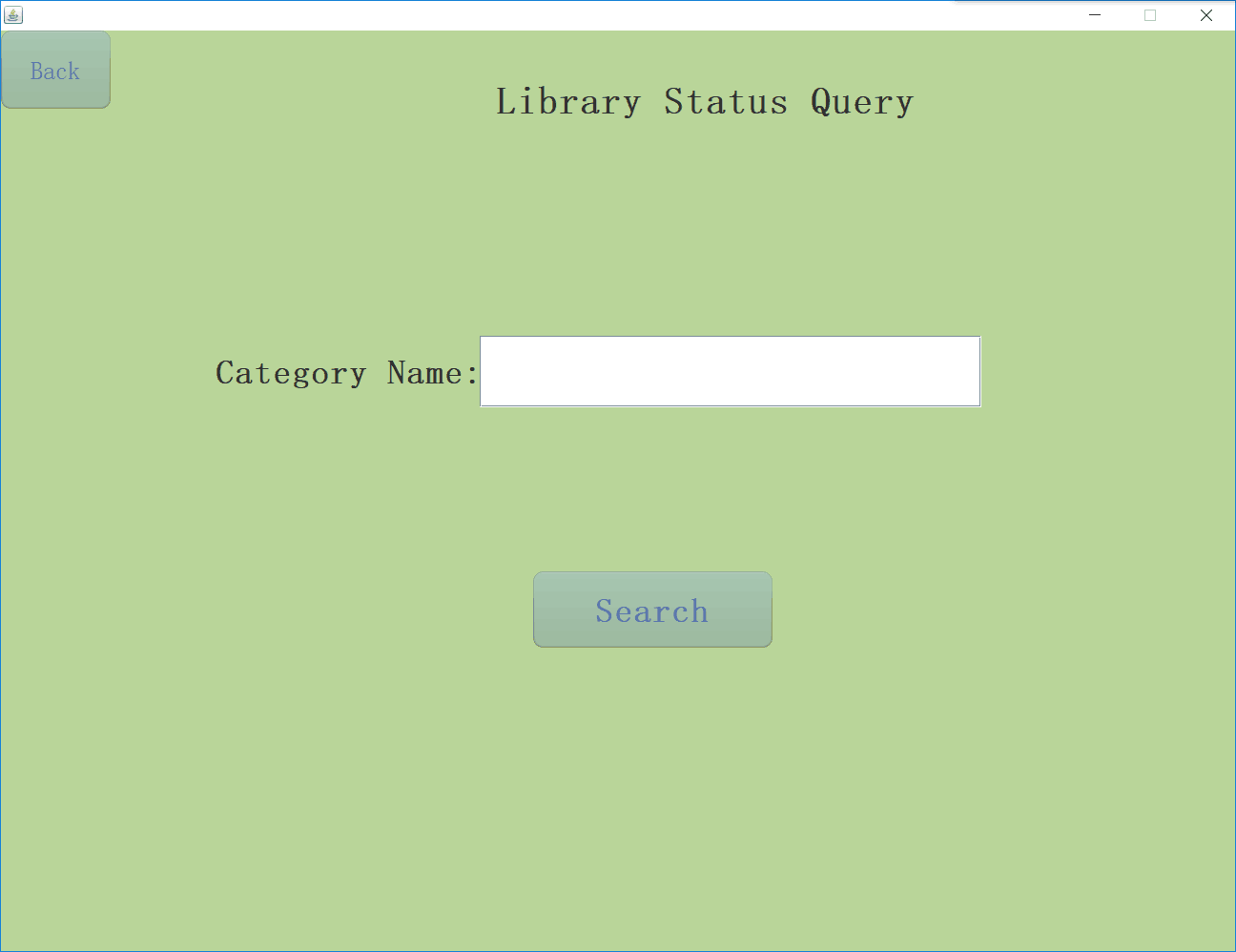


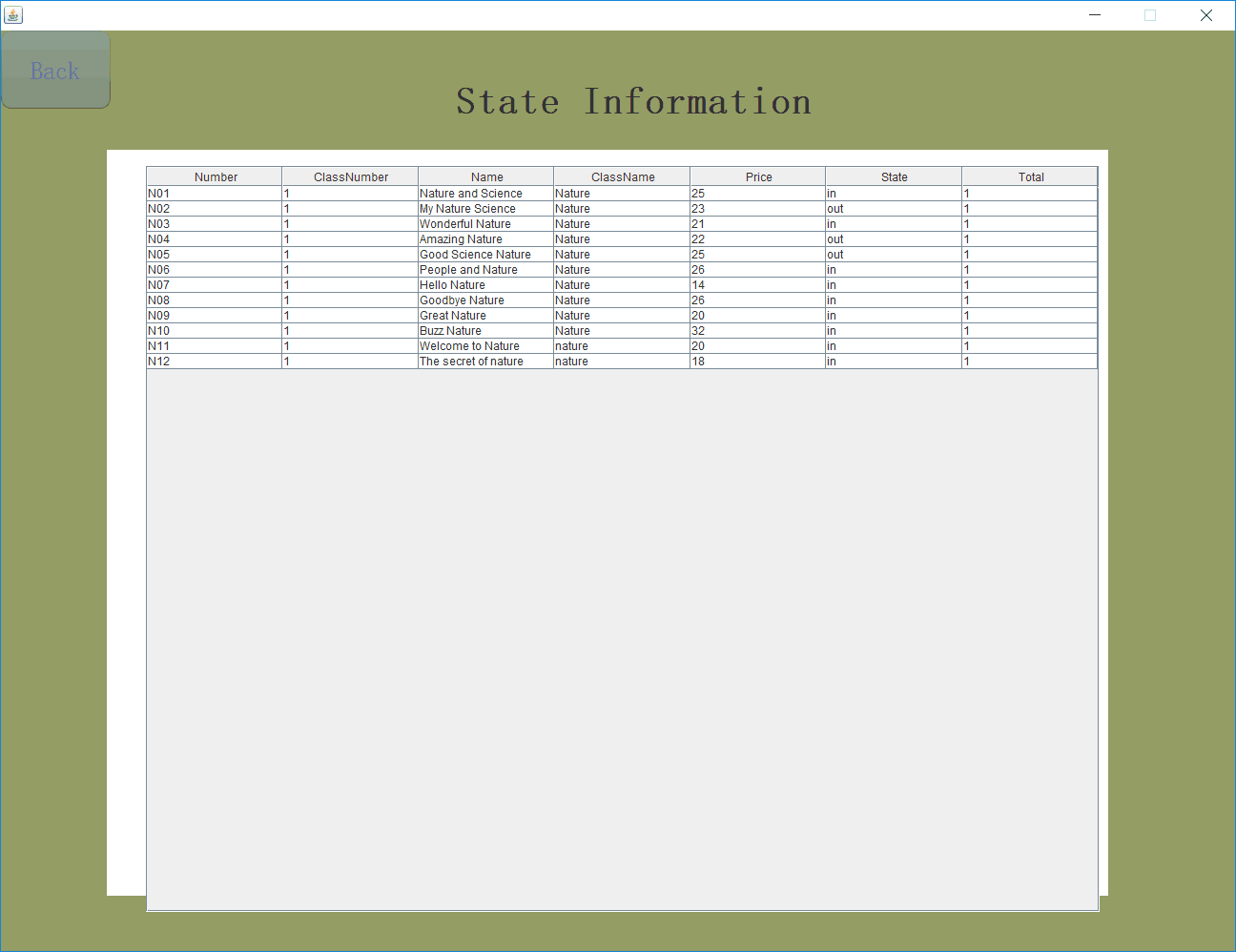


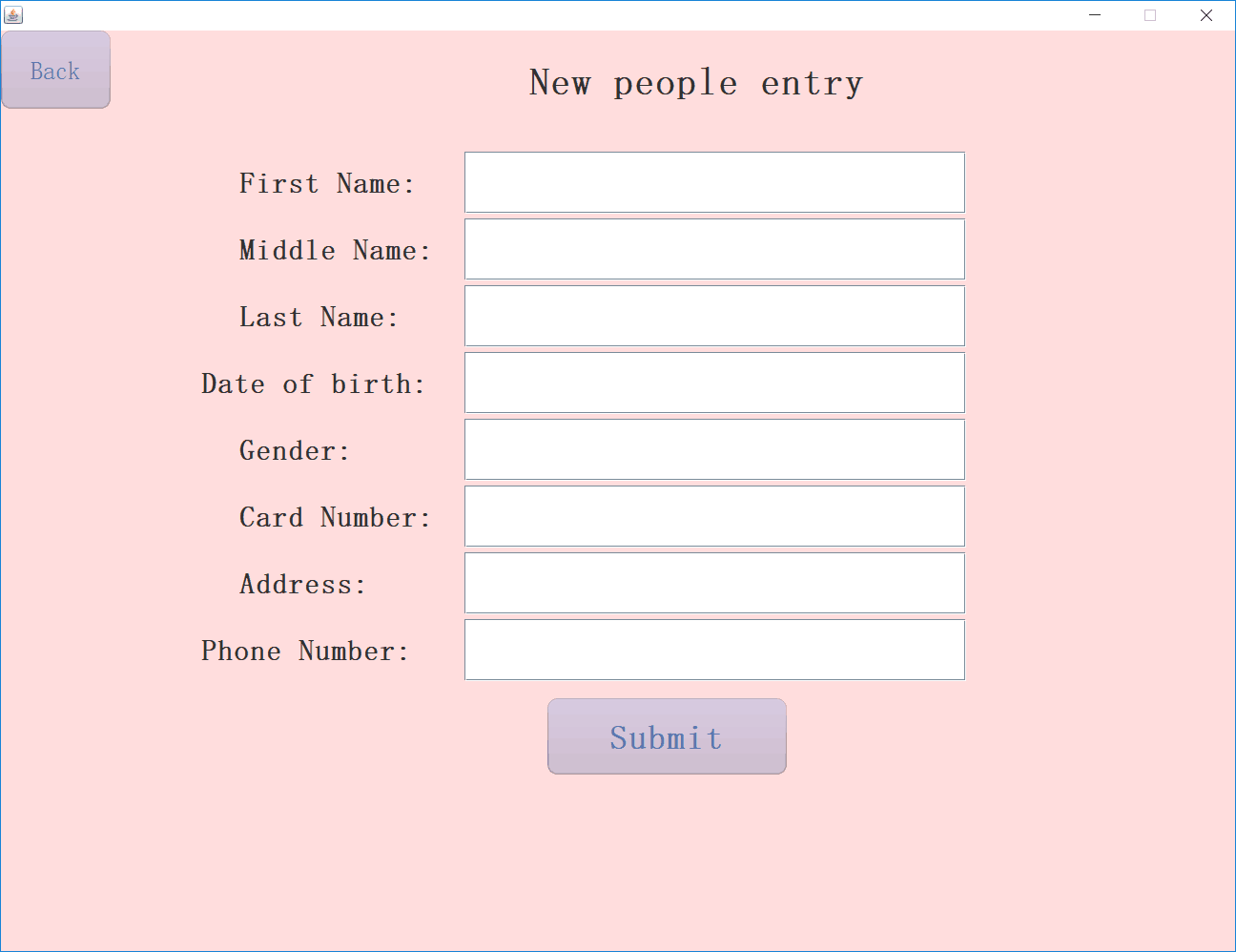


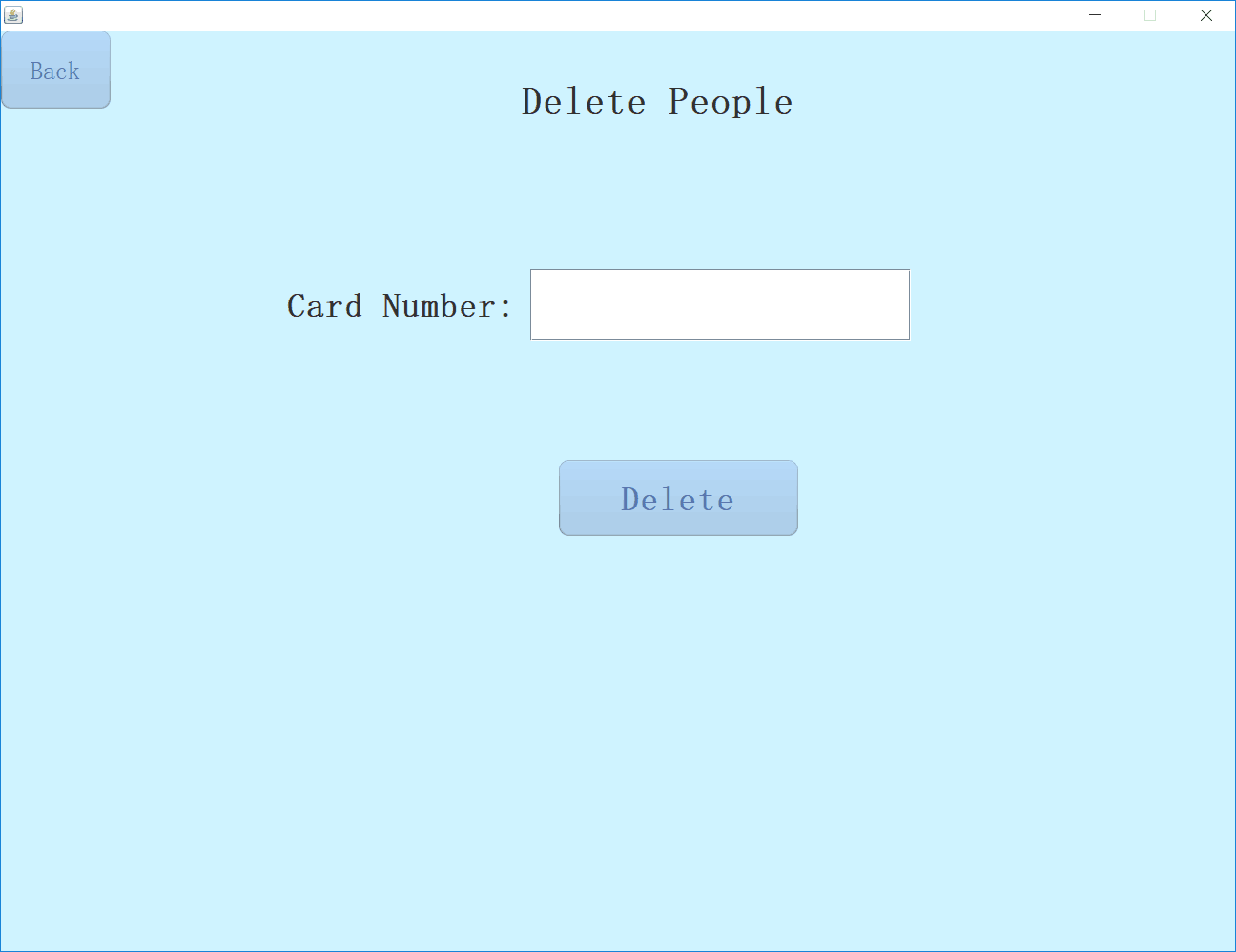


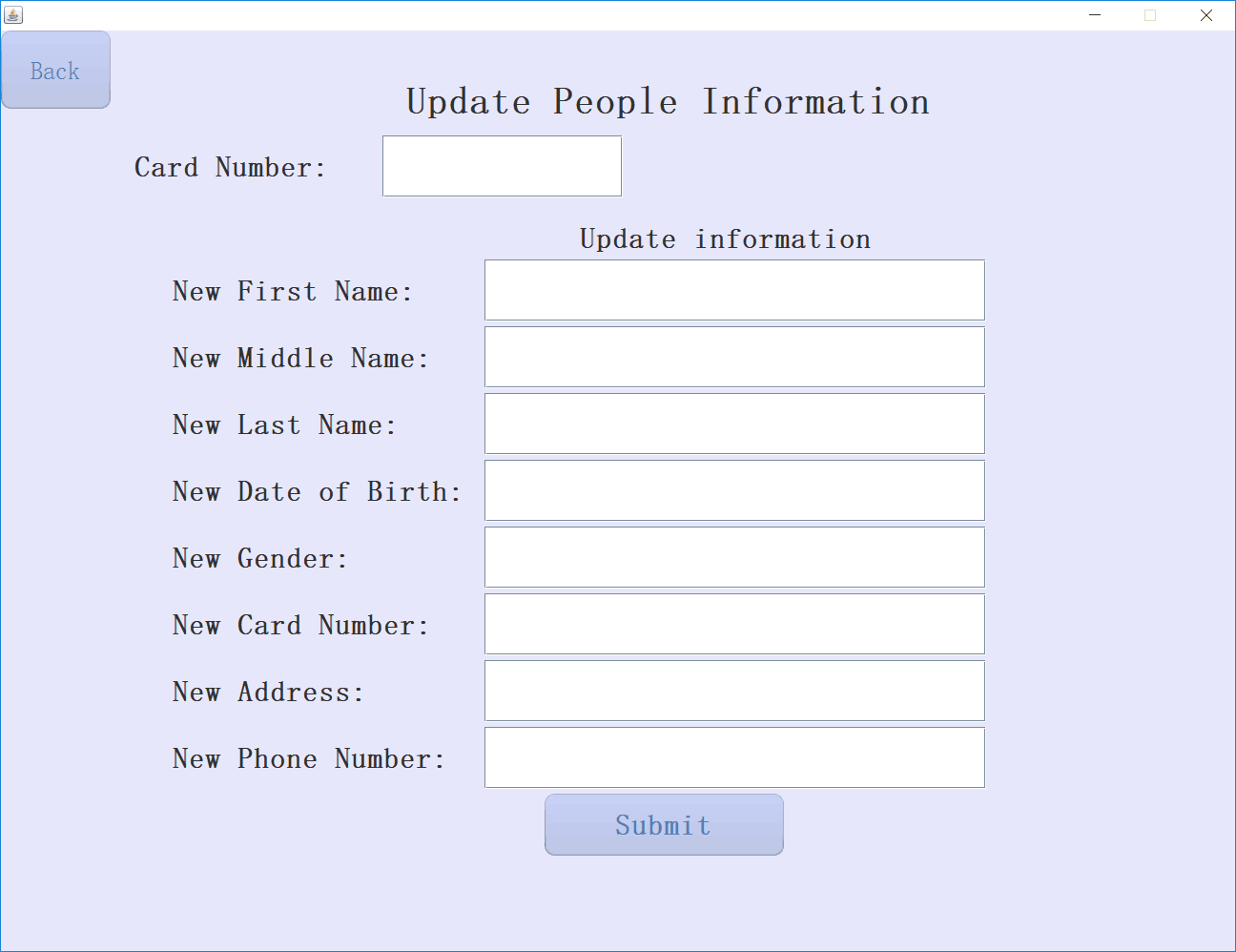




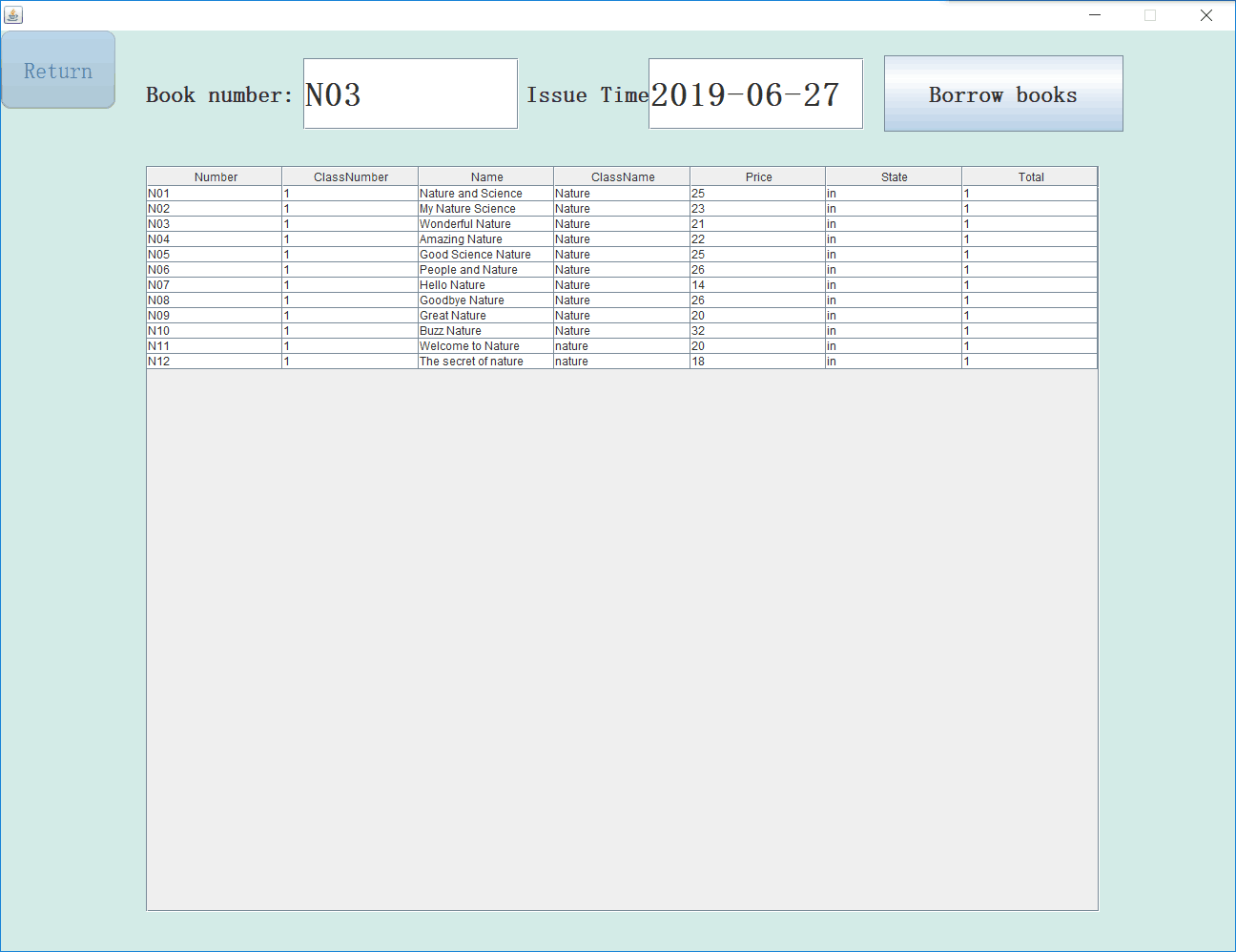


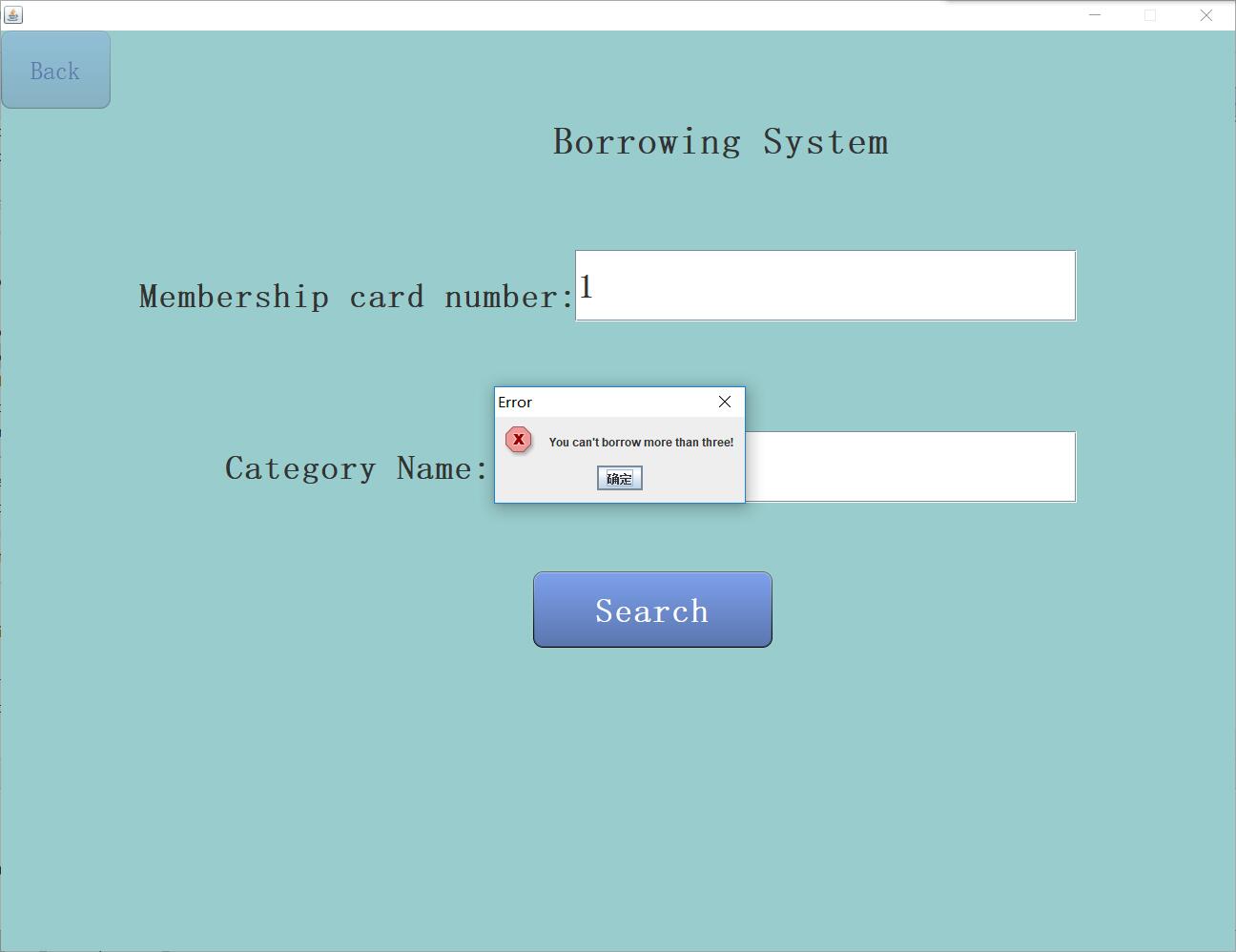


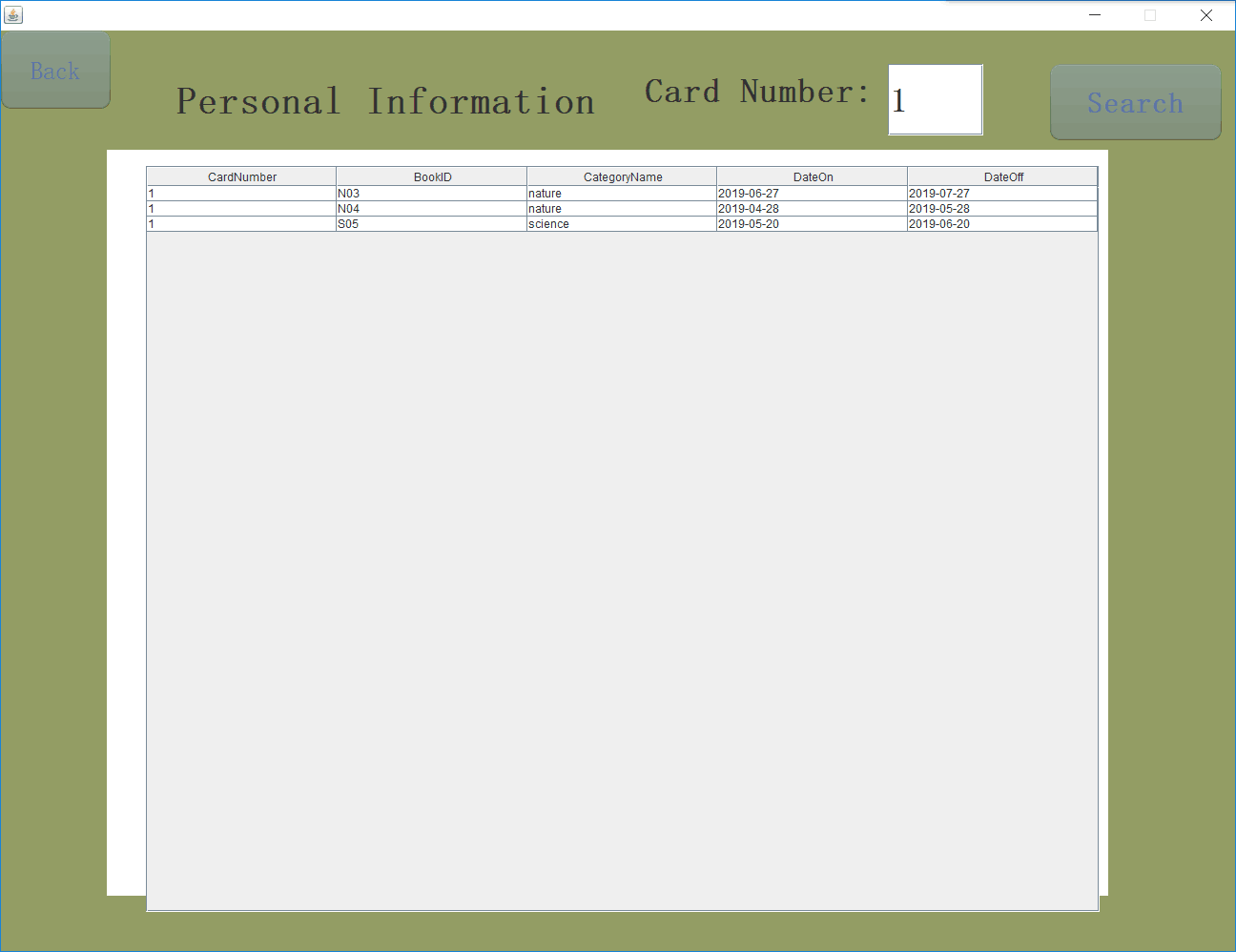


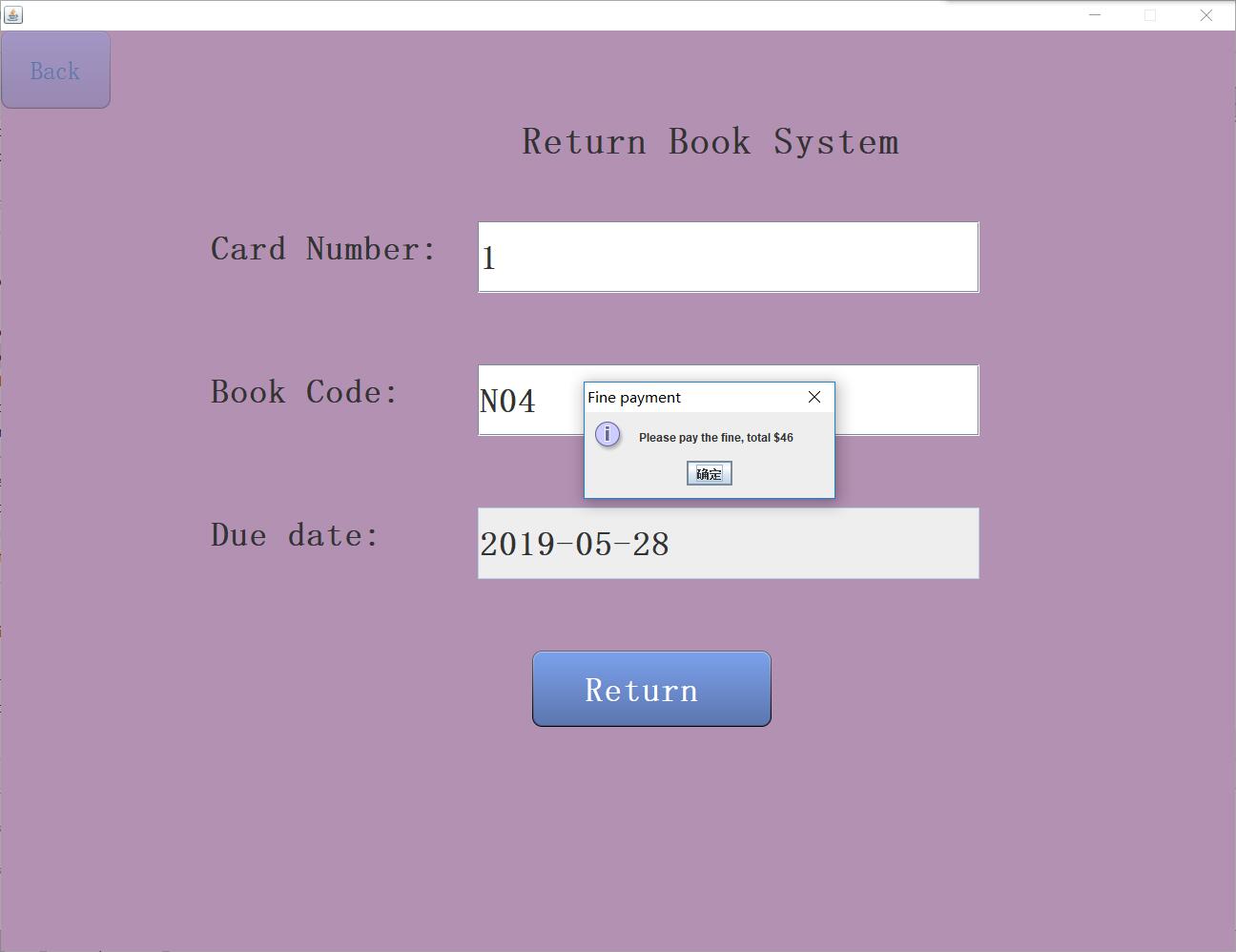


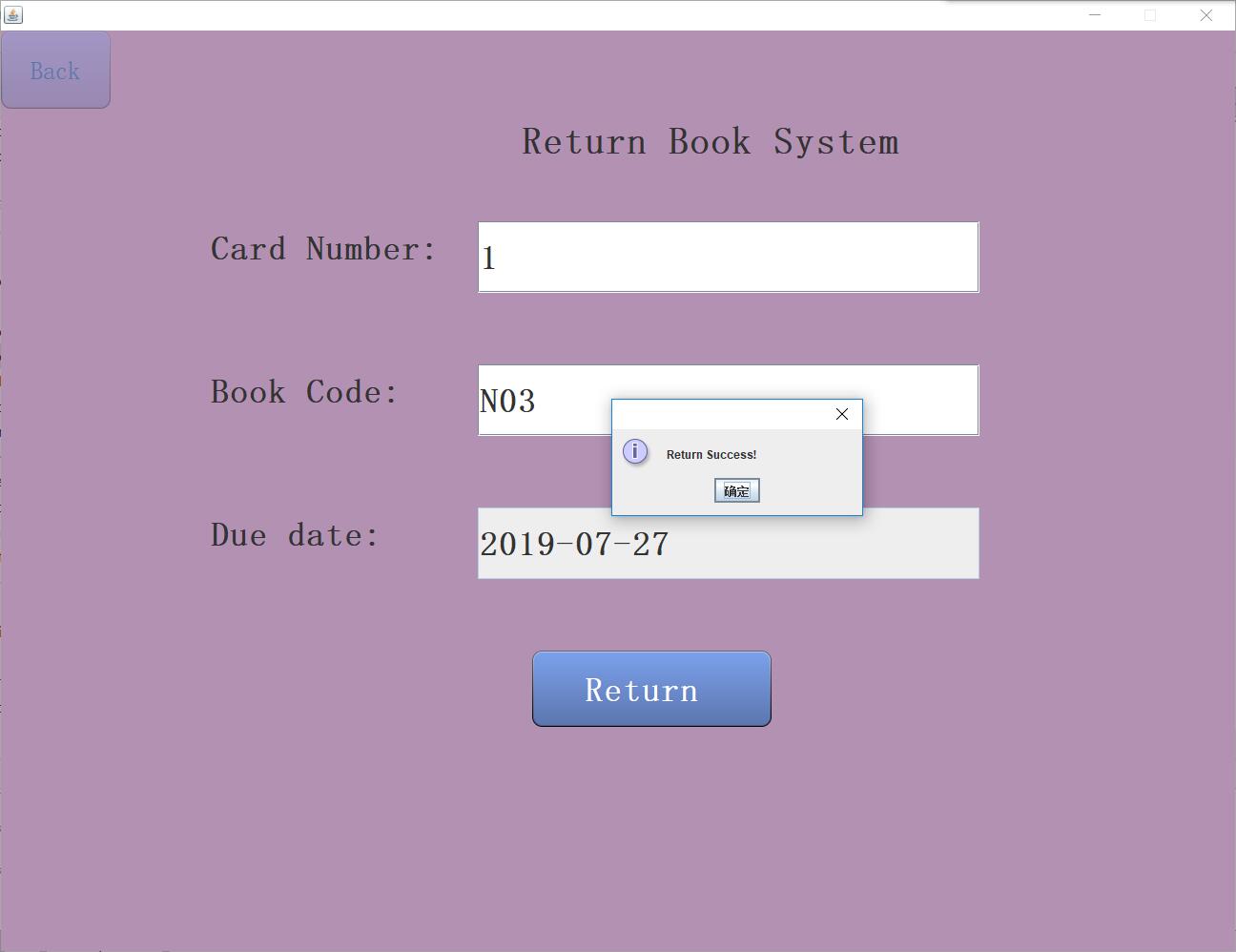
Borrow and return book is the most important part of this management.The user interface and result are below.











Important code:

public class Information {

//JDBC initialization

public static String username = "…";

public static String password = "…";

