# A PROJECT REPORT

# ON

# LIBRARY MANAGEMENT SYSTEM

# For

# BE(Bachelor of Engineering)-First Year

# Submitted By

# Fearlessness

**composed**

Alvin（姜燕辉）No. R8J9400058

Arthur（吴水海）No. R8J9400043

Burton（杨孟霖）No. R8J9400048

Dronames（周思宇）No. R8J9400057

Charles（周煜程）No. R8J9400060

StartDate：01 May 2019

EndDate：19 May 2019

Class B6

**Internal Guide External Guide Liu De Cai Mr. Ashok**

**Computer course instructor Computer course instructor**

**** **NIIT NIIT**

PROJECT INDEX

|  |  |  |
| --- | --- | --- |
| No. | Particulars | Page No. |
| 1 | Cover page | 01 |
| 2 | Project Index | 02-03 |
| 3 | Team Description & Overview | 04 |
| 4 | Acknowledgement | 05 |
| 5 | Introduction and Software Requirements | 06 |
| 6 | Requirement Analysis | 07 |
| 7 | Software Design | 07-08 |
| 8 | About the Modules   * Login module * Student file management module * Book information management module * Retrieval module * Time management module * Book loan and return module * Tool module | 09-10 |
| 9 | Database Design | 10 |
| 10 | Testing | 10 |
| 11 | Implementation and Maintenance | 10-11 |
| 12 | Cost estimation | 12 |
| 13 | Use Case diagram and Data Flow Diagram | 13 |
| 14 | **Snapshots**   * Login module * Student file management module * Book information management module * Retrieval module * Time management module * Book loan and return module * Tool module | 14-24 |
| 15 | **Source code**   * **LogIn.java** * **Studentdetail.java** * **StudentInfo.java** * **AddStudent.java** * **DelectStudent.java** * **EditStudent.java** * **StudentView.java** * **BookDetail.java** * **BookInfo.java** * **AddBook.java** * **BorrowList.java** * **DelectBook.java** * **EditBook.java** * **BookRetrieval.java** * **BookSearch.java** * **BookTable.java** * **StudentTable.java** * **SelectEditBook.java** * **SelectEditStudent.java** * **StudentRetrieval.java** * **Date.java** * **BorrowBook.java** * **ReturnBook.java** * **tool.java** * **Home.java** * **font.java** * **SQL** | 24-123  24-28  29-32  33-34  35-38  39-41  42-45  46-50  51-54  54-55  56-59  60-61  62-64  65-69  70-74  75-77  78  79  80-81  82-83  84-88、  89  90-94  95-98  99-116  117-119  120  121-123 |

TEAM DESCRIPTION & OVERVIEW

**Team Introduction**

Our team, Fearlessness, is a dedicated, cooperative and innovative team. We all try our best to contribute more to the team with our own abilities. When encountering problems, we actively face them, fight them to the end, and gather everyone's wisdom to solve problems together. At the same time, we strive for excellence. With careful inspection and serious thinking, we constantly improve the function and structure of the project in a series of tests, in order to strive to make the written program to bring more comfortable experience to users.

**Team Numbers团队成员**

1.Name: Charles（周煜程）

The Main Responsible Part: Item design

The Specific Duties: Study various types of library management systems，understand the general framework of the library management system, draw lessons from the problems, and synthesize the above information to determine the specific framework.

2.Name: Arthur（吴水海）

The Main Responsible Part: Project development

The Specific Duties: Implement the framework's requirements through code.

3.Name: Dronames（周思宇）

The Main Responsible Part: Graphic design

The Specific Duties: Provide the required pictures for the project and complete the interface function design required by the software.

4.Name: Burton（杨孟霖）

The Main Responsible Part: Project testing

The Specific Duties: Monitor project performance under different conditions and identify potential problems.

5.Name: Alvin（姜燕辉）

The Main Responsible Part: Document writing

The Specific Duties: According to the framework and function of the program, complete the document writing.

**ACKNOWLEDGEMENT**

The successful completion of this project is inseparable from the joint efforts of everyone. Therefore, we would like to express our sincere thanks to show our respect and gratitude.

First of all, we would like to thank the teachers who taught us. They are Liu De Cai and Mr. Ashok. It is because of the flowers of knowledge that they irrigate with sweat that enable us to acquire all kinds of abilities in ignorance to create our own projects. It is because of the sword of genuine knowledge that they honed with experience that allows us to peek into our mistakes in the confusion to make up for our own shortcomings.

Secondly, we would like to thank our classmates. In the busy time of project development, they were also willing to take time to answer our questions and help us revise the program.

Next, we would like to thank the people who have contributed their wisdom on the Internet. Some of the articles they published on the Internet and the answers to some of the questions provided us with a lot of experience that can be used for reference, and also give us some ideas to improve the project.

Finally, we would like to thank ourselves for working hard during this period. Without our continuous efforts, we would not have completed the project now.

All in all, thanks to all the people who have made a contribution to this project.

**PROJECT REPORT**

**HUMAN**

**LIBRARY MANAGEMENT SYSTEM**

**Introduction**

**Project background**

The name of the project: Library Management System

The developer of this project is the library management system software development team, and the user is the library.

This product is designed for the needs of computer management books, which can complete the main functions of reader registration, addition and deletion of books, reader borrowing and returning books, modifying library information and so on.

**Project purpose**

In the environment of information age, this project puts forward the basic goal of realizing the library information management and resource sharing, so as to promote the pace of moving towards the digital library, from the book storage registration to query browsing, from reader registration to book borrowing, and an overall automatic management model has been formed. Its main function is to manage the library data information efficiently, quickly and stably. The project can mainly achieve the following functions: querying books, borrowing books, book management, reader information management, etc. Through this system, the librarian can easily realize the operation of book query, book borrowing and returning and so on. Meanwhile, the librarian can also complete the information management of the library conveniently and quickly. Through the analysis and research on the present situation and demand of the school library management system, the project uses JDBC technology to design and implement the library management system of the school library in MySQL as the back-end database, which realizes the digitization of the library information management and improves the efficiency and standardization of management.

**Software Requirements**

* J2sdk1.6
* Oracle 10g Database Server/MySQL
* IDE like IntelliJ IDEA

**Object-oriented design philosophy of the Project**

This project is an application and going to be developed using Java and MySQL technologies like JDBC similar to all other software projects. We adopt the idea of object - oriented to construct the program framework. The framework of this project will be discussed under the following heads.

**Requirement Analysis**

Before starting the project, the first thing which should be considered are the requirements. The requirements to be fill filled by the new system can be uncovered through a proper analysis of the existing system for its problems this is known as requirement analysis. Requirement analysis is the most important part to be considered before developing any project as be must be aware of the objective s that be wish to achieve by making this project. This analysis finds out the problems faced by the user or the client and the possible solution to whose problems.

* The amount of book storage and business volume is huge, so it is not feasible to rely solely on traditional bookkeeping management.
* The library needs to provide detailed information of books and library inventory for the school or needed People in society, so establishing a huge database is needed.
* Librarians need to manage the situation of borrowing and returning books, and register the borrowers. The workload is heavy. And the library management system can handle these functions efficiently and reduce the workload of Librarians.

**Software Design**

Software design is a solution- “how to approach the creation of a new system”. This important phase is composed of several steps.

* External design
* Architectural design
* Detailed design

**External Design**

As for the external design, we would like to say that due to the time limit, we can only add pleasing pictures in line with the theme of the library on the interface to improve the user experience. However, after a survey of some users, we chose the small window display mode to give users a small and light look and feel psychologically, with large keys for convenient operation.

**Internal Design**

Our internal design is divided into two parts: user interface and administrator interface. This clear division of labor enables the system to operate more efficiently. Users will be able to modify their passwords, and they will also be able to query books, borrow, return and other operations. Correspondingly, the administrator has more rights, in addition to the above functions, the administrator can also view the situation of book lending, and take the action of adding, deleting these operations, this directive is not only for the book, but also includes the user's information.

**Architectural design**

In terms of architecture design, we had an in-depth discussion and finally decided to carry out modular design, which is also the most popular way of architecture design.It can help us to meet the requirements presented in the requirements analysis.You can identify the following modules in our project:

* Login module
* Student file management module
* Book information management module
* Retrieval module
* Time management module
* Book loan and return module
* Tool module

**About the Modules**

**Login module**

This module makes sure that all access to the system is proper and authenticated. All the links and navigation paths are available for manipulating any kink of information only after an authorized person enters a valid set of user ID and PASSWORD. So this module serve as an entry point for the system and only the authorized person can access it.

**Student file management module**

This module provides an interface for new learners to enter information. The information added once can also be edited. All information about student files is stored. The profile includes fields such as ID, name, gender, year, major, login password, and so on.

**Book information management module**

This module provides an interface for updating information about library books. All information about books is stored. Profiles include fields for adding, lending, deleting, returning, and modifying information.

**Book retrieval module**

The book retrieval module provides an interface for both students and administrators to access the database and view the book information. We have conducted in-depth research and exploration on this module. It now has fuzzy search. This feature allows the user to find all the information by entering only a few characters of the book.

**Time management module**

Details about the date of return can be found in this module. This interface provides the timing function, starting from the successful borrow timing. After the specified return time, the system will prohibit the ID of the book to borrow another book until it is returned. The administrator can view the borrowing report of all the books.

**Book loan and return module**

The library loan and return module is also provided to the administrator and user interface. The administrator's interface is designed to prevent user error. When borrowing and returning books, the borrower needs to enter the ID and the number of the book.

**Database Design**

The whole project is in fact an information management system, which interacts with the database having a number of tables. The tables with their column names and data type are as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| book | | | | |
| **Field Name** | **Data Type** | **Size** | **Primary Key** | **Null** |
| bkid | Varchar30 |  | Y | N |
| bkname | Varchar30 |  |  | N |
| author | Varchar30 |  |  |  |
| press | Varchar30 |  |  |  |
| bkdate | Varchar50 |  |  |  |
| introduction | Varchar50 |  |  |  |
| Sum | Int11 |  |  |  |
| num | Int11 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| borrow | | | | |
| **Field Name** | **Data Type** | **Size** | **Primary Key** | **Null** |
| bkid | Varchar30 |  | Y | N |
| stid | Varchar20 |  |  |  |
| brdate | Varchar50 |  |  |  |
| redate | Varchar50 |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| manager | | | | |
| **Field Name** | **Data Type** | **Size** | **Primary Key** | **Null** |
| mgname | Varchar30 |  | Y | N |
| mgpassword | Varchar30 |  |  | N |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| student | | | | |
| **Field Name** | **Data Type** | **Size** | **Primary Key** | **Null** |
| stid | Varchar20 |  | Y | N |
| stname | Varchar20 |  |  |  |
| sex | Varchar20 |  |  |  |
| proname | Varchar20 |  |  |  |
| grade | Int11 |  |  |  |
| stpassword | Varchar20 |  |  |  |
| department | Varchar20 |  |  |  |

**Testing**

For the compiled system, we ran a series of tests to make sure it didn't get errors at runtime. These are some of the methods that we use.

**Debugging**

**Unit Testing**

**System Testing**

**Integration Testing**

**Acceptance Testing**

**Login module**

The login module should always protect data from unauthorized IP access. Only authorized people have access to the available interfaces and links to add, update, and search for important data. Each visit to the system passes through this module and is redirected to the normal page only after a valid login and password have been entered.

Development of this module includes writing code for the following files

* **LogIn.java**

**Student file management module**

The student file management module is provided to the administrator to operate the student information. This module should link with the database to update all student information in real time to ensure the efficiency of the system.

Development of this module includes writing code for the following files

* **Studentdetail.java**
* **StudentInfo.java**
* **AddStudent.java**
* **DelectStudent.java**
* **EditStudent.java**
* **StudentView.java**

**Book information management module**

Book information management module is provided to the administrator to operate the book information. This module should be linked to the database to update all the book information in real time to ensure that students can borrow books in time.

Development of this module includes writing code for the following files

* **BookDetail.java**
* **BookInfo.java**
* **AddBook.java**
* **BorrowList.java**
* **DelectBook.java**
* **EditBook.java**

**Retrieval module**

The retrieval module provides an accurate fuzzy search. What it brings is a greater possibility for users to search for books, and of course it also enables administrators to search for user information precisely. Development of this module includes writing code for the following files

* **BookRetrieval.java**
* **BookSearch.java**
* **BookTable.java**
* **StudentTable.java**
* **SelectEditBook.java**
* **SelectEditStudent.java**
* **StudentRetrieval.java**

**Time management module**

The time control module will enable the system to calculate the date when the borrower should return the book and prohibit him from borrowing new books again if he fails to return the book on time.

Development of this module includes writing code for the following files

* **Date.java**

**Book loan and return module**

The library borrowing and returning module is responsible for receiving and processing the user's borrowing and returning requests and synchronizing the information with the administrator in time.

Development of this module includes writing code for the following files

* **BorrowBook.java**
* **ReturnBook.java**

**Tool module**

The utility module encapsulates some commonly used classes in some programs to achieve the purpose of reuse.

* **tool.java**
* **Home.java**
* **font.java**
* **SQL**

**COST ESTIMATION**

Today, software is the most expensive element of virtually a computer - based systems. So it necessary to estimate cost of project for a developer, because

cost overrun can be disastrous for the developer.

For cost estimation of this project (RBRS), I carry out the different tasks in

This project according to the time period gives below. First we estimate the

Cost in working hours.

* Analysis carried out 1 day from 01 May 2019 to 1 May 2019.
* Design carried out 1 day from 01 May 2019 to 1 May 2019.
* Coding carried out 14 days from 02 May 2019 to 15 May 2019.
* Testing carried out 2 days from 15 May 2019 to 16 May 2019.
* Report Writing (Documentation of the project) carried out 3 days from

17 May 2019 to 19 May 2019

Given Table shows starting and ending date of each work task and total working days. Table also shows concurrency between two or more work tasks. Code generation, Report Working, and Integration & Testing work tasks have been completed in concurrent manner.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Work Done By** | **Work task** | **Starting Date** | **Ending Date** | **Duration(day)** |
| Charles（周煜程） | Analysis | 1 May 2019 | 1May 2019 | 1 |
| Arthur（吴水海） | Coding | 1May 2019 | 1May 2019 | 14 |
| Dronames（周思宇） | Design | 21May 2019 | 5June 2019 | 1 |
| Burton（杨孟霖） | Testing | 15May 2019 | 16May 2019 | 2 |
| Alvin（姜燕辉） | Reporting writing | 17May2019 | 19May 2019 | 3 |
|  |  |  |  | Total Day = 20 |

This software has been developed in 20 days. I work on this project 6 hours a day.

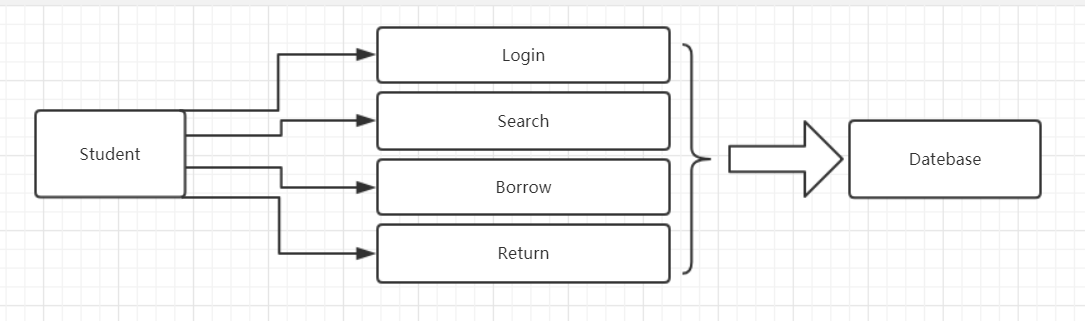
The total hours = 20\*6 = 120

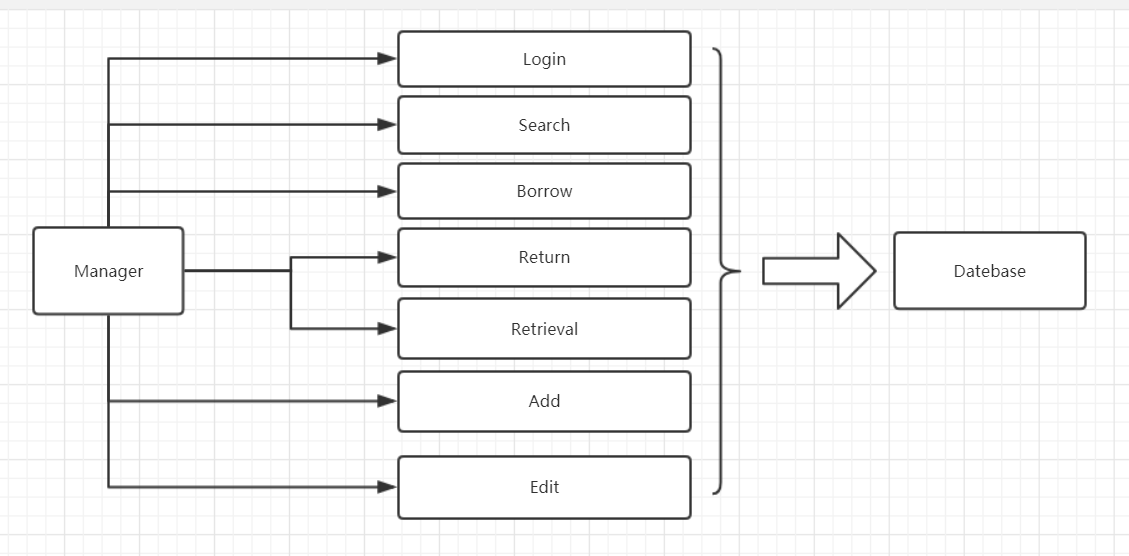
Estimated the working cost of manpower is $10/Hour.

Then the total cost of the project is = 120\*10 = 1200

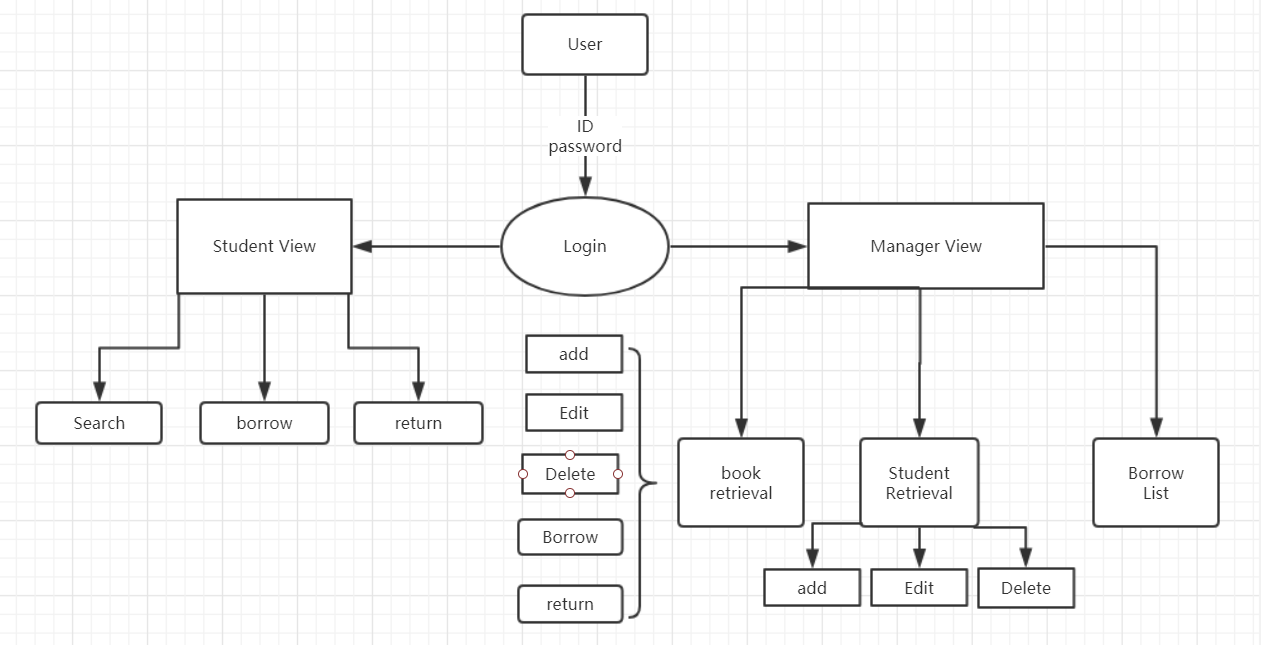
So the Estimated Total cost of the Project is One Thousand, Two Hundred,

**Use Case diagram**



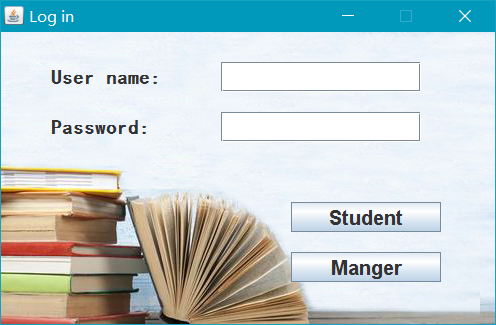


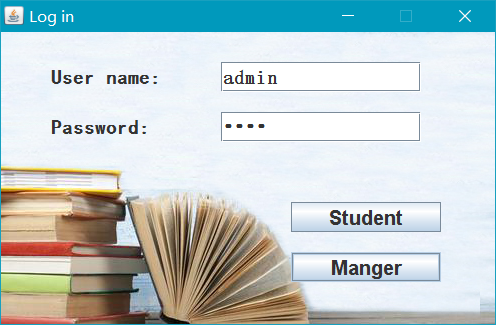
**Data Flow Diagram:**

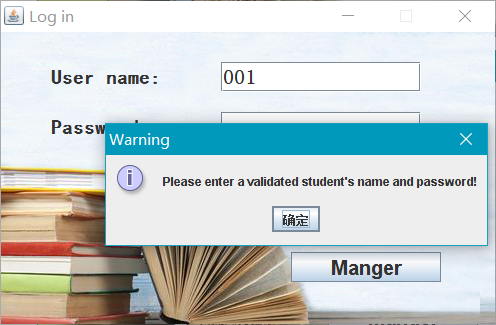


**Snapshots**

**Login module**









**LogIn.java**

package com.company.View;

import com.company.Tool.tool;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

import java.sql.SQLException;

//登录界面

public class LogIn {

public static void main(String[] args) {

new Main();

}

}

class Main extends JFrame implements ActionListener{

public Main() {

bg = new JLabel(img);

jFrame.setTitle("Log in");

jFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jFrame.setVisible(true);

jFrame.setResizable(false);

jFrame.setLocationRelativeTo(null);

jFrame.setSize(img.getIconWidth(),img.getIconHeight());

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

userName.setBounds(50,30,150,30);

password.setBounds(50,80,150,30);

nameField.setBounds(220,30,200,30);

passField.setBounds(220,80,200,30);

Student.setBounds(290,170,150,30);

Manger.setBounds(290,220,150,30);

displayArea.setBounds(50,200,300,50);

nameField.setFont(font.JTextFiledFont());

passField.setFont(font.JTextFiledFont());

userName.setFont(font.JLableFont());

password.setFont(font.JLableFont());

Student.setFont(font.JBottonFont());

Manger.setFont(font.JBottonFont());

Student.addActionListener(this);

Manger.addActionListener(this);

jPanel.add(userName);

jPanel.add(nameField);

jPanel.add(password);

jPanel.add(passField);

jPanel.add(Student);

jPanel.add(Manger);

jPanel.add(displayArea);

jPanel.add(bg);

jFrame.add(jPanel);

}

private JFrame jFrame=new JFrame();

private JLabel bg;

private ImageIcon img=new ImageIcon("images/Log.jpg");

private font font=new font();

private JPanel jPanel = new JPanel();

private JLabel displayArea=new JLabel();

private JLabel userName = new JLabel("User name:");

private JLabel password = new JLabel("Password:");

private JTextField nameField = new JTextField();

private JPasswordField passField = new JPasswordField();

private JButton Manger = new JButton("Manger");

private JButton Student=new JButton("Student");

@Override

public void actionPerformed(ActionEvent e) {

tool op=new tool();

if(e.getSource()==Student){

ResultSet resultSet = null;

try {

resultSet = op.stlogCheck(nameField.getText(),passField.getText());

if(resultSet.next()==true){

jFrame.dispose();

new StudentView(nameField.getText());

}

else {

JOptionPane.showMessageDialog(null, "Please enter a validated student's name and password!", "Warning", JOptionPane.INFORMATION\_MESSAGE);

nameField.setText("");

passField.setText("");

}

} catch (SQLException ex) {

ex.printStackTrace();

}

}

if(e.getSource()==Manger){

ResultSet resultSet = null;

try {

resultSet=op.mglogCheck(nameField.getText(),passField.getText());

if(resultSet.next()==true){

jFrame.dispose();

new MainJFrame();

}

else {

JOptionPane.showMessageDialog(null, "Please enter a validated manger's name and password!", "Warning", JOptionPane.INFORMATION\_MESSAGE);

nameField.setText("");

passField.setText("");

}

} catch (SQLException ex) {

ex.printStackTrace();

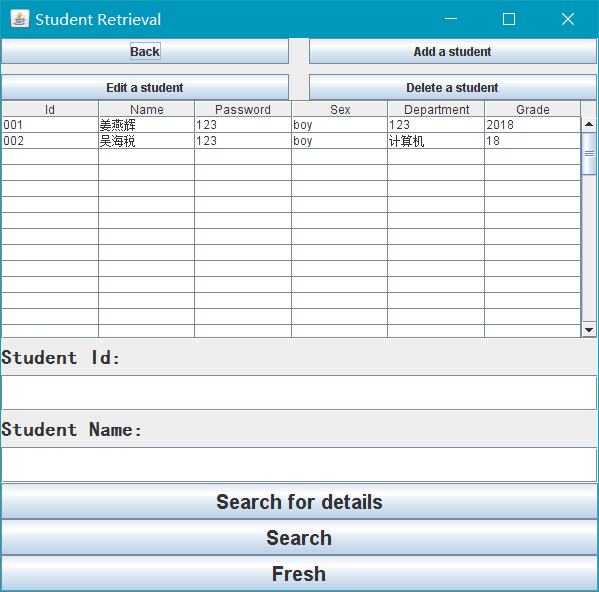
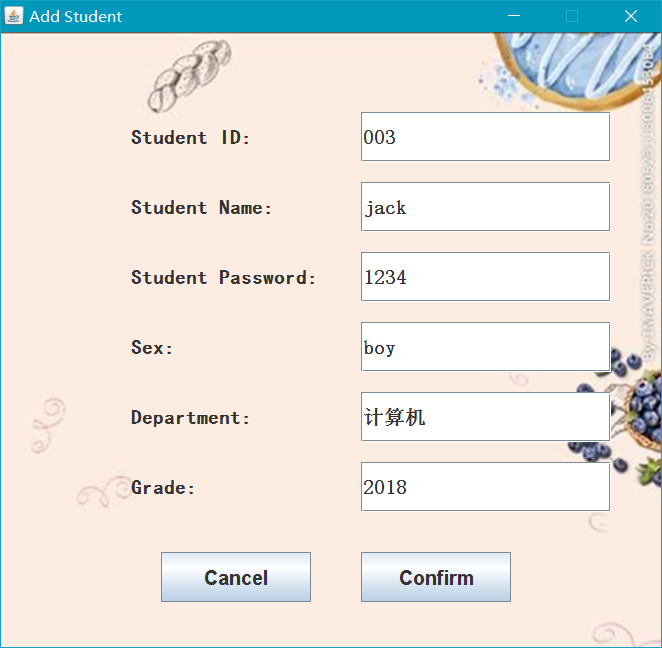
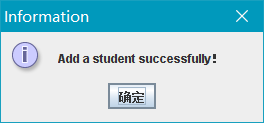
}

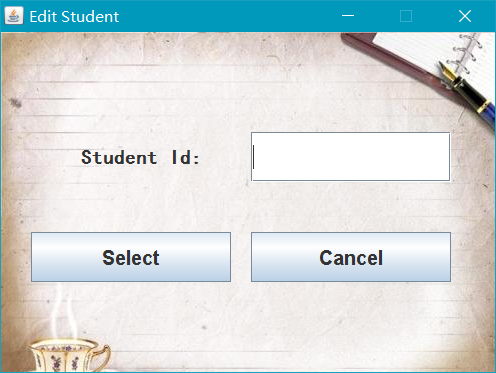
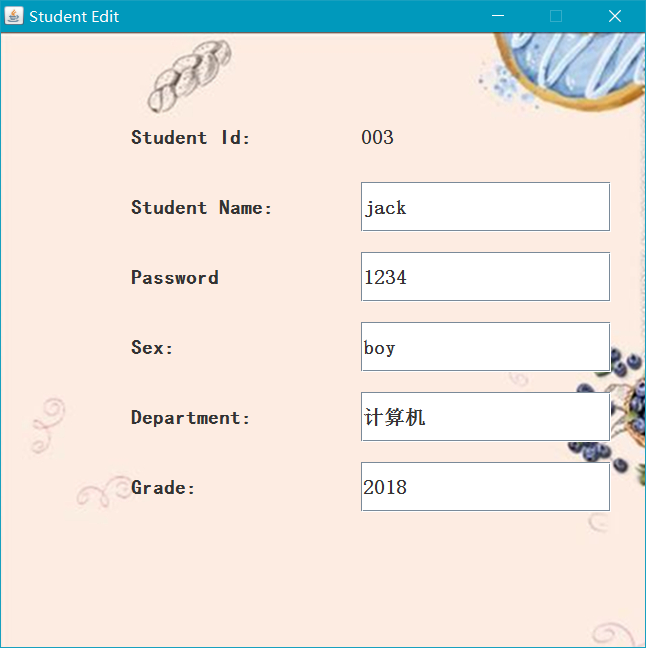
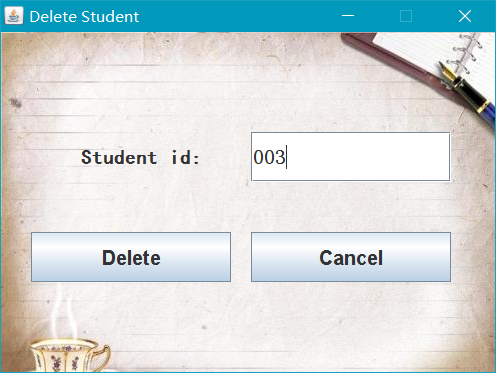
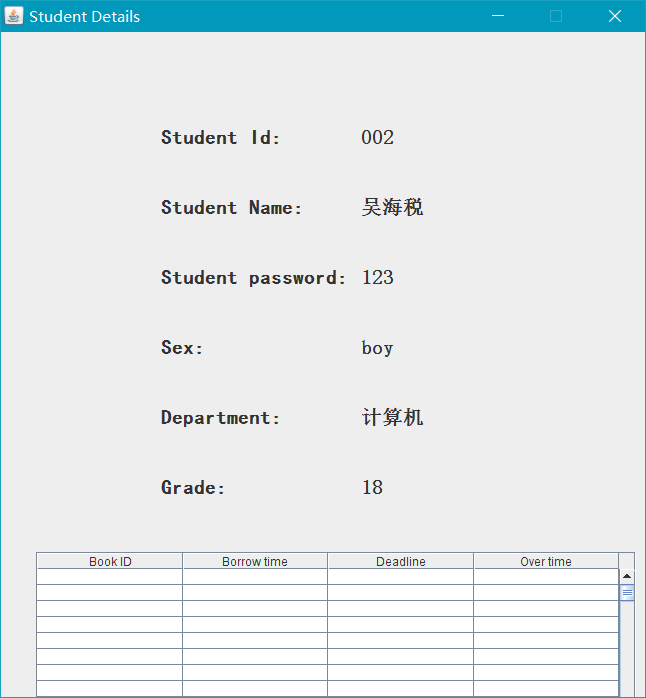
}