public static String JDBC\_URL = " …";

public static String search\_classname = "";

public static String search\_peoplename = "";

public static ArrayList<Book> bookArray = new ArrayList<Book>();

public static ArrayList<People> peopleArray = new ArrayList<People>();

public static String manager = "";

public static String cardnumber = "";

public static String Tto="";

//public static ArrayList<String> bootablearray = new ArrayList<String>();

}

public class TableOperate {

public TableOperate() {

try{

Class.forName("com.mysql.cj.jdbc.Driver");

System.out.println("Connect to MySQL successfully");

String url=Information.JDBC\_URL;

Connection conn = DriverManager.getConnection(url, Information.username, Information.password);

Statement stmt = conn.createStatement();

System.out.println("");

ResultSet rs = stmt.executeQuery("select \* from temp");

while (rs.next()){

System.out.print(rs.getInt(1) + "\t");

System.out.print(rs.getString(2) + "\t");

System.out.print(rs.getString(3) + "\t");

System.out.println();

}

rs.close();

stmt.close();

conn.close();

}catch(Exception e)

{

e.printStackTrace();

}

}

Main interface:

package com.cyn;

public class MainInterface extends JFrame implements ActionListener{

public static JPanel contentPane;

//Book

public static BookInterface bookJPanel = new BookInterface();