}

}

**Student file management module**

**Studentdetail.java**

package com.company.Student;

import com.company.Tool.font;

import com.company.Tool.tool;

import com.sun.org.apache.bcel.internal.generic.SIPUSH;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

public class Studentdetail extends JFrame implements ActionListener {

private tool op = new tool();

public Studentdetail(String id) {

bg=new JLabel(img);

this.setTitle("Student Details");

this.setLocation(300,100);

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setSize(650,700);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

scrollPane.setBounds(0, 0, 800, 300);

StudentId.setBounds(160,80,150,50);

StudentName.setBounds(160,150,150,50);

password.setBounds(160,220,200,50);

sex.setBounds(160,290,150,50);

dpt.setBounds(160,360,150,50);

grade.setBounds(160,430,150,50);

IdField.setBounds(360,80,150,50);

nameField.setBounds(360,150,150,50);

passwordField.setBounds(360,220,150,50);

sexField.setBounds(360,290,150,50);

dptField.setBounds(360,360,150,50);

gradeField.setBounds(360,430,150,50);

scrollPane.setBounds(35,520,600,200);

StudentId.setFont(font.JLableFont());

StudentName.setFont(font.JLableFont());

sex.setFont(font.JLableFont());

password.setFont(font.JLableFont());

dpt.setFont(font.JLableFont());

grade.setFont(font.JLableFont());

IdField.setFont(font.JTextFiledFont());

nameField.setFont(font.JTextFiledFont());

passwordField.setFont(font.JTextFiledFont());

sexField.setFont(font.JTextFiledFont());

dptField.setFont(font.JTextFiledFont());

gradeField.setFont(font.JTextFiledFont());

StudentInfo student = new StudentInfo(id);

ob1 = op.borrowListForStudent(id);

for (int i = 0; i < 4; i++) {

DefaultTableModel books = new DefaultTableModel(ob1, s);

for (int n = 0; n < 6; n++) {

for (int m = 0; m < 4; m++) {

ob1[n][m] = this.ob1[n][m];

}

table.setModel(books);

table.invalidate();

}

}

ArrayList<String> strArray = new ArrayList<String>();

//获得对应学生编号所有信息的数组

strArray = op.outputStudent(student);

//赋值处理

IdField.setText(id);

nameField.setText(strArray.get(1));

passwordField.setText(strArray.get(2));

sexField.setText(strArray.get(3));

dptField.setText(strArray.get(4));

gradeField.setText(strArray.get(5));

jPanel.setLayout(null);

jPanel.add(StudentId);

jPanel.add(IdField);

jPanel.add(StudentName);

jPanel.add(nameField);

jPanel.add(password);

jPanel.add(passwordField);

jPanel.add(sex);

jPanel.add(sexField);

jPanel.add(dpt);

jPanel.add(dptField);

jPanel.add(grade);

jPanel.add(gradeField);

jPanel.add(details);

jPanel.add(scrollPane);

jPanel.add(bg);

container.add(jPanel);

table.setPreferredScrollableViewportSize(new Dimension(400, 100));

cancel.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

}

private font font=new font();

private Container container = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel StudentId = new JLabel("Student Id:");

private JLabel StudentName = new JLabel("Student Name:");

private JLabel password=new JLabel("Student password:");

private JLabel sex = new JLabel("Sex:");

private JLabel dpt = new JLabel("Department:");

private JLabel grade = new JLabel("Grade:");

private JLabel IdField = new JLabel();

private JLabel nameField = new JLabel();

private JLabel passwordField=new JLabel();

private JLabel sexField = new JLabel();

private JLabel dptField = new JLabel();

private JLabel gradeField = new JLabel();

private JLabel details = new JLabel("Borrow and reaturn details");

private JButton cancel = new JButton("Cancel");

private Object[] s = {"Book ID", "Borrow time", "Deadline", "Over time"};

private Object[][] ob1 = new Object[7][5];

private JTable table = new JTable(ob1, s);

private JScrollPane scrollPane = new JScrollPane(table);

private JLabel bg;

private ImageIcon img=new ImageIcon("C:\\Users\\13278\\IdeaProjects\\library\\images\\Book.jpg");

}

**StudentInfo.java**

package com.company.Student;

public class StudentInfo {

//不含参数的构造函数

public StudentInfo(){

};

//以学生所有信息为形参的构造函数

public StudentInfo(String ID, String name, String password, String sex, String dpt, int grade) {

this.ID =ID;

this.name = name;

this.password=password;

this.sex = sex;

this.dpt = dpt;

this.grade = grade;

}

//以学生编号和姓名为形参的构造函数

public StudentInfo(String id, String name) {

this.ID = id;

this.name = name;

}

//以学生编号为形参的构造函数

public StudentInfo(String id) {

this.ID = id;

}

//以下函数返回学生的各种信息

public String getID() {

return ID;

}

public String getName() {

return name;

}

public String getSex() {

return sex;

}

public String getDpt() {

return dpt;

}

public String getPassword() {

return password;

}

public int getGrade() {

return grade;

}

private String ID;//学生编号

private String password;//学生密码

private String name;//学生名字

private String sex;//学生性别

private String dpt;//学生专业

private int grade;//学生年级

}

**AddStudent.java**

package com.company.View;

import com.company.Tool.font;

import com.company.Student.StudentInfo;

import com.company.Tool.tool;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

//添加学生界面

public class AddStudent extends JFrame implements ActionListener {

tool op = new tool();

public AddStudent() {

bg=new JLabel(img);

this.setTitle("Add Student");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),650);

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

jPanel.setLayout(null);

StudentId.setBounds(130,80,150,50);

StudentName.setBounds(130,150,150,50);

StudentPassword.setBounds(130,220,200,50);

sex.setBounds(130,290,150,50);

dpt.setBounds(130,360,150,50);

grade.setBounds(130,430,150,50);

IdField.setBounds(360,80,250,50);

nameField.setBounds(360,150,250,50);

passwordField.setBounds(360,220,250,50);

sexField.setBounds(360,290,250,50);

dptField.setBounds(360,360,250,50);

gradeField.setBounds(360,430,250,50);

confirm.setBounds(360,520,150,50);

cancel.setBounds(160,520,150,50);

StudentId.setFont(font.JLableFont());

StudentName.setFont(font.JLableFont());

StudentPassword.setFont(font.JLableFont());

sex.setFont(font.JLableFont());

dpt.setFont(font.JLableFont());

grade.setFont(font.JLableFont());

IdField.setFont(font.JTextFiledFont());

nameField.setFont(font.JTextFiledFont());

passwordField.setFont(font.JTextFiledFont());

sexField.setFont(font.JTextFiledFont());

dptField.setFont(font.JTextFiledFont());

gradeField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

confirm.setFont(font.JBottonFont());

jPanel.add(StudentId);

jPanel.add(IdField);

jPanel.add(StudentName);

jPanel.add(nameField);

jPanel.add(StudentPassword);

jPanel.add(passwordField);

jPanel.add(sex);

jPanel.add(sexField);

jPanel.add(dpt);

jPanel.add(dptField);

jPanel.add(grade);

jPanel.add(gradeField);

jPanel.add(cancel);

jPanel.add(confirm);

jPanel.add(bg);

c.add(jPanel);

cancel.addActionListener(this);

confirm.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == confirm) {

this.dispose();

StudentInfo student = new StudentInfo(IdField.getText(),nameField.getText(),passwordField.getText(),sexField.getText(),dptField.getText(),Integer.parseInt(gradeField.getText()));

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.addStudentJudgement();

int n = 0;

int replicate = 0;

while (n < Integer.parseInt(strArray.get(0))) {

n++;

if (IdField.getText().equals(strArray.get(n))) {

replicate++;

}

}

if (replicate == 0) {

if (!sexField.getText().equals("boy")

&& !sexField.getText().equals("girl")) {

JOptionPane

.showMessageDialog(

null,

"In the Sex field, you can only input 'boy' or 'girl'！",

"Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

op.addStudent(student);

JOptionPane.showMessageDialog(null,

"Add a student successfully！", "Information",

JOptionPane.INFORMATION\_MESSAGE);

}

} else {

JOptionPane.showMessageDialog(null,

"This student(id) has already existed!", "Warning",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

private font font=new font();

private Container c = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel StudentId = new JLabel("Student ID:");

private JLabel StudentName = new JLabel("Student Name:");

private JLabel StudentPassword =new JLabel("Student Password:");

private JLabel sex = new JLabel("Sex:");

private JLabel dpt = new JLabel("Department:");

private JLabel grade = new JLabel("Grade:");

private JTextField IdField = new JTextField();

private JTextField nameField = new JTextField();

private JTextField passwordField =new JTextField();

private JTextField sexField = new JTextField();

private JTextField dptField = new JTextField();

private JTextField gradeField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton confirm = new JButton("Confirm");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/Book.jpg");

}

**DelectStudent.java**

package com.company.View;

import com.company.Student.StudentInfo;

import com.company.Tool.tool;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

//删除学生界面

class DeleteStudent extends JFrame implements ActionListener {

public DeleteStudent() {

bg=new JLabel(img);

this.setTitle("Delete Student");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),img.getIconHeight());

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

StduentId.setBounds(80,100,150,50);

idField.setBounds(250,100,200,50);

cancel.setBounds(250,200,200,50);

delete.setBounds(30,200,200,50);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

StduentId.setFont(font.JLableFont());

idField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

delete.setFont(font.JBottonFont());

jPanel.add(StduentId);

jPanel.add(idField);

jPanel.add(cancel);

jPanel.add(delete);

jPanel.add(bg);

c.add(jPanel);

cancel.addActionListener(this);

delete.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == delete) {

StudentInfo student = new StudentInfo(idField.getText());

this.dispose();

if (op.deleteStudentCheck(idField.getText()) != 0) {

JOptionPane

.showMessageDialog(

null,

"Delete the student unsuccessfully. Please return books first!",

"Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

if (op.deleteStudent(student) == 1) {

JOptionPane.showMessageDialog(null,

"Delete the student successfully！", "Information",

JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane

.showMessageDialog(

null,

"Delete the student unsuccessfully. The student does not exist!",

"Warning", JOptionPane.INFORMATION\_MESSAGE);

}

}

}

}

private font font=new font();

private tool op = new tool();

private Container c = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel StduentId = new JLabel("Student id:");

private JTextField idField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton delete = new JButton("Delete");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/jframe1.jpg");

}

**EditStudent.java**

package com.company.View;

import com.company.Student.StudentInfo;

import com.company.Tool.tool;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

//学生信息修改界面

public class EditStudent extends JFrame implements ActionListener {

tool op = new tool();

public EditStudent(String id) {

bg=new JLabel(img);

this.setTitle("Student Edit");

this.setLocation(300,100);

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setSize(650,650);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

StudentId.setBounds(130,80,150,50);

StudentName.setBounds(130,150,150,50);

StudentPassword.setBounds(130,220,150,50);

sex.setBounds(130,290,150,50);

dpt.setBounds(130,360,150,50);

grade.setBounds(130,430,150,50);

IdField.setBounds(360,80,250,50);

nameField.setBounds(360,150,250,50);

passwordField.setBounds(360,220,250,50);

sexField.setBounds(360,290,250,50);

dptField.setBounds(360,360,250,50);

gradeField.setBounds(360,430,250,50);

save.setBounds(160,520,150,50);

cancel.setBounds(360,520,150,50);

StudentId.setFont(font.JLableFont());

StudentName.setFont(font.JLableFont());

StudentPassword.setFont(font.JLableFont());

sex.setFont(font.JLableFont());

dpt.setFont(font.JLableFont());

grade.setFont(font.JLableFont());

IdField.setFont(font.JTextFiledFont());

nameField.setFont(font.JTextFiledFont());

passwordField.setFont(font.JTextFiledFont());

sexField.setFont(font.JTextFiledFont());

dptField.setFont(font.JTextFiledFont());

gradeField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

save.setFont(font.JBottonFont());

IdField.setText(id);

StudentInfo student = new StudentInfo(id);

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.outputStudent(student);

nameField.setText(strArray.get(1));

passwordField.setText(strArray.get(2));

sexField.setText(strArray.get(3));

dptField.setText(strArray.get(4));

gradeField.setText(strArray.get(5));

cancel.setFont(font.JBottonFont());

save.setFont(font.JBottonFont());

jPanel.add(StudentId);

jPanel.add(IdField);

jPanel.add(StudentName);

jPanel.add(nameField);

jPanel.add(StudentPassword);

jPanel.add(passwordField);

jPanel.add(sex);

jPanel.add(sexField);

jPanel.add(dpt);

jPanel.add(dptField);

jPanel.add(grade);

jPanel.add(gradeField);

jPanel.add(bg);

jPanel.add(cancel);

jPanel.add(save);

c.add(jPanel);

cancel.addActionListener(this);

save.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == save) {

StudentInfo student = new StudentInfo(IdField.getText(), nameField.getText(),passwordField.getText(), sexField.getText(), dptField.getText(), Integer.parseInt(gradeField.getText()));

if (!sexField.getText().equals("boy")

&& !sexField.getText().equals("girl")) {

JOptionPane

.showMessageDialog(null, "In the Sex field, you can only input 'boy' or 'girl'！", "Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

op.inputStudent(student);

JOptionPane.showMessageDialog(null, "Edit the student successfully！", "Information",

JOptionPane.INFORMATION\_MESSAGE);

this.dispose();

}

}

}

private Container c = getContentPane();

private JPanel jPanel = new JPanel();

private font font=new font();

private JLabel StudentId = new JLabel("Student Id:");

private JLabel StudentName = new JLabel("Student Name:");

private JLabel StudentPassword=new JLabel("Password");

private JLabel sex = new JLabel("Sex:");

private JLabel dpt = new JLabel("Department:");

private JLabel grade = new JLabel("Grade:");

private JLabel IdField = new JLabel();

private JTextField passwordField=new JTextField();

private JTextField nameField = new JTextField();

private JTextField sexField = new JTextField();

private JTextField dptField = new JTextField();

private JTextField gradeField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton save = new JButton("Save");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/Book.jpg");

}

**StudentView.java**

package com.company.View;

import com.company.Student.StudentInfo;

import com.company.Tool.font;

import com.company.Tool.tool;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

public class StudentView extends JFrame implements ActionListener {

public StudentView(String id){

bg=new JLabel(img);

jFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

jFrame.setLocation(450,250);

jFrame.setVisible(true);

jFrame.setSize(img.getIconWidth(),img.getIconHeight());

jFrame.setTitle("Student View");

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

search.setBounds(750,50,200,50);

Borrow.setBounds(750,200,200,50);

Return.setBounds(750,350,200,50);

save.setBounds(750,500,200,50);

Back.setBounds(750,650,200,50);

StudentId.setBounds(50,70,200,50);

StudentName.setBounds(50,180,250,50);

StudentPassword.setBounds(50,290,150,50);

sex.setBounds(50,400,200,50);

dpt.setBounds(50,510,200,50);

grade.setBounds(50,620,200,50);