public static InsertBook insertbookJPanel = new InsertBook();

public static DeleteBook deletebookJPanel = new DeleteBook();

public static UpdateBook updatebookJPanel = new UpdateBook();

public static State stateJPanel = new State();

public static State\_Information stateInfoJPanel = new State\_Information();

//People

public static PeopleInterface peopleJPanel = new PeopleInterface();

public static InsertPeople insertpeopleJPanel = new InsertPeople();

public static DeletePeople deletepeopleJPanel = new DeletePeople();

public static UpdatePeople updatepeopleJPanel = new UpdatePeople();

public static SearchInfo searchInfoPeopleJPanel=new SearchInfo();

//Issue

public static UserInterface userJPanel = new UserInterface();

public static BorrowBook borrowJPanel = new BorrowBook();

public static BorrowBook\_Information borrowInfoJPanel = new BorrowBook\_Information();

public static ReturnBook returnJPanel = new ReturnBook();

choose interface

public static PersonalInformation personalJPanel = new PersonalInformation();

public ChooseInterface()

{

setLayout(null);

back = new JButton("Back");

back.setFont(new Font("宋体", Font.PLAIN, 26));

back.setBounds(0, 0, 115, 82);

add(back);

back.addActionListener(this);

label = new JLabel("Operation Interface ");

label.setHorizontalAlignment(SwingConstants.CENTER);

label.setFont(new Font("宋体", Font.BOLD, 40));

label.setBounds(450, 47, 550, 105);

add(label);

btn\_book = new JButton("Book");

btn\_book.setFont(new Font("宋体", Font.BOLD, 40));

btn\_book.setBounds(150, 200, 300, 550);

add(btn\_book);

btn\_book.addActionListener(this);

btn\_people = new JButton("Membership ");

btn\_people.setFont(new Font("宋体", Font.BOLD, 40));

btn\_people.setBounds(500, 200, 300, 550);

add(btn\_people);

btn\_people.addActionListener(this);

btn\_issue = new JButton("Issue");

btn\_issue.setFont(new Font("宋体", Font.BOLD, 40));

btn\_issue.setBounds(850, 200, 300, 550);

add(btn\_issue);

btn\_issue.addActionListener(this);

RButton

public class RButton extends JButton {

private static final long serialVersionUID = 39082560987930759L;

public static final Color BUTTON\_COLOR1 = new Color(125, 161, 237);

public static final Color BUTTON\_COLOR2 = new Color(91, 118, 173);

public static final Color BUTTON\_FOREGROUND\_COLOR = Color.WHITE;

private boolean hover;

public RButton(String name) {

this.setText(name);

setFont(new Font("system", Font.PLAIN, 12));

setBorderPainted(false);

setForeground(BUTTON\_COLOR2);

setFocusPainted(false);

setContentAreaFilled(false);

addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent e) {

setForeground(BUTTON\_FOREGROUND\_COLOR);

hover = true;

repaint();