IdField.setBounds(400,70,200,50);

nameField.setBounds(400,180,200,50);

passwordField.setBounds(400,290,200,50);

sexField.setBounds(400,400,200,50);

dptField.setBounds(400,510,200,50);

gradeField.setBounds(400,620,200,50);

Borrow.setFont(font.JBottonFont());

Return.setFont(font.JBottonFont());

Back.setFont(font.JBottonFont());

search.setFont(font.JBottonFont());

save.setFont(font.JBottonFont());

StudentId.setFont(font.Title());

StudentName.setFont(font.Title());

StudentPassword.setFont(font.Title());

sex.setFont(font.Title());

dpt.setFont(font.Title());

grade.setFont(font.Title());

IdField.setForeground(Color.RED);

nameField.setForeground(Color.RED);

passwordField.setForeground(Color.RED);

sexField.setForeground(Color.RED);

dptField.setForeground(Color.RED);

gradeField.setForeground(Color.RED);

IdField.setFont(font.Title());

nameField.setFont(font.Title());

passwordField.setFont(font.Title());

sexField.setFont(font.Title());

dptField.setFont(font.Title());

gradeField.setFont(font.Title());

jPanel.add(StudentId);

jPanel.add(StudentName);

jPanel.add(StudentPassword);

jPanel.add(sex);

jPanel.add(dpt);

jPanel.add(grade);

jPanel.add(IdField);

jPanel.add(nameField);

jPanel.add(passwordField);

jPanel.add(sexField);

jPanel.add(dptField);

jPanel.add(gradeField);

jPanel.add(save);

jPanel.add(search);

jPanel.add(Borrow);

jPanel.add(Return);

jPanel.add(Back);

IdField.setText(id);

StudentInfo student = new StudentInfo(id);

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.outputStudent(student);

nameField.setText(strArray.get(1));

passwordField.setText(strArray.get(2));

sexField.setText(strArray.get(3));

dptField.setText(strArray.get(4));

gradeField.setText(strArray.get(5));

jPanel.add(bg);

jFrame.add(jPanel);

save.addActionListener(this);

search.addActionListener(this);

Borrow.addActionListener(this);

Return.addActionListener(this);

Back.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==search){

new BookSearch();

}

if(e.getSource()==Borrow){

new BorrowBook();

}

if(e.getSource()==Return){

new ReturnBook();

}

if(e.getSource()==Back){

jFrame.dispose();

new Main();

}

if(e.getSource()==save){

StudentInfo student = new StudentInfo(IdField.getText(), nameField.getText(),passwordField.getText(), sexField.getText(), dptField.getText(), Integer.parseInt(gradeField.getText()));

if (!sexField.getText().equals("boy")

&& !sexField.getText().equals("girl")) {

JOptionPane

.showMessageDialog(null, "In the Sex field, you can only input 'boy' or 'girl'！", "Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

op.inputStudent(student);

JOptionPane.showMessageDialog(null, "Edit the student successfully！", "Information",

JOptionPane.INFORMATION\_MESSAGE);

this.dispose();

}

}

}

private tool op = new tool();

private font font=new font();

private JFrame jFrame=new JFrame();

private JPanel jPanel=new JPanel();

private JButton search=new JButton("Search");

private JButton Borrow=new JButton("Borrow");

private JButton Return=new JButton("Return");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/Student.jpg");

private JButton Back=new JButton("Back");

private JLabel StudentId = new JLabel("Student Id:");

private JLabel StudentName = new JLabel("Student Name:");

private JLabel StudentPassword=new JLabel("Password");

private JLabel sex = new JLabel("Sex:");

private JLabel dpt = new JLabel("Department:");

private JLabel grade = new JLabel("Grade:");

private JLabel IdField = new JLabel();

private JTextField passwordField=new JTextField();

private JLabel nameField = new JLabel();

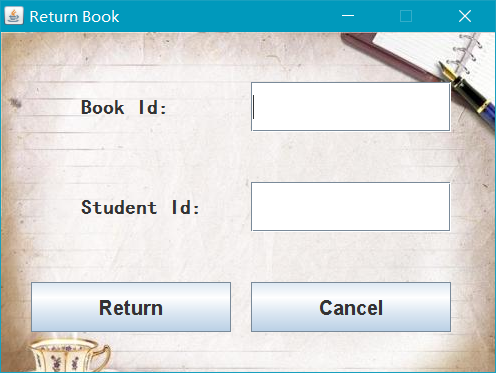
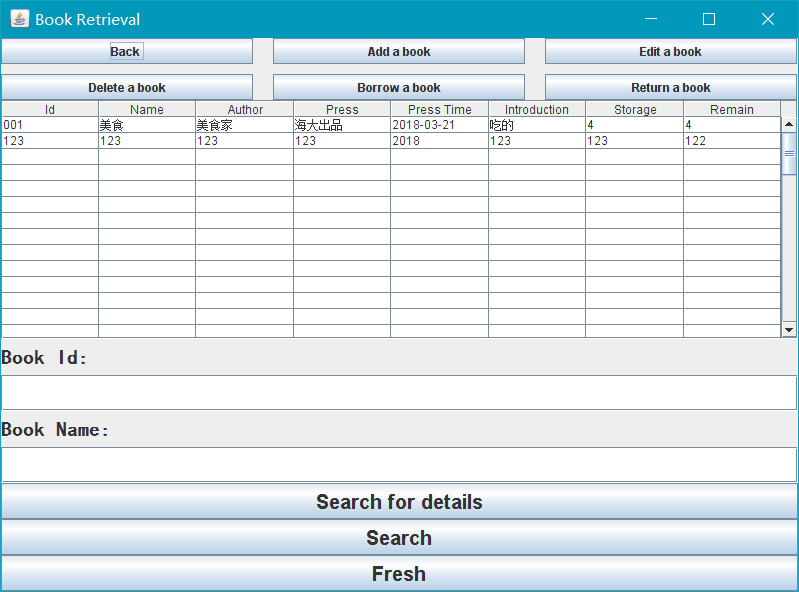
private JLabel sexField = new JLabel();

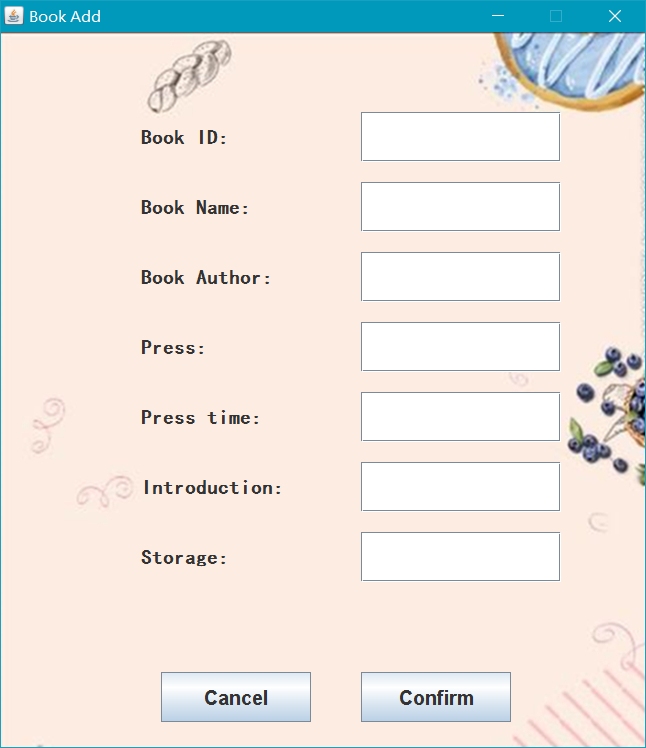
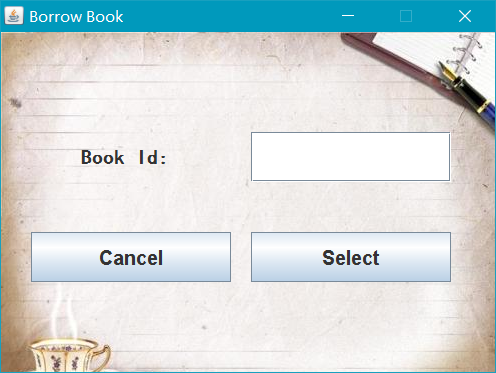
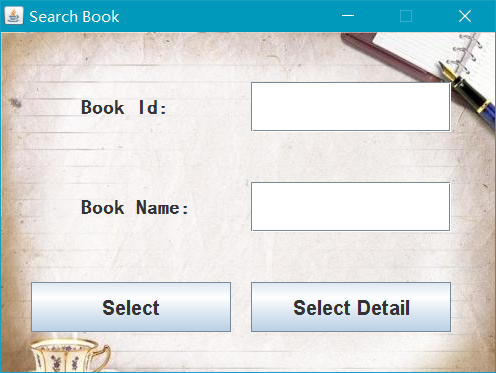
private JLabel dptField = new JLabel();

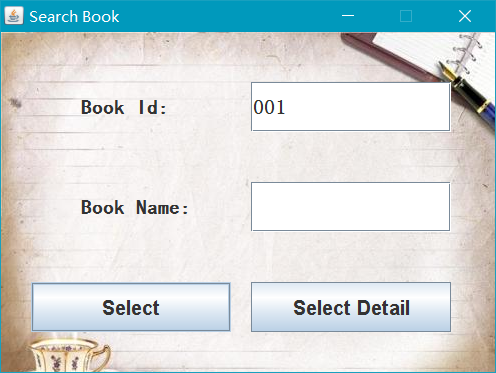
private JLabel gradeField = new JLabel();

private JButton save = new JButton("Save");

}

**Book information management module**

*   
* **BookDetail.java**
* package com.company.Book;
* import com.company.Tool.tool;
* import com.company.Tool.font;
* import javax.swing.\*;
* import javax.swing.table.DefaultTableModel;
* import java.awt.\*;
* import java.awt.event.ActionEvent;
* import java.awt.event.ActionListener;
* import java.util.ArrayList;
* //获得图书详细信息界面
* public class BookDetails extends JFrame implements ActionListener {
* tool op = new tool();
* //创建图书搜索表格
* public BookDetails(String id) {
* bg=new JLabel(img);
* this.setTitle("Book Details");
* this.setLocation(300,100);
* this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);
* this.setSize(650,900);
* this.setVisible(true);
* this.setResizable(false);
* jPanel.setLayout(null);
* bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());
* BookInfo book = new BookInfo(id);
* ob1 = op.borrowListForBook(id);
* for (int i = 0; i < 5; i++) {
* DefaultTableModel books = new DefaultTableModel(ob1, s);
* for (int n = 0; n < 7; n++) {
* for (int m = 0; m < 4; m++) {
* ob1[n][m] = this.ob1[n][m];
* }
* table.setModel(books);
* table.invalidate();
* }
* }
* ArrayList<String> strArray = new ArrayList<String>();
* strArray = op.outputBook(book);
* idField.setText(id);
* nameField.setText(strArray.get(1));
* authorField.setText(strArray.get(2));
* pressField.setText(strArray.get(3));
* pressTimeField.setText(strArray.get(4));
* abstractField.setText(strArray.get(5));
* storageField.setText(strArray.get(6));
* remainField.setText(strArray.get(7));
* bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());
* bookid.setBounds(160,80,150,50);
* bookName.setBounds(160,150,150,50);
* author.setBounds(160,220,150,50);
* press.setBounds(160,290,150,50);
* pressTime.setBounds(160,360,150,50);
* bookIntroduction.setBounds(160,430,150,50);
* storage.setBounds(160,500,150,50);
* remain.setBounds(160,570,150,50);
* details.setBounds(100,640,500,50);
* idField.setBounds(360,80,150,50);
* nameField.setBounds(360,150,150,50);
* authorField.setBounds(360,220,150,50);
* pressField.setBounds(360,290,150,50);
* pressTimeField.setBounds(360,360,150,50);
* abstractField.setBounds(360,430,150,50);
* storageField.setBounds(360,500,150,50);
* remainField.setBounds(360,570,150,50);
* scrollPane.setBounds(35,700,600,200);
* bookid.setFont(font.JLableFont());
* bookName.setFont(font.JLableFont());
* author.setFont(font.JLableFont());
* press.setFont(font.JLableFont());
* pressTime.setFont(font.JLableFont());
* bookIntroduction.setFont(font.JLableFont());
* storage.setFont(font.JLableFont());
* remain.setFont(font.JLableFont());
* details.setFont(font.Title());
* idField.setFont(font.JTextFiledFont());
* nameField.setFont(font.JTextFiledFont());
* authorField.setFont(font.JTextFiledFont());
* pressField.setFont(font.JTextFiledFont());
* pressTimeField.setFont(font.JTextFiledFont());
* abstractField.setFont(font.JTextFiledFont());
* storageField.setFont(font.JTextFiledFont());
* remainField.setFont(font.JTextFiledFont());
* jPanel.add(bookid);
* jPanel.add(idField);
* jPanel.add(bookName);
* jPanel.add(nameField);
* jPanel.add(author);
* jPanel.add(authorField);
* jPanel.add(press);
* jPanel.add(pressField);
* jPanel.add(pressTime);
* jPanel.add(pressTimeField);
* jPanel.add(bookIntroduction);
* jPanel.add(abstractField);
* jPanel.add(storage);
* jPanel.add(storageField);
* jPanel.add(remain);
* jPanel.add(remainField);
* jPanel.add(details);
* jPanel.add(scrollPane);
* jPanel.add(bg);
* container.add(jPanel);
* table.setPreferredScrollableViewportSize(new Dimension(400, 100));
* cancel.addActionListener(this);
* }
* @Override
* public void actionPerformed(ActionEvent e) {
* if (e.getSource() == cancel) {
* this.dispose();
* }
* }//设置行为监听器
* private font font=new font();
* private Container container = getContentPane();
* private JPanel jPanel= new JPanel();
* private JLabel bookid = new JLabel("Book Id:");
* private JLabel bookName = new JLabel("Book Name:");
* private JLabel author = new JLabel("Author:");
* private JLabel press = new JLabel("Press:");
* private JLabel pressTime = new JLabel("Press time:");
* private JLabel bookIntroduction = new JLabel("Introduction:");
* private JLabel storage = new JLabel("Storage:");
* private JLabel remain = new JLabel("Remain:");
* private JLabel idField = new JLabel();
* private JLabel nameField = new JLabel();
* private JLabel authorField = new JLabel();
* private JLabel pressField = new JLabel();
* private JLabel pressTimeField = new JLabel();
* private JLabel abstractField = new JLabel();
* private JLabel storageField = new JLabel();
* private JLabel remainField = new JLabel();
* private JButton cancel = new JButton("Cancel");
* private JLabel details = new JLabel("Borrow and reaturn details");
* private Object[] s = { "Student Id", "Borrow time", "Deadline", "Over time" };
* private Object[][] ob1 = new Object[7][4];
* private JTable table = new JTable(ob1, s);
* private JScrollPane scrollPane = new JScrollPane(table);
* private JLabel bg;
* private ImageIcon img=new ImageIcon("images/Book.jpg");
* }
* **BookInfo.java**
* package com.company.Book;
* //图书概况
* public class BookInfo {
* //不含参数的构造函数
* public BookInfo(){
* }
* //以图书编号为形参的构造函数
* public BookInfo(String Id) {
* this.Id = Id;
* }
* //以图书编号和书名为形参的构造函数
* public BookInfo(String Id, String name) {
* this.Id = Id;
* this.name = name;
* }
* //以所以图书信息为形参的构造函数
* public BookInfo(String Id, String name, String author, String press,
* String pressTime, String bookIntroduction, int total, int remain) {
* this.Id = Id;
* this.name = name;
* this.author = author;
* this.press = press;
* this.pressTime = pressTime;
* this.bookIntroduction = bookIntroduction;
* this.total = total;
* this.remain = remain;
* }
* //返回图书各类信息的函数
* public String getId() {
* return Id;
* }
* public String getName() {
* return name;
* }
* public String getAuthor() {
* return author;
* }
* public String getPress() {
* return press;
* }
* public String getPressTime() {
* return pressTime;
* }
* public String getBookIntroduction() {
* return bookIntroduction;
* }
* public int getTotal() {
* return total;
* }
* public int getRemain() {
* return remain;
* }
* private String Id;//图书编号
* private String name;//书名
* private String author;//作者
* private String press;//出版社
* private String pressTime;//出版时间
* private String bookIntroduction;//图书简介
* private int total;//图书总量
* private int remain;//图书库存量
* }
* **AddBook.java**
* package com.company.View;
* import com.company.Book.BookInfo;
* import com.company.Tool.tool;
* import com.company.Tool.font;
* import javax.swing.\*;
* import java.awt.\*;
* import java.awt.event.ActionEvent;
* import java.awt.event.ActionListener;
* import java.util.ArrayList;
* //添加书籍界面
* public class AddBook extends JFrame implements ActionListener {
* tool op = new tool();
* public AddBook() {
* bg=new JLabel(img);
* this.setTitle("Book Add");
* this.setLocation(300,100);
* this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);
* this.setSize(650,750);
* this.setVisible(true);
* this.setResizable(false);
* jPanel.setLayout(null);
* bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());
* bookId.setBounds(140,80,150,50);
* bookName.setBounds(140,150,150,50);
* bookAuthor.setBounds(140,220,150,50);
* press.setBounds(140,290,150,50);
* pressTime.setBounds(140,360,150,50);
* bookAbstract.setBounds(140,430,150,50);
* storage.setBounds(140,500,150,50);
* idField.setBounds(360,80,200,50);
* nameField.setBounds(360,150,200,50);
* authorField.setBounds(360,220,200,50);
* pressField.setBounds(360,290,200,50);
* pressTimeField.setBounds(360,360,200,50);
* abstractField.setBounds(360,430,200,50);
* storageField.setBounds(360,500,200,50);
* confirm.setBounds(360,640,150,50);
* cancel.setBounds(160,640,150,50);
* jPanel.setLayout(null);
* bookId.setFont(font.JLableFont());
* bookName.setFont(font.JLableFont());
* bookAuthor.setFont(font.JLableFont());
* press.setFont(font.JLableFont());
* pressTime.setFont(font.JLableFont());
* bookAbstract.setFont(font.JLableFont());
* storage.setFont(font.JLableFont());
* abstractField.setFont(font.JTextFiledFont());
* idField.setFont(font.JTextFiledFont());
* nameField.setFont(font.JTextFiledFont());
* authorField.setFont(font.JTextFiledFont());
* pressField.setFont(font.JTextFiledFont());
* pressTimeField.setFont(font.JTextFiledFont());
* storageField.setFont(font.JTextFiledFont());
* cancel.setFont(font.JBottonFont());
* confirm.setFont(font.JBottonFont());
* jPanel.add(bookId);
* jPanel.add(idField);
* jPanel.add(bookName);
* jPanel.add(nameField);
* jPanel.add(bookAuthor);
* jPanel.add(authorField);
* jPanel.add(press);
* jPanel.add(pressField);
* jPanel.add(pressTime);
* jPanel.add(pressTimeField);
* jPanel.add(bookAbstract);
* jPanel.add(abstractField);
* jPanel.add(storage);
* jPanel.add(storageField);
* jPanel.add(cancel);
* jPanel.add(confirm);
* jPanel.add(bg);
* c.add(jPanel);
* cancel.addActionListener(this);
* confirm.addActionListener(this);
* }
* @Override
* public void actionPerformed(ActionEvent e) {
* if (e.getSource() == cancel) {
* this.dispose();
* }
* if (e.getSource() == confirm) {
* this.dispose();
* BookInfo book = new BookInfo(idField.getText(), nameField.getText(), authorField.getText(), pressField.getText(), pressTimeField.getText(), abstractField.getText(), Integer.parseInt(storageField.getText()), Integer.parseInt(storageField.getText()));
* ArrayList<String> strArray = new ArrayList<String>();
* strArray = op.addBookJudgement();
* int n = 0;
* int replicate = 0;
* while (n < Integer.parseInt(strArray.get(0))) {
* n++;
* if (idField.getText().equals(strArray.get(n))) {
* replicate++;
* }
* }
* if (replicate == 0) {
* op.addBook(book);
* JOptionPane.showMessageDialog(null, "Add a book successfully！", "Information", JOptionPane.INFORMATION\_MESSAGE);
* } else {
* JOptionPane.showMessageDialog(null, "This book(id) has already existed!", "Warning", JOptionPane.INFORMATION\_MESSAGE);
* }
* }
* }
* private font font=new font();
* private Container c = getContentPane();
* private JPanel jPanel = new JPanel();
* private JLabel bookId = new JLabel("Book ID:");
* private JLabel bookName = new JLabel("Book Name:");
* private JLabel bookAuthor = new JLabel("Book Author:");
* private JLabel press = new JLabel("Press:");
* private JLabel pressTime = new JLabel("Press time:");
* private JLabel bookAbstract = new JLabel("Introduction:");
* private JLabel storage = new JLabel("Storage:");
* private JTextField idField = new JTextField();
* private JTextField nameField = new JTextField();
* private JTextField authorField = new JTextField();
* private JTextField pressField = new JTextField();
* private JTextField pressTimeField = new JTextField();
* private JTextField abstractField = new JTextField();
* private JTextField storageField = new JTextField();
* private JButton cancel = new JButton("Cancel");
* private JButton confirm = new JButton("Confirm");
* private JLabel bg;
* private ImageIcon img=new ImageIcon("images/Book.jpg");
* }
* **BorrowList.java**
* package com.company.View;
* import com.company.Tool.tool;
* import com.company.Tool.font;
* import javax.swing.\*;
* import javax.swing.table.DefaultTableModel;
* import java.awt.\*;
* import java.awt.event.ActionEvent;
* import java.awt.event.ActionListener;
* //借阅列表界面
* public class BorrowList extends JFrame implements ActionListener {
* public BorrowList() {
* Object[][] ob1 = op.borrowList();
* for (int i = 0; i < 5; i++) {
* DefaultTableModel list = new DefaultTableModel(ob, s);
* for (int n = 0; n < 100; n++) {
* for (int m = 0; m < 5; m++) {
* ob[n][m] = ob1[n][m];
* }
* table.setModel(list);
* table.invalidate();
* }
* }
* jPanel.setLayout(new GridLayout(1,1,10,10));
* back.setFont(font.JBottonFont());
* container.add(jPanel, BorderLayout.CENTER);
* container.add(jPanel1,BorderLayout.SOUTH);
* jPanel.add(scrollPane);
* jPanel1.setLayout(new GridLayout(1,1,10,10));
* jPanel1.add(back);
* table.setPreferredScrollableViewportSize(new Dimension(400, 400));
* back.addActionListener(this);
* }
* @Override
* public void actionPerformed(ActionEvent e) {
* if (e.getSource() == back) {
* this.dispose();
* new MainJFrame();
* }
* }
* private font font=new font();
* private tool op = new tool();
* private Container container = getContentPane();
* private JPanel jPanel = new JPanel();
* private JPanel jPanel1 = new JPanel();
* private JButton back=new JButton("Back");
* private Object[] s = { "Book id", "Student id", "Borrow time", "Deadline","OverTime"};
* private Object[][] ob = new Object[100][5];
* private JTable table = new JTable(ob, s);
* private JScrollPane scrollPane = new JScrollPane(table);
* }
* **DelectBook.java**
* package com.company.View;
* import com.company.Book.BookInfo;
* import com.company.Tool.tool;
* import com.company.Tool.font;
* import javax.swing.\*;
* import java.awt.\*;
* import java.awt.event.ActionEvent;
* import java.awt.event.ActionListener;
* //删除图书界面
* public class DeleteBook extends JFrame implements ActionListener {
* public DeleteBook() {
* bg=new JLabel(img);
* this.setTitle("Borrow Book");
* this.setLocation(300,200);
* this.setSize(img.getIconWidth(),img.getIconHeight());
* this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);
* this.setVisible(true);
* this.setResizable(false);
* bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());
* bookId.setBounds(80,100,150,50);
* idField.setBounds(250,100,200,50);
* cancel.setBounds(30,200,200,50);
* delete.setBounds(250,200,200,50);
* jPanel.setLayout(null);
* bookId.setFont(font.JLableFont());
* idField.setFont(font.JTextFiledFont());
* cancel.setFont(font.JBottonFont());
* delete.setFont(font.JBottonFont());
* jPanel.add(bookId);
* jPanel.add(idField);
* jPanel.add(cancel);
* jPanel.add(delete);
* jPanel.add(bg);
* container.add(jPanel);
* cancel.addActionListener(this);
* delete.addActionListener(this);
* }
* public void actionPerformed(ActionEvent e) {
* // TODO Auto-generated method stub
* if (e.getSource() == cancel) {
* this.dispose();
* }
* if (e.getSource() == delete) {
* BookInfo book = new BookInfo(idField.getText());
* this.dispose();
* if (op.deleteBookCheck(idField.getText()) != 0) {
* JOptionPane
* .showMessageDialog(
* null,
* "Delete the book unsuccessfully. The book bas been borrowed!",
* "Warning", JOptionPane.INFORMATION\_MESSAGE);
* } else {
* if (op.deleteBook(book) == 1) {
* JOptionPane.showMessageDialog(null,
* "Delete the book successfully！", "Information",
* JOptionPane.INFORMATION\_MESSAGE);
* } else {
* JOptionPane
* .showMessageDialog(
* null,
* "Delete the book unsuccessfully. The book does not exist!",
* "Warning", JOptionPane.INFORMATION\_MESSAGE);
* }
* }
* }
* }
* private font font=new font();
* private tool op = new tool();
* private Container container = getContentPane();
* private JPanel jPanel = new JPanel();
* private JLabel bookId = new JLabel("Book Id:");
* private JTextField idField = new JTextField();
* private JButton cancel = new JButton("Cancel");
* private JButton delete = new JButton("Delete");
* private JLabel bg;
* private ImageIcon img=new ImageIcon("images/jframe1.jpg");
* }
* **EditBook.java**
* package com.company.View;
* import com.company.Book.BookInfo;
* import com.company.Tool.tool;
* import com.company.Tool.font;
* import javax.swing.\*;
* import java.awt.\*;
* import java.awt.event.ActionEvent;
* import java.awt.event.ActionListener;
* import java.util.ArrayList;
* //图书信息修改界面
* public class EditBook extends JFrame implements ActionListener {
* public EditBook(String id) {
* bg=new JLabel(img);
* this.setTitle("Book Edit");
* this.setLocation(300,100);
* this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);
* this.setSize(650,750);
* this.setVisible(true);
* this.setResizable(false);
* jPanel.setLayout(null);
* bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());
* remain1.setBounds(170,30,350,50);
* bookId.setBounds(160,80,150,50);
* bookName.setBounds(160,150,150,50);
* author.setBounds(160,220,150,50);
* press.setBounds(160,290,150,50);
* pressTime.setBounds(160,360,150,50);
* bookAbstract.setBounds(160,430,150,50);
* storage.setBounds(160,500,150,50);
* remain.setBounds(160,570,150,50);
* idField.setBounds(360,80,250,50);
* nameField.setBounds(360,150,250,50);
* authorField.setBounds(360,220,250,50);
* pressField.setBounds(360,290,250,50);
* pressTimeField.setBounds(360,360,250,50);
* introductionField.setBounds(360,430,250,50);
* storageField.setBounds(360,500,250,50);
* jTextField1.setBounds(360,570,250,50);
* save.setBounds(360,640,150,50);
* cancel.setBounds(160,640,150,50);
* BookInfo book = new BookInfo(id);
* ArrayList<String> strArray = new ArrayList<String>();
* strArray = op.outputBook(book);
* idField.setText(id);
* nameField.setText(strArray.get(1));
* authorField.setText(strArray.get(2));
* pressField.setText(strArray.get(3));
* pressTimeField.setText(strArray.get(4));
* introductionField.setText(strArray.get(5));
* storageField.setText(strArray.get(6));
* jTextField.setText(strArray.get(6));
* jTextField1.setText(strArray.get(7));
* bookId.setFont(font.JLableFont());
* bookName.setFont(font.JLableFont());
* author.setFont(font.JLableFont());
* press.setFont(font.JLableFont());
* pressTime.setFont(font.JLableFont());
* bookAbstract.setFont(font.JLableFont());
* storage.setFont(font.JLableFont());
* remain.setFont(font.JLableFont());
* remain1.setFont(font.Title());
* idField.setFont(font.JTextFiledFont());
* nameField.setFont(font.JTextFiledFont());
* authorField.setFont(font.JTextFiledFont());
* pressField.setFont(font.JTextFiledFont());
* pressTimeField.setFont(font.JTextFiledFont());
* introductionField.setFont(font.JTextFiledFont());
* storageField.setFont(font.JTextFiledFont());
* jTextField1.setFont(font.JTextFiledFont());
* cancel.setFont(font.JBottonFont());
* save.setFont(font.JBottonFont());
* jPanel.setLayout(null);
* jPanel.add(bookId);
* jPanel.add(idField);
* jPanel.add(bookName);
* jPanel.add(nameField);
* jPanel.add(author);
* jPanel.add(authorField);
* jPanel.add(press);
* jPanel.add(pressField);
* jPanel.add(pressTime);
* jPanel.add(pressTimeField);
* jPanel.add(bookAbstract);
* jPanel.add(introductionField);
* jPanel.add(storage);
* jPanel.add(storageField);
* jPanel.add(remain);
* jPanel.add(remain1);
* jPanel.add(jTextField1);
* jPanel.add(cancel);
* jPanel.add(save);
* jPanel.add(bg);
* c.add(jPanel);
* cancel.addActionListener(this);
* save.addActionListener(this);
* }
* @Override
* public void actionPerformed(ActionEvent e) {
* if (e.getSource() == cancel) {
* this.dispose();
* }
* if (e.getSource() == save) {
* int i = Integer.parseInt(storageField.getText())
* - Integer.parseInt(jTextField.getText())
* + Integer.parseInt(jTextField1.getText());
* BookInfo book = new BookInfo(idField.getText(),
* nameField.getText(), authorField.getText(),
* pressField.getText(), pressTimeField.getText(),
* introductionField.getText(), Integer.parseInt(storageField
* .getText()), i);
* op.inputBook(book);
* JOptionPane.showMessageDialog(null, "Edit the book successfully！",
* "Information", JOptionPane.INFORMATION\_MESSAGE);
* this.dispose();
* }
* }
* private font font=new font();
* private Container c = getContentPane();
* private JPanel jPanel = new JPanel();
* private tool op = new tool();
* private JLabel bookId = new JLabel("Book Id:");
* private JLabel bookName = new JLabel("Book Name:");
* private JLabel author = new JLabel("Book Author:");
* private JLabel press = new JLabel("Press:");
* private JLabel pressTime = new JLabel("Press time:");
* private JLabel bookAbstract = new JLabel("Abstract:");
* private JLabel storage = new JLabel("Storage:");
* private JLabel remain = new JLabel("Remain:");
* private JLabel remain1 = new JLabel("Update with storage");
* private JLabel idField = new JLabel();
* private JTextField nameField = new JTextField();
* private JTextField authorField = new JTextField();
* private JTextField pressField = new JTextField();
* private JTextField pressTimeField = new JTextField();
* private JTextField introductionField = new JTextField();
* private JTextField storageField = new JTextField();
* private JTextField jTextField = new JTextField();
* private JTextField jTextField1 = new JTextField();
* private JLabel bg;
* private ImageIcon img=new ImageIcon("images/Book.jpg");
* private JButton cancel = new JButton("Cancel");
* private JButton save = new JButton("Save");
* }
* **Retrieval module**
* 