}

@Override

public void mouseExited(MouseEvent e) {

setForeground(BUTTON\_COLOR2);

hover = false;

repaint();

}

});

}

@Override

protected void paintComponent(Graphics g) {

Graphics2D g2d = (Graphics2D) g.create();

int h = getHeight();

int w = getWidth();

float tran = 1F;

if (!hover) {

tran = 0.3F;

}

g2d.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,

RenderingHints.VALUE\_ANTIALIAS\_ON);

GradientPaint p1;

GradientPaint p2;

if (getModel().isPressed()) {

p1 = new GradientPaint(0, 0, new Color(0, 0, 0), 0, h - 1,

new Color(100, 100, 100));

p2 = new GradientPaint(0, 1, new Color(0, 0, 0, 50), 0, h - 3,

new Color(255, 255, 255, 100));

} else {

p1 = new GradientPaint(0, 0, new Color(100, 100, 100), 0, h - 1, new Color(0, 0, 0));

p2 = new GradientPaint(0, 1, new Color(255, 255, 255, 100), 0, h - 3, new Color(0, 0, 0, 50));

}

g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC\_OVER, tran));

RoundRectangle2D.Float r2d = new RoundRectangle2D.Float(0, 0, w - 1, h - 1, 20, 20);

Shape clip = g2d.getClip();

g2d.clip(r2d);

GradientPaint gp = new GradientPaint(0.0F, 0.0F, BUTTON\_COLOR1, 0.0F, h, BUTTON\_COLOR2, true);

g2d.setPaint(gp);

g2d.fillRect(0, 0, w, h);

g2d.setClip(clip);

g2d.setPaint(p1);

g2d.drawRoundRect(0, 0, w - 1, h - 1, 20, 20);

g2d.setPaint(p2);

g2d.drawRoundRect(1, 1, w - 3, h - 3, 18, 18);

g2d.dispose();

super.paintComponent(g);

}

}

Book

package com.cyn;

public class Book {

public String number;

public String classnumber;

public String name;

public String classname;

public String price;

public String state;

public String total;

public String current;

public String dateon;

public String dateoff;

}

People

package com.cyn;

package com.cyn;

public class People {

public String firstname;

public String middlename;

public String lastname;

public String birthday;

public String gender;

public String card\_number;

public String address;

public String tel;

}

Testing:

|  |  |  |
| --- | --- | --- |
| Test Condition Id: 1 | Tester: Cynthia | |
| Requirements Addressed: User interface/usability | | |
| Test Objective: Test conducted to check if the book can be added/delete/update. | | |
| Data Field:  I add Book id:S11; category id:2; book name:my science;Categoryname:science;price:12;state:in;number:1  I can also delete it and update it. | | Value Entered:  Book id:S11; category id:2; book name:my science;  Category name:science;price:12;state:in;number:1 |
| Expected Result: Test conducted to check if the user could add/delete/update the book. | | |
| Observed Result: I can add/delete/update book from the database. | | |
| Test Passed / Failed: The test was successful. | | |

|  |  |  |
| --- | --- | --- |
| Test Condition Id: 1 | Tester: Carole | |
| Requirements Addressed: User interface/usability | | |
| Test Objective: Test conducted to check if the people can be added/delete/update. | | |
| Data Field:  Firstname:v;middlename:a;lastname:e;birthday:5.14;gender:male;cardnumber:5;address:123;tel:12345678 | | Value Entered:  Firstname:he;middlename:xuan;lastname:ru;birthday:4.11;gender:male;cardnumber:5;address:123;tel:15544433322 |
| Expected Result: Test conducted to check if the user could add/delete/update the people in the database. | | |
| Observed Result: I can add/delete/update people from the database. | | |
| Test Passed / Failed: The test was successful. | | |

|  |  |  |
| --- | --- | --- |
| Test Condition Id: 1 | Tester: Dorjane | |
| Requirements Addressed: User interface/usability | | |
| Test Objective: Test conducted to check if people can borrow or return the book. | | |
| Data Field:  book | | Value Entered:  Borrow book：book number:N04  Issue time:2019-04-28  Return book: card number：1  Book code：N04  Due date：2019-05-28 |
| Expected Result: Test conducted to check if the user could borrow and return book. People can not borrow more than 3 books.If people return book too late,he will pay the fine. | | |
| Observed Result: People can borrow and return books from the library. People can not borrow more than 3 books.If people return book too late,he will pay the fine.The picture is displayed above in the user interface part.When return N04 book,the people should pay fine. | | |
| Test Passed / Failed: The test was successful. | | |

|  |  |  |
| --- | --- | --- |
| Test Condition Id: 1 | Tester: Teixeira | |
| Requirements Addressed: User interface/usability | | |
| Test Objective: Test conducted to check the books are existed in the database. | | |
| Data Field:  Books | | Value Entered: |
| Expected Result: Test conducted to check if the books are existed in the system. | | |
| ObservedResult: | | |
| Test Passed / Failed: The test was successful. | | |

|  |  |  |
| --- | --- | --- |
| Test Condition Id: 3.2 | Tester: Carole | |
| Requirements Addressed: User interface/usability | | |
| Test Objective: Test conducted to check if people are existed in the database. | | |
| Data Field:  people | | Value Entered:enter the attribute of membership,such as first name,middle name,last name. |
| Expected Result: Test conducted to check if people are existed in the system. | | |
| ObservedResult: | | |
| Test Passed / Failed: The test was successful. | | |

**HARDWARE AND SOFTWARE REQUIREMENTS**

Hardware: PIV-3.0GHz 865/915 Chipset 800 MHz FSB

512 MB RAM(2\*256MB)

80 GB7200 RPM Hard Drive

CD ROM Drive

10/100NIC

Optical Wheel Mouse

Keyboard

17’’Color Monitor

Operating system:Windows 10

Software: IntelliJ,MySQL WorkBench.