**BookRetrieval.java**

package com.company.View;

import com.company.Book.BookDetails;

import com.company.Book.BookInfo;

import com.company.Tool.SQL;

import com.company.Tool.tool;

import com.company.Tool.font;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

//图书检索界面

class BookRetrieval extends JFrame implements ActionListener, SQL {

private tool op = new tool();

public BookRetrieval() {

ob = op.allBook();

for (int i = 0; i < 5; i++) {

DefaultTableModel books = new DefaultTableModel(ob, s);

for (int n = 0; n < 40; n++) {

for (int m = 0; m < 8; m++) {

ob[n][m] = this.ob[n][m];

}

table.setModel(books);

table.invalidate();

}

}

jFrame.add(jPanel2, BorderLayout.NORTH);

jFrame.add(jPanel1, BorderLayout.CENTER);

jFrame.add(jPanel3, BorderLayout.SOUTH);

bookId.setFont(font.JLableFont());

bookName.setFont(font.JLableFont());

id.setFont(font.JTextFiledFont());

name.setFont(font.JTextFiledFont());

search.setFont(font.JBottonFont());

fresh.setFont(font.JBottonFont());

search1.setFont(font.JBottonFont());

jPanel2.setLayout(new GridLayout(2, 4, 20, 10));

jPanel2.add(back);

jPanel2.add(addBook);

jPanel2.add(editBook);

jPanel2.add(deleteBook);

jPanel2.add(borrowBook);

jPanel2.add(returnBook);

jPanel1.setLayout(new GridLayout(1,1,10,10));

jPanel1.add(scrollPane);

jPanel3.setLayout(new GridLayout(7, 1, 0, 0));

jPanel3.add(bookId);

jPanel3.add(id);

jPanel3.add(bookName);

jPanel3.add(name);

jPanel3.add(search);

jPanel3.add(search1);

jPanel3.add(fresh);

table.setPreferredScrollableViewportSize(new Dimension(400, 300));

fresh.addActionListener(this);

addBook.addActionListener(this);

search.addActionListener(this);

search1.addActionListener(this);

back.addActionListener(this);

deleteBook.addActionListener(this);

editBook.addActionListener(this);

borrowBook.addActionListener(this);

returnBook.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == search) {

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.addBookJudgement();

String s1 = id.getText();

String s2 = name.getText();

int n = 0;

int replicate = 0;

while (n < Integer.parseInt(strArray.get(0))) {

n++;

if (s1.equals(strArray.get(n))) {

replicate++;

}

}

ArrayList<String> strArray1 = new ArrayList<String>();

strArray1 = op.addBookJudgement1();

int n1 = 0;

int replicate1 = 0;

while (n1 < Integer.parseInt(strArray1.get(0))) {

n1++;

if (s2.equals(strArray1.get(n1))) {

replicate1++;

}

}

if (replicate == 0 && replicate1 == 0) {

JOptionPane.showMessageDialog(null, "Please input a correct book id or name!", "Warning", JOptionPane.INFORMATION\_MESSAGE);

} else if (replicate != 0 || replicate1 != 0) {

if (replicate == 0 && replicate1 != 0) {

BookInfo book = new BookInfo(s1, s2);

String s = op.searchBookByName(book);

new BookDetails(s);

} else if (replicate != 0) {

new BookDetails(s1);

}

}

}

if(e.getSource()==search1){

new BookTable(id.getText(),name.getText());

}

if (e.getSource() == back) {

this.dispose();

new MainJFrame();

}

if (e.getSource() == addBook) {

new AddBook();

}

if (e.getSource() == deleteBook) {

new DeleteBook();

}

if (e.getSource() == borrowBook) {

new BorrowBook();

}

if (e.getSource() == returnBook) {

new ReturnBook();

}

if (e.getSource() == editBook) {

new SelectEditBook();

}

if(e.getSource()==fresh){

this.dispose();

BookRetrieval page1 = new BookRetrieval();

page1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

page1.setTitle("Book Retrieval");

page1.setLocation(500,300);

page1.setSize(815, 600);

page1.setVisible(true);

}

}

private font font=new font();

private Container jFrame = getContentPane();

private JPanel jPanel1 = new JPanel();

private JPanel jPanel2 = new JPanel();

private JPanel jPanel3 = new JPanel();

private JLabel bookId = new JLabel("Book Id:");

private JLabel bookName = new JLabel("Book Name:");

private JTextField id = new JTextField();

private JTextField name = new JTextField();

private JButton back = new JButton("Back ");

private JButton addBook = new JButton("Add a book");

private JButton deleteBook = new JButton("Delete a book");

private JButton editBook = new JButton("Edit a book");

private JButton search = new JButton("Search for details");

private JButton search1=new JButton("Search");

private JButton fresh=new JButton("Fresh");

private JButton borrowBook = new JButton("Borrow a book");

private JButton returnBook = new JButton("Return a book");

private Object[] s = { "Id", "Name", "Author", "Press", "Press Time",

"Introduction", "Storage", "Remain" };

private Object[][] ob = new Object[200][8];

private JTable table = new JTable(ob, s);

private JScrollPane scrollPane = new JScrollPane(table);

}

**BookSearch.java**

package com.company.View;

import com.company.Book.BookDetails;

import com.company.Book.BookInfo;

import com.company.Tool.tool;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

public class BookSearch extends JFrame implements ActionListener {

public BookSearch(){

bg=new JLabel(img);

this.setTitle("Search Book");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),img.getIconHeight());

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

bookId.setBounds(80,50,150,50);

bookname.setBounds(80,150,150,50);

idField.setBounds(250,50,200,50);

nameField.setBounds(250,150,200,50);

select.setBounds(30,250,200,50);

select1.setBounds(250,250,200,50);

bookId.setFont(font.JLableFont());

idField.setFont(font.JTextFiledFont());

bookname.setFont(font.JLableFont());

nameField.setFont(font.JTextFiledFont());

select.setFont(font.JBottonFont());

select1.setFont(font.JBottonFont());

jPanel.add(bookId);

jPanel.add(idField);

jPanel.add(bookname);

jPanel.add(nameField);

jPanel.add(select1);

jPanel.add(select);

jPanel.add(bg);

container.add(jPanel);

select.addActionListener(this);

select1.addActionListener(this);

}

private font font=new font();

private tool op=new tool();

private Container container = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel bookId = new JLabel("Book Id:");

private JLabel bookname=new JLabel("Book Name:");

private JTextField idField = new JTextField();

private JTextField nameField=new JTextField();

private JButton select1 = new JButton("Select Detail");

private JButton select = new JButton("Select");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/jframe1.jpg");

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource()==select){

BookTable f=new BookTable(idField.getText(),nameField.getText());

f.setVisible(true);

f.setSize(800, 400);

f.setLocationRelativeTo(null);

f.setTitle("Book Table");

}

if(e.getSource()==select1){

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.addBookJudgement();

String s1 = idField.getText();

String s2 = nameField.getText();

int n = 0;

int replicate = 0;

while (n < Integer.parseInt(strArray.get(0))) {

n++;

if (s1.equals(strArray.get(n))) {

replicate++;

}

}

ArrayList<String> strArray1 = new ArrayList<String>();

strArray1 = op.addBookJudgement1();

int n1 = 0;

int replicate1 = 0;

while (n1 < Integer.parseInt(strArray1.get(0))) {

n1++;

if (s2.equals(strArray1.get(n1))) {

replicate1++;

}

}

if (replicate == 0 && replicate1 == 0) {

JOptionPane.showMessageDialog(null, "Please input a correct book id or name!", "Warning", JOptionPane.INFORMATION\_MESSAGE);

} else if (replicate != 0 || replicate1 != 0) {

if (replicate == 0 && replicate1 != 0) {

BookInfo book = new BookInfo(s1, s2);

String s = op.searchBookByName(book);

new BookDetails(s);

} else if (replicate != 0) {

new BookDetails(s1);

}

}

}

}

}

**BookTable.java**

package com.company.View;

import com.company.Tool.tool;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

public class BookTable extends JFrame {

public BookTable(String id,String name){

ob1=op.NameBook(name);

for (int i = 0; i < 5; i++) {

DefaultTableModel books = new DefaultTableModel(ob1, s);

for (int n = 0; n < 20; n++) {

for (int m = 0; m < 8; m++) {

ob1[n][m] = this.ob1[n][m];

}

table.setModel(books);

table.invalidate();

}

}

jPanel1.setLayout(new GridLayout(1,1,10,10));

jPanel1.add(scrollPane);

container.add(jPanel1, BorderLayout.CENTER);

}

private Container container = getContentPane();

private JPanel jPanel1 = new JPanel();

private tool op=new tool();

private Object[] s={"Id","Name","Author","Press","Press Time","Introduction","Storage", "Remain"};

private Object[][] ob1 = new Object[200][8];

private JTable table = new JTable(ob1, s);

private JScrollPane scrollPane = new JScrollPane(table);

}

**StudentTable.java**

package com.company.View;

import com.company.Tool.tool;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

public class StudentTable extends JFrame {

public StudentTable(String id,String name){

ob1=op.NameStudent(name);

for (int i = 0; i < 4; i++) {

DefaultTableModel books = new DefaultTableModel(ob1, s);

for (int n = 0; n < 6; n++) {

for (int m = 0; m < 6; m++) {

ob1[n][m] = this.ob1[n][m];

}

table.setModel(books);

table.invalidate();

}

}

jPanel1.setLayout(new GridLayout(1,1,10,10));

jPanel1.add(scrollPane);

container.add(jPanel1, BorderLayout.CENTER);

}

private Container container = getContentPane();

private JPanel jPanel1 = new JPanel();

private tool op=new tool();

private Object[] s={"Id","Name","Password","Sex","Department","Grade"};

private Object[][] ob1 = new Object[200][6];

private JTable table = new JTable(ob1, s);

private JScrollPane scrollPane = new JScrollPane(table);

}

**SelectEditBook.java**

package com.company.View;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

//编辑图书界面

public class SelectEditBook extends JFrame implements ActionListener {

public SelectEditBook() {

bg=new JLabel(img);

this.setTitle("Borrow Book");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),img.getIconHeight());

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

bookId.setBounds(80,100,150,50);

idField.setBounds(250,100,200,50);

cancel.setBounds(30,200,200,50);

select.setBounds(250,200,200,50);

bookId.setFont(font.JLableFont());

idField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

select.setFont(font.JBottonFont());

jPanel.setLayout(null);

jPanel.add(bookId);

jPanel.add(idField);

jPanel.add(cancel);

jPanel.add(select);

jPanel.add(bg);

container.add(jPanel);

cancel.addActionListener(this);

select.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == select) {

this.dispose();

try {

new EditBook(idField.getText());

} catch (Exception e2) {

JOptionPane.showMessageDialog(null, "Please input the book id correctly!",

"Warning", JOptionPane.INFORMATION\_MESSAGE);

}

}

}

private font font=new font();

private Container container = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel bookId = new JLabel("Book Id:");

private JTextField idField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton select = new JButton("Select");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/jframe1.jpg");

}

**SelectEditStudent.java**

package com.company.View;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

//编辑学生界面

public class SelectEditStudent extends JFrame implements ActionListener {

public SelectEditStudent() {

bg=new JLabel(img);

this.setTitle("Edit Student");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),img.getIconHeight());

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

StudentId.setBounds(80,100,150,50);

IdField.setBounds(250,100,200,50);

cancel.setBounds(250,200,200,50);

select.setBounds(30,200,200,50);

StudentId.setFont(font.JLableFont());

IdField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

select.setFont(font.JBottonFont());

jPanel.add(StudentId);

jPanel.add(IdField);

jPanel.add(cancel);

jPanel.add(select);

jPanel.add(bg);

container.add(jPanel);

cancel.addActionListener(this);

select.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == select) {

this.dispose();

try {

new EditStudent(IdField.getText());

} catch (Exception e2) {

JOptionPane.showMessageDialog(null,

"Please input the student id correctly!", "Warning",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

private font font=new font();

private Container container = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel StudentId = new JLabel("Student Id:");

private JTextField IdField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton select = new JButton("Select");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/jframe1.jpg");

}

**StudentRetrieval.java**

package com.company.View;

import com.company.Student.Studentdetail;

import com.company.Student.StudentInfo;

import com.company.Tool.font;

import com.company.Tool.SQL;

import com.company.Tool.tool;

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

//学生检索界面

class StudentRetrieval extends JFrame implements ActionListener, SQL {

public StudentRetrieval() {

op.allStudent();

ob = op.allStudent();

for (int i = 0; i < 5; i++) {

DefaultTableModel students = new DefaultTableModel(ob, s);

for (int n = 0; n < 40; n++) {

for (int m = 0; m < 5; m++) {

ob[n][m] = this.ob[n][m];

}

table.setModel(students);

table.invalidate();

}

}

container.add(jPanel, BorderLayout.NORTH);

container.add(jPanel1, BorderLayout.CENTER);

container.add(jPanel2, BorderLayout.SOUTH);

jPanel.setLayout(new GridLayout(2, 5, 20, 10));

jPanel.add(back);

jPanel.add(addStudent);

jPanel.add(editStudent);

jPanel.add(deleteStudent);

jPanel1.setLayout(new GridLayout(1,1,10,10));

jPanel1.add(scrollPane);

studentID.setFont(font.JLableFont());

studentName.setFont(font.JLableFont());

id.setFont(font.JTextFiledFont());

name.setFont(font.JTextFiledFont());

search.setFont(font.JBottonFont());

search1.setFont(font.JBottonFont());

fresh.setFont(font.JBottonFont());

jPanel2.setLayout(new GridLayout(7, 1, 0, 0));

jPanel2.add(studentID);

jPanel2.add(id);

jPanel2.add(studentName);

jPanel2.add(name);

jPanel2.add(search);

jPanel2.add(search1);

jPanel2.add(fresh);

fresh.addActionListener(this);

addStudent.addActionListener(this);

search.addActionListener(this);

search1.addActionListener(this);

back.addActionListener(this);

deleteStudent.addActionListener(this);

editStudent.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

//搜索事件

if (e.getSource() == search) {

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.addStudentJudgement();

String s1 = id.getText();

String s2 = name.getText();

int n = 0;

int replicate = 0;

while (n < Integer.parseInt(strArray.get(0))) {

n++;

if (s1.equals(strArray.get(n))) {

replicate++;

}

}

ArrayList<String> strArray1 = new ArrayList<String>();

strArray1 = op.addStudentJudgement1();

int n1 = 0;

int replicate1 = 0;

while (n1 < Integer.parseInt(strArray1.get(0))) {

n1++;

if (s2.equals(strArray1.get(n1))) {

replicate1++;

}

}

if (replicate == 0 && replicate1 == 0) {

JOptionPane.showMessageDialog(null,

"Please input a correct student id or name!",

"Warning", JOptionPane.INFORMATION\_MESSAGE);

} else if (replicate != 0 || replicate1 != 0) {

if (replicate == 0 && replicate1 != 0) {

StudentInfo student = new StudentInfo(s1, s2);

String s=op.searchStudentByName(student);

new Studentdetail(s);

} else if (replicate != 0) {

new Studentdetail(s1);

}

}

}

if(e.getSource()==search1) {

StudentTable f = new StudentTable(id.getText(), name.getText());

f.setVisible(true);

f.setSize(600, 400);

f.setLocationRelativeTo(null);

f.setTitle("Student Table");

}

//返回时发生的事件

if (e.getSource() == back) {

this.dispose();

new MainJFrame();

}

//添加学生事件

if (e.getSource() == addStudent) {

new AddStudent();

}

//删除学生事件

if (e.getSource() == deleteStudent) {

new DeleteStudent();

}

//编辑学生事件

if (e.getSource() == editStudent) {

new SelectEditStudent();

}

//刷新事件

if(e.getSource()==fresh){

this.dispose();

StudentRetrieval page2 = new StudentRetrieval();

page2.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

page2.setTitle("Student Retrieval");

page2.setLocation(500,400);

page2.setSize(615, 600);

page2.setVisible(true);

}

}

private font font=new font();

private tool op = new tool();

private Container container = getContentPane();

private JPanel jPanel1 = new JPanel();

private JPanel jPanel = new JPanel();

private JPanel jPanel2 = new JPanel();

private JLabel studentID = new JLabel("Student Id:");

private JLabel studentName = new JLabel("Student Name:");

private JTextField id= new JTextField();

private JTextField name = new JTextField();

private JButton back = new JButton("Back");

private JButton addStudent = new JButton("Add a student");

private JButton deleteStudent = new JButton("Delete a student");

private JButton editStudent = new JButton("Edit a student");

private JButton fresh =new JButton("Fresh");

private JButton search1=new JButton("Search");

private JButton search = new JButton("Search for details");

private Object[] s = { "Id", "Name","Password","Sex", "Department", "Grade" };

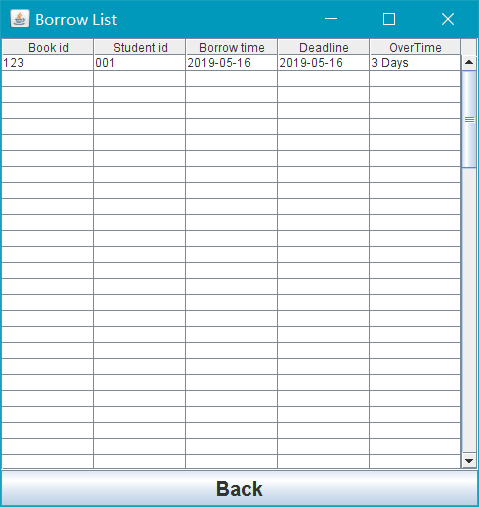
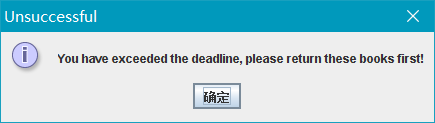
private Object[][] ob = new Object[200][6];

private JTable table = new JTable(ob, s);

private JScrollPane scrollPane = new JScrollPane(table);

}

**Time management module**



**Date.java**

package com.company.Tool;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

public class Date {

//转换日期的字符串为日期格式

static String borrowTimeInterface(long time) {

SimpleDateFormat df = (SimpleDateFormat) DateFormat.getInstance();

df.applyPattern("yyyy-MM-dd");

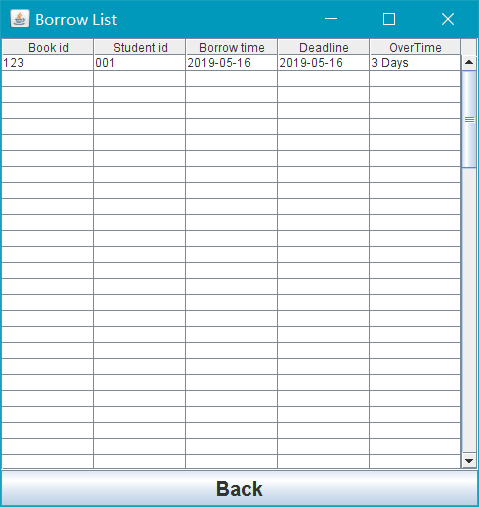
String s = df.format(time);

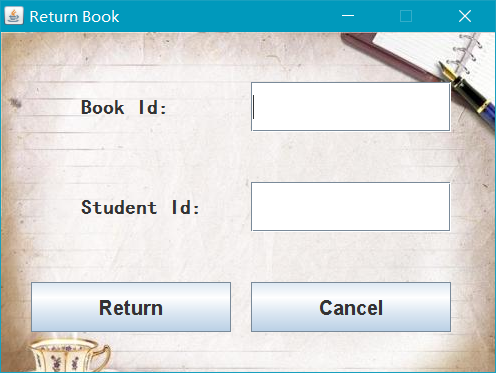
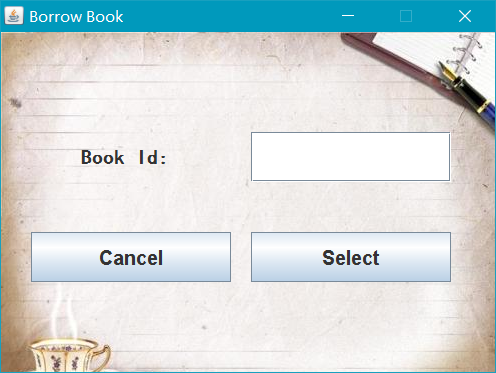
return s;

}

}

**Book loan and return module**





**BorrowBook.java**

package com.company.View;

import com.company.Student.StudentInfo;

import com.company.Tool.font;

import com.company.Book.BookInfo;

import com.company.Tool.tool;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

//借书界面

public class BorrowBook extends JFrame implements ActionListener {

tool tool = new tool();

public BorrowBook() {

bg=new JLabel(img);

this.setTitle("Borrow Book");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),img.getIconHeight());

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

BookId.setBounds(80,50,150,50);

StudentId.setBounds(80,150,150,50);

idField.setBounds(250,50,200,50);

StudentIdField.setBounds(250,150,200,50);

cancel.setBounds(30,250,200,50);

borrow.setBounds(250,250,200,50);

BookId.setFont(font.JLableFont());

StudentId.setFont(font.JLableFont());

idField.setFont(font.JTextFiledFont());

StudentIdField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

borrow.setFont(font.JBottonFont());

jPanel.add(BookId);

jPanel.add(idField);

jPanel.add(StudentId);

jPanel.add(StudentIdField);

jPanel.add(cancel);

jPanel.add(borrow);

jPanel.add(bg);

c.add(jPanel);

cancel.addActionListener(this);

borrow.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == borrow) {

String bookId = idField.getText();

ArrayList<String> strArray = new ArrayList<String>();

strArray = tool.addBookJudgement();

int n = 0;

int replicate = 0;

while (n < Integer.parseInt(strArray.get(0))) {

n++;

if (bookId.equals(strArray.get(n))) {

replicate++;

}

}

String StudentId = StudentIdField.getText();

ArrayList<String> strArray1 = new ArrayList<String>();

strArray1 = tool.addStudentJudgement();

int n1 = 0;

int replicate1 = 0;

while (n1 < Integer.parseInt(strArray1.get(0))) {

n1++;

if (StudentId.equals(strArray1.get(n1))) {

replicate1++;

}

}

if (replicate == 0 || replicate1 == 0) {

JOptionPane

.showMessageDialog(

null,

"Please input a correct book id and a student id!",

"Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

BookInfo book = new BookInfo(bookId);

ArrayList<String> s = new ArrayList<String>();

s = tool.outputBook(book);

if (Integer.parseInt(s.get(7)) > 0) //判断库存总量是否大于一本

{

int s1 = tool.reBorrowCheck(idField.getText(),

StudentIdField.getText());

if (s1 == 1) //判断是否借过本书

{

JOptionPane

.showMessageDialog(

null,

"You have borrowed this book, can not reborrow it!",

"Unsuccessful",

JOptionPane.INFORMATION\_MESSAGE);

} else {

long currentTime = System.currentTimeMillis();

if (tool.deadLineCheck(StudentIdField.getText(),

currentTime) != 0) {

JOptionPane

.showMessageDialog(

null,

"You have exceeded the deadline, please return these books first!",

"Unsuccessful",

JOptionPane.INFORMATION\_MESSAGE);

} else {

BookInfo book1 = new BookInfo(s.get(0), s.get(1),

s.get(2), s.get(3), s.get(4), s.get(5),Integer.parseInt(s.get(6)),

Integer.parseInt(s.get(7))-1)

;

tool.inputBook(book1);

String borrowTime = Long.toString(System

.currentTimeMillis());

String deadline = Long.toString(System

.currentTimeMillis() + 2592000000l);

tool.insertBorrow(idField.getText(),

StudentIdField.getText(), borrowTime,

deadline);

JOptionPane

.showMessageDialog(

null,

"Borrowed this book successfully, you have 30 days to enjoy this book!",

"Successful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

} else {

JOptionPane.showMessageDialog(null,

"This book has been borrowed!", "Unsuccessful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

this.dispose();

}

}

private font font=new font();

private Container c = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel BookId = new JLabel("Book Id:");

private JTextField idField = new JTextField();

private JLabel StudentId = new JLabel("Student Id:");

private JTextField StudentIdField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton borrow = new JButton("Borrow");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/jframe1.jpg");

}

**ReturnBook.java**

package com.company.View;

import com.company.Book.BookInfo;

import com.company.Tool.tool;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

//还书界面

public class ReturnBook extends JFrame implements ActionListener {

public ReturnBook() {

c.add(jPanel);

bg=new JLabel(img);

this.setTitle("Return Book");

this.setLocation(300,200);

this.setSize(img.getIconWidth(),img.getIconHeight());

this.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

this.setVisible(true);

this.setResizable(false);

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

bookId.setBounds(80,50,150,50);

studentId.setBounds(80,150,150,50);

idField.setBounds(250,50,200,50);

studentIdField.setBounds(250,150,200,50);

returnBook.setBounds(30,250,200,50);

cancel.setBounds(250,250,200,50);

bookId.setFont(font.JLableFont());

idField.setFont(font.JTextFiledFont());

studentId.setFont(font.JLableFont());

studentIdField.setFont(font.JTextFiledFont());

cancel.setFont(font.JBottonFont());

returnBook.setFont(font.JBottonFont());

jPanel.add(bookId);

jPanel.add(idField);

jPanel.add(studentId);

jPanel.add(studentIdField);

jPanel.add(cancel);

jPanel.add(returnBook);

jPanel.add(bg);

cancel.addActionListener(this);

returnBook.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == cancel) {

this.dispose();

}

if (e.getSource() == returnBook) {

String bookId = idField.getText();

ArrayList<String> strArray = new ArrayList<String>();

strArray = op.addBookJudgement();

int n = 0;

int replicate = 0;

while (n < Integer.parseInt(strArray.get(0))) {

n++;

if (bookId.equals(strArray.get(n))) {

replicate++;

}

}

String studentId = studentIdField.getText();

ArrayList<String> strArray1 = new ArrayList<String>();

strArray1 = op.addStudentJudgement();

int n1 = 0;

int replicate1 = 0;

while (n1 < Integer.parseInt(strArray1.get(0))) {

n1++;

if (studentId.equals(strArray1.get(n1))) {

replicate1++;

}

}

if (replicate == 0 || replicate1 == 0) {

JOptionPane.showMessageDialog(null, "Please input a correct book id and a student id!", "Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

if (op.returnCheck(idField.getText(),

studentIdField.getText()) == 1) {

JOptionPane.showMessageDialog(null, "No result found!", "Warning", JOptionPane.INFORMATION\_MESSAGE);

} else {

BookInfo book = new BookInfo(bookId);

ArrayList<String> s = new ArrayList<String>();

s = op.outputBook(book);

BookInfo book1 = new BookInfo(s.get(0), s.get(1),

s.get(2), s.get(3), s.get(4), s.get(5),

Integer.parseInt(s.get(6)), Integer.parseInt(s.get(7)) + 1);

op.inputBook(book1);

if(op.deleteResult(idField.getText(), studentIdField.getText())==1){

JOptionPane.showMessageDialog(null, "Return the book successful!", "Successful", JOptionPane.INFORMATION\_MESSAGE);

this.dispose();

}else{

JOptionPane.showMessageDialog(null, "Failed to return the book!", "Unsuccessful", JOptionPane.INFORMATION\_MESSAGE);

}

}

}

}

}

private font font=new font();

private tool op = new tool();

private Container c = getContentPane();

private JPanel jPanel = new JPanel();

private JLabel bookId = new JLabel("Book Id:");

private JTextField idField = new JTextField();

private JLabel studentId = new JLabel("Student Id:");

private JTextField studentIdField = new JTextField();

private JButton cancel = new JButton("Cancel");

private JButton returnBook = new JButton("Return");

private JLabel bg;

private ImageIcon img=new ImageIcon("images/jframe1.jpg");

}

**tool.java**

package com.company.Tool;

import com.company.Student.StudentInfo;

import com.company.Book.BookInfo;

import java.sql.\*;

import java.util.ArrayList;

public class tool implements SQL{

static PreparedStatement preparedStatement=null;

static Connection connection=null;

static ResultSet resultSet=null;

//加载驱动+连接数据库

public tool() {

try {

Class.forName(Driver);

connection = DriverManager.getConnection(Url, User, Password);

} catch (Exception e) {

e.printStackTrace();

}

}

//检测管理员登录

public ResultSet mglogCheck(String name, String password) throws SQLException {

try {

PreparedStatement ps = connection.prepareStatement(sql1);

ps.setString(1,name);

ps.setString(2,password);

resultSet = ps.executeQuery();

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

//检测学生登录

public ResultSet stlogCheck(String id,String password) throws SQLException {

try {

PreparedStatement ps = connection.prepareStatement(sql27);

ps.setString(1,id);

ps.setString(2,password);

resultSet = ps.executeQuery();

} catch (Exception e) {

e.printStackTrace();

}

return resultSet;

}

//返回包含所有图书编号的数组

public ArrayList addBookJudgement() {

ArrayList<String> strArray = new ArrayList<String>();

ArrayList<String> strArray1 = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql2);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("bkid");

strArray.add(s);

count++;

}

strArray.add(Integer.toString(count));

for (; count >= 0; count--) {

strArray1.add(strArray.get(count));

}

} catch (Exception e) {

e.printStackTrace();

}

return strArray1;

}

//返回包含所有书名的数组

public ArrayList addBookJudgement1() {

ArrayList<String> strArray = new ArrayList<String>();

ArrayList<String> strArray1 = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql3);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("bkname");

strArray.add(s);

count++;

}

strArray.add(Integer.toString(count));

for (; count >= 0; count--) {

strArray1.add(strArray.get(count));

}

} catch (Exception e) {

e.printStackTrace();

}

return strArray1;

}

//返回包含所有学生编号的数组

public ArrayList addStudentJudgement() {

ArrayList<String> strArray = new ArrayList<String>();

ArrayList<String> strArray1 = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql4);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("stid");

strArray.add(s);

count++;

}

strArray.add(Integer.toString(count));

for (; count >= 0; count--) {

strArray1.add(strArray.get(count));

}

} catch (Exception e) {

e.printStackTrace();

}

return strArray1;

}

//返回包含所有学生姓名的数组

public ArrayList addStudentJudgement1() {

ArrayList<String> strArray = new ArrayList<String>();

ArrayList<String> strArray1 = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql5);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("stname");

strArray.add(s);

count++;

}

strArray.add(Integer.toString(count));

for (; count >= 0; count--) {

strArray1.add(strArray.get(count));

}

} catch (Exception e) {

e.printStackTrace();

}

return strArray1;

}

//增加学生

public void addStudent(StudentInfo student) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql6);

preparedStatement.setString(1, student.getID());

preparedStatement.setString(2, student.getName());

preparedStatement.setString(3,student.getPassword());

preparedStatement.setString(4, student.getSex());

preparedStatement.setString(5, student.getDpt());

preparedStatement.setInt(6, student.getGrade());

preparedStatement.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

//增加图书

public void addBook(BookInfo book) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql7);

preparedStatement.setString(1, book.getId());

preparedStatement.setString(2, book.getName());

preparedStatement.setString(3, book.getAuthor());

preparedStatement.setString(4, book.getPress());

preparedStatement.setString(5, book.getPressTime());

preparedStatement.setString(6, book.getBookIntroduction());

preparedStatement.setInt(7, book.getTotal());

preparedStatement.setInt(8, book.getRemain());

preparedStatement.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

//返回对应书名的图书编号

public String searchBookByName(BookInfo book) {

String s = null;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql8);

preparedStatement.setString(1, book.getName());

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

s = resultSet.getString("bkid");

}

} catch (Exception e) {

e.printStackTrace();

}

return s;

}

//返回对应的学生名字的学生编号

public String searchStudentByName(StudentInfo student) {

String s = null;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql9);

preparedStatement.setString(1, student.getName());

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

s = resultSet.getString("stid");

}

} catch (Exception e) {

e.printStackTrace();

}

return s;

}

//删除图书

public int deleteBook(BookInfo book) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql10);

preparedStatement.setString(1, book.getId());

int count = preparedStatement.executeUpdate();

if (count == 1) {

return 1;

} else {

return 0;

}

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

//删除学生

public int deleteStudent(StudentInfo student) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql11);

preparedStatement.setString(1, student.getID());

int count = preparedStatement.executeUpdate();

if (count == 1) {

return 1;

} else {

return 0;

}

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

//返回包含对应图书编号的所有图书信息的数组

public ArrayList outputBook(BookInfo book) {

ArrayList<String> strArray = new ArrayList<String>();

try {

String s=null;

PreparedStatement preparedStatement = connection.prepareStatement(sql12);

preparedStatement.setString(1, book.getId());

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

s = resultSet.getString("bkid");

strArray.add(s);

s = resultSet.getString("bkname");

strArray.add(s);

s = resultSet.getString("author");

strArray.add(s);

s = resultSet.getString("press");

strArray.add(s);

s = resultSet.getString("bkdate");

strArray.add(s);

s = resultSet.getString("introduction");

strArray.add(s);

s = resultSet.getString("sum");

strArray.add(s);

s = resultSet.getString("num");

strArray.add(s);

}

} catch (Exception e) {

e.printStackTrace();

}

return strArray;

}

//返回对应学生编号的所有学生信息的数组

public ArrayList outputStudent(StudentInfo student) {

ArrayList<String> strArray = new ArrayList<String>();

try {

String s=null;

PreparedStatement preparedStatement = connection.prepareStatement(sql13);

preparedStatement.setString(1, student.getID());

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

s = resultSet.getString("stid");

strArray.add(s);

s = resultSet.getString("stname");

strArray.add(s);

s= resultSet.getString("stpassword");

strArray.add(s);

s = resultSet.getString("sex");

strArray.add(s);

s = resultSet.getString("department");

strArray.add(s);

s = resultSet.getString("grade");

strArray.add(s);

}

} catch (Exception e) {

e.printStackTrace();

}

return strArray;

}

//更新图书信息

public void inputBook(BookInfo book) {

ArrayList<String> strArray = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql14);

preparedStatement.setString(1, book.getName());

preparedStatement.setString(2, book.getAuthor());

preparedStatement.setString(3, book.getPress());

preparedStatement.setString(4, book.getPressTime());

preparedStatement.setString(5, book.getBookIntroduction());

preparedStatement.setInt(6, book.getTotal());

preparedStatement.setInt(7, book.getRemain());

preparedStatement.setString(8, book.getId());

preparedStatement.executeUpdate();

} catch (Exception e) {

System.out.println(e);

}

}

//更新学生信息

public void inputStudent(StudentInfo student) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql15);

preparedStatement.setString(1, student.getName());

preparedStatement.setString(2,student.getPassword());

preparedStatement.setString(3, student.getSex());

preparedStatement.setString(4, student.getDpt());

preparedStatement.setInt(5, student.getGrade());

preparedStatement.setString(6, student.getID());

preparedStatement.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

//返回图书详细信息的表格

public Object[][] allBook() {

Object[][] ob = new Object[60][8];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql16);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("bkid");

ob[n][1] = resultSet.getString("bkname");

ob[n][2] = resultSet.getString("author");

ob[n][3] = resultSet.getString("press");

ob[n][4] = resultSet.getString("bkdate");

ob[n][5] = resultSet.getString("introduction");

ob[n][6] = resultSet.getString("sum");

ob[n][7] = resultSet.getString("num");

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//返回对应名字图书的详细信息表格

public Object[][] NameBook(String bookName) {

Object[][] ob = new Object[60][8];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql26);

preparedStatement.setString(1,bookName);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("bkid");

ob[n][1] = resultSet.getString("bkname");

ob[n][2] = resultSet.getString("author");

ob[n][3] = resultSet.getString("press");

ob[n][4] = resultSet.getString("bkdate");

ob[n][5] = resultSet.getString("introduction");

ob[n][6] = resultSet.getString("sum");

ob[n][7] = resultSet.getString("num");

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//返回学生详细信息的表格

public Object[][] allStudent() {

Object[][] ob = new Object[60][6];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql17);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("stid");

ob[n][1] = resultSet.getString("stname");

ob[n][2]= resultSet.getString("stpassword");

ob[n][3] = resultSet.getString("sex");

ob[n][4] = resultSet.getString("department");

ob[n][5] = resultSet.getString("grade");

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//返回对应姓名的学生详细信息表格

public Object[][] NameStudent(String studentName) {

Object[][] ob = new Object[60][6];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql25);

preparedStatement.setString(1,studentName);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("stid");

ob[n][1] = resultSet.getString("stname");

ob[n][2]= resultSet.getString("stpassword");

ob[n][3] = resultSet.getString("sex");

ob[n][4] = resultSet.getString("department");

ob[n][5] = resultSet.getString("grade");

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//借阅时候往数据库library的borrow表添加数据

public void insertBorrow(String bookId, String studentId, String borrowTime, String deadline) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql18);

preparedStatement.setString(1, bookId);

preparedStatement.setString(2, studentId);

preparedStatement.setString(3, borrowTime);

preparedStatement.setString(4, deadline);

preparedStatement.executeUpdate();

} catch (Exception e) {

e.printStackTrace();

}

}

//查询图书借阅情况

public int reBorrowCheck(String bookId, String studentId) {

int i = 0;

ArrayList<String> strArray = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql19);

preparedStatement.setString(1, bookId);

preparedStatement.setString(2, studentId);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("bkid");

strArray.add(s);

s = resultSet.getString("stid");

strArray.add(s);

count++;

}

System.out.println(strArray);

if (count >= 1) {

i = 1;

}

} catch (Exception e) {

e.printStackTrace();

}

return i;

}

//查询图书归还日期

public int deadLineCheck(String StudentId, long currentTime) {

int i = 0;

ArrayList<String> strArray = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql20);

preparedStatement.setString(1, StudentId);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("redate");

strArray.add(s);

count++;

}

for (; count > 0; count--) {

if (Long.parseLong(strArray.get(count - 1)) <= currentTime) {

i++;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return i;

}

//查询图书归还情况

public int returnCheck(String bookId, String studentId) {

int i = 0;

ArrayList<String> strArray = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql21);

preparedStatement.setString(1, bookId);

preparedStatement.setString(2, studentId);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

String s = resultSet.getString("bkid");

strArray.add(s);

s = resultSet.getString("stid");

strArray.add(s);

}

System.out.println(strArray);

if (strArray.get(0) != bookId

|| strArray.get(1) != studentId) {

i = 0;

}

} catch (Exception e) {

e.printStackTrace();

i = 1;

}

return i;

}

//删除学生

public int deleteResult(String bookNumber, String studentId) {

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql22);

preparedStatement.setString(1, bookNumber);

preparedStatement.setString(2, studentId);

int count = preparedStatement.executeUpdate();

if (count == 1) {

return 1;

} else {

return 0;

}

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

//创建借阅表

public Object[][] borrowList() {

Date dateCal = new Date();

Object[][] ob = new Object[100][5];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql23);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("bkid");

ob[n][1] = resultSet.getString("stid");

String s = dateCal.borrowTimeInterface(Long.parseLong(resultSet

.getString("brdate")));

ob[n][2] = s;

long re = Long.parseLong(resultSet.getString("redate"));

s = dateCal.borrowTimeInterface(re);

ob[n][3] = s;

long re1 = System.currentTimeMillis();

if (re1 > re) {

ob[n][4] = Long.toString(((re1 - re) / 86400000)+1) + " Days";

} else {

ob[n][4] = "NOT";

}

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//返回学生的借阅表（显示学生借阅信息）

public Object[][] borrowListForStudent(String studentId) {

Date dateCal = new Date();

Object[][] ob = new Object[100][4];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql20);

preparedStatement.setString(1, studentId);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("bkid");

String s = dateCal.borrowTimeInterface(Long.parseLong(resultSet

.getString("brdate")));

ob[n][1] = s;

long re = Long.parseLong(resultSet.getString("redate"));

s = dateCal.borrowTimeInterface(re);

ob[n][2] = s;

long re1 = System.currentTimeMillis();

if (re1 > re) {

ob[n][3] = Long.toString(((re1 - re) / 86400000)+1) + " Days";

} else {

ob[n][3] = "NOT";

}

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//创建图书借阅表（显示图书的借阅情况）

public Object[][] borrowListForBook(String bookId) {

Date dateCal = new Date();

Object[][] ob = new Object[100][8];

int n = 0;

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql24);

preparedStatement.setString(1, bookId);

resultSet = preparedStatement.executeQuery();

while (resultSet.next()) {

ob[n][0] = resultSet.getString("stid");

String s = dateCal.borrowTimeInterface(Long.parseLong(resultSet

.getString("brdate")));

ob[n][1] = s;

long re = Long.parseLong(resultSet.getString("redate"));

s = dateCal.borrowTimeInterface(re);

ob[n][2] = s;

long re1 = System.currentTimeMillis();

if (re1 > re) {

ob[n][3] = Long.toString(((re1 - re) / 86400000)+1) + " Days";

} else {

ob[n][3] = "NOT";

}

n++;

}

} catch (Exception e) {

e.printStackTrace();

}

return ob;

}

//返回借阅表的对应图书信息

public int deleteBookCheck(String bookId) {

int i = 0;

ArrayList<String> strArray = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql23);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("bkid");

strArray.add(s);

count++;

}

for (; count > 0; count--) {

if (strArray.get(count - 1).equals(bookId)) {

i++;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return i;

}

//返回借阅表的对应学生信息

public int deleteStudentCheck(String studentId) {

int i = 0;

ArrayList<String> strArray = new ArrayList<String>();

try {

PreparedStatement preparedStatement = connection.prepareStatement(sql23);

resultSet = preparedStatement.executeQuery();

int count = 0;

while (resultSet.next()) {

String s = resultSet.getString("stid");

strArray.add(s);

count++;

}

for (; count > 0; count--) {

if (strArray.get(count - 1).equals(studentId)) {

i++;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return i;

}

}

**Home.java**

package com.company.View;

import com.company.Tool.font;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

//程序主界面

public class Home {

}

class MainJFrame extends JFrame implements ActionListener {

public MainJFrame() {

bg=new JLabel(img);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

this.setTitle("Library Management System");

this.setLocation(430,350);

this.setResizable(false);

this.setVisible(true);

this.setSize(img.getIconWidth(),img.getIconHeight());

jPanel.setLayout(null);

bg.setBounds(0,0,img.getIconWidth(),img.getIconHeight());

book.setBounds(180,50,400,80);

student.setBounds(180,180,400,80);

list.setBounds(180,300,400,80);

back.setBounds(180,420,400,80);

jPanel.add(student);

jPanel.add(list);

jPanel.add(back);

jPanel.add(book);

jPanel.add(bg);

container.add(jPanel);

welcome.setFont(font.JBottonFont());

book.setFont(font.JBottonFont());

student.setFont(font.JBottonFont());

list.setFont(font.JBottonFont());

back.setFont(font.JBottonFont());

list.addActionListener(this);

book.addActionListener(this);

student.addActionListener(this);

back.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == book) {

this.dispose();

BookRetrieval page1 = new BookRetrieval();

page1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

page1.setTitle("Book Retrieval");

page1.setLocation(500,300);

page1.setSize(815, 600);

page1.setVisible(true);

}

if (e.getSource() == student) {

this.dispose();

StudentRetrieval page2 = new StudentRetrieval();

page2.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

page2.setTitle("Student Retrieval");

page2.setLocation(500,400);

page2.setSize(615, 600);

page2.setVisible(true);

}

if (e.getSource() == list) {

this.dispose();

BorrowList page2 = new BorrowList();

page2.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

page2.setTitle("Borrow List");

page2.setLocation(300,200);

page2.setSize(495, 515);

page2.setVisible(true);

}

if(e.getSource()==back){

this.dispose();

new Main();

}

}

private JLabel bg;

private ImageIcon img=new ImageIcon("images/Manger.jpg");

private com.company.Tool.font font=new com.company.Tool.font();

private Container container = getContentPane();

private JPanel jPanel=new JPanel();

private JLabel welcome = new JLabel("Welcome to my library management system");

private JButton book = new JButton("Book Retrieval");

private JButton student = new JButton("Student Retrieval");

private JButton list = new JButton("Borrow List");

private JButton back=new JButton("Back");

}

**font.java**

package com.company.Tool;

import java.awt.\*;

public class font {

//设置标题字体

public Font Title(){

Font font=new Font("楷体",1,30);

return font;

}

//设置标签的字体

public Font JLableFont(){

Font font=new Font("楷体",1,20);

return font;

}

//设置按钮的字体

public Font JBottonFont(){

Font font=new Font("00",1,20);

return font;

}

//设置文本的字体

public Font JTextFiledFont(){

Font font=new Font("宋体",1,20);

return font;

}

}

**SQL**

package com.company.Tool;

public interface SQL {

String Url="jdbc:mysql://localhost:3306/library?useSSL=false&useUnicode=true&characterEncoding=utf8";

String User="root";//数据库用户名

String Password="1234";//数据库密码

String Driver ="com.mysql.jdbc.Driver";//驱动

//SQL功能语句

String sql1="select \* from manger where mgname=? and mgpassword=?;";

String sql2 = "select bkid FROM book;";

String sql3 = "select bkname FROM book;";

String sql4 = "select stid From student;";

String sql5 = "select stname FROM student;";

String sql6 = "iNSERT INTO student(stid,stname,stpassword,sex,department,grade) VALUES (?,?,?,?,?,?);";

String sql7 = "iNSERT INTO book(bkid,bkname,author,press,bkdate,introduction,sum,num) VALUES (?,?,?,?,?,?,?,?);";

String sql8 = "select bkid FROM book where bkname=?;";

String sql9 = "select stid FROM student where stname=?;";

String sql10 = "delete FROM book where bkid=?;";

String sql11 = "delete FROM student where stid=?;";

String sql12 = "select \* FROM book where bkid=?;";

String sql13 = "select \* FROM student where stid=?;";

String sql14 = "update book set bkname=?,author=?,press=?,bkdate=?,introduction=?,sum=?,num=? where bkid=?;";

String sql15 = "update student set stname=?,stpassword=?,sex=?,department=?,grade=? where stid=?;";

String sql16 = "select \* FROM book;";

String sql17 = "select \* FROM student;";

String sql18 = "insert Into borrow ( bkid , stid,brdate, redate) values(?,?,?,?)";

String sql19 = "select \* FROM borrow where bkid=? and stid=?;";

String sql20 = "select \* FROM borrow where stid=?;";

String sql21 = "select \* FROM borrow where bkid=? and stid=?;";

String sql22 = "delete FROM borrow where bkid=? and stid=?;";

String sql23 = "select \* FROM borrow;";

String sql24 = "select \* FROM borrow where bkid=?;";

String sql25 = "select \* From student where stname=?;";

String sql26 = "select \* From book where bkname=?;";

String sql27 = "select \* from student where stid=? and stpassword=?;";

}