Southwest University

Course Design

Couse Name Fundamentals of Database Systems

Semester 2019 - 2020- 2

Grade 2018 Class CS4

Student Name 陈志宏 邢嘉路 曾令瀚

Student No 222018321102100 陈志宏

222018321102112 邢嘉路

222018321102115 曾令瀚

Tutor Quan Zou

School of Computer and Information Science

目录

[1 Roles of labor 3](#_Toc43653339)

[2 Overview 3](#_Toc43653340)

[3 Design Process 4](#_Toc43653341)

[4 Brief instructions for use 10](#_Toc43653342)

[5 Main interface screenshots 11](#_Toc43653343)

[6 Summary 11](#_Toc43653344)

[7 References 12](#_Toc43653345)

### Roles of labor

邢嘉路：The general analysis of the database; Partial needs analysis of the program; Database code preparation; Implementation of database connection; The layout design and master plan of the paper.

陈志宏：The analysis and design of database; The specific design of the database; Troubleshoot and verify the database code; Writing of Java code of some programs; Writing of specific content of the paper.

曾令瀚：The analysis and design of the database; Partial needs analysis of the program; Database code preparation; Program interface design and code writing; Writing of specific content of the paper.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Members | Database Analysis | Database Design | Requirement analysis and function modules | Code implementation | Document writing | Interface design |
| 邢嘉路 | 40% | 33.33% | %20 | %20 | %60 | %20 |
| 陈志宏 | 40% | 33.33% | %30 | %40 | %20 | %40 |
| 曾令瀚 | 20% | 33.33% | %50 | %40 | %20 | %40 |

### Overview

This article introduces the student epidemic situation management system developed by our group. This system uses MySQL as the back-end database and designs the project according to the Model View Controller structure. The epidemic situation management system is mainly used for the management of the collection of student health information. The main task is to use a computer to carry out daily management of various information such as student basic information and health status, such as querying student information, inserting information, and deleting redundant information. This system realizes the information management of health status collection and completes the basic functions of the student epidemic situation management system.

The student's outbreak management system is a database management system using Eclipse development software, Mysql-workbench and navicat as database visualization development. The basic steps to develop this system are: problem definition, feasibility study, requirement analysis, overall design, detailed design, and software testing. System flowchart, data flow diagram, data dictionary, state transition diagram, specifications, hierarchy diagram, structure diagram and program flowchart are used in system analysis. In the system design process, the MVC three-tier architecture mode is adopted. Each module is organized in a more detailed and organized manner, and modules required by multiple classes are encapsulated into public classes to reduce code redundancy. In the design of the data table of the database library, strive to design simple and practical. In the specific implementation stage of the system, try to make the window simple and beautiful.

### Design Process

#### 3.1 Problem definition

**The general definition of the problem:**

This software should realize the insertion, deletion, modification of students' health status information, query of students' basic information and update of students' status.

**System specific requirements:**

The interface should be simple and beautiful

Use MySQL database

Use Java language

The software supports complete business processes

Own sufficient fault tolerance and robustness

#### 3.2 Availability

##### 3.2.1 General concept

Designed with Java + MySQL and designed with Java AWT. Display student information through the interaction of the table component and the database. Insert, delete, and modify database operations through the interaction of text boxes and buttons with tables and databases.

##### 3.2.2Flow chart

To describe the physical system and express the flow of data between the components of the system.

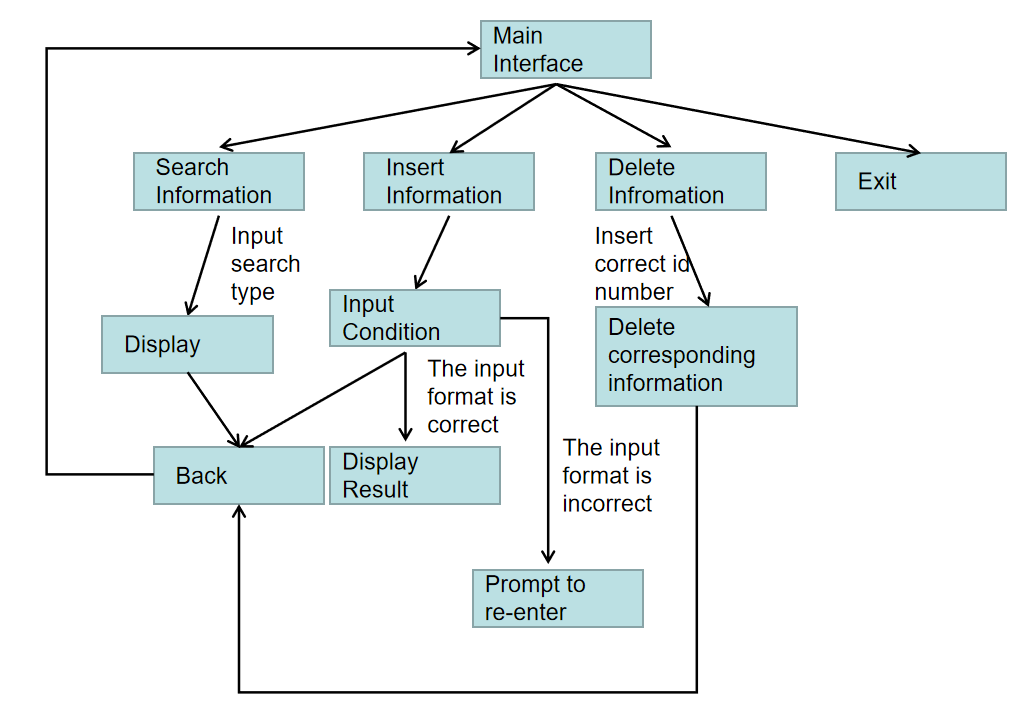


Figure 1.1

##### 3.2.3Data flow diagram

**depicts the transformation of information flow and data as it moves from input to output**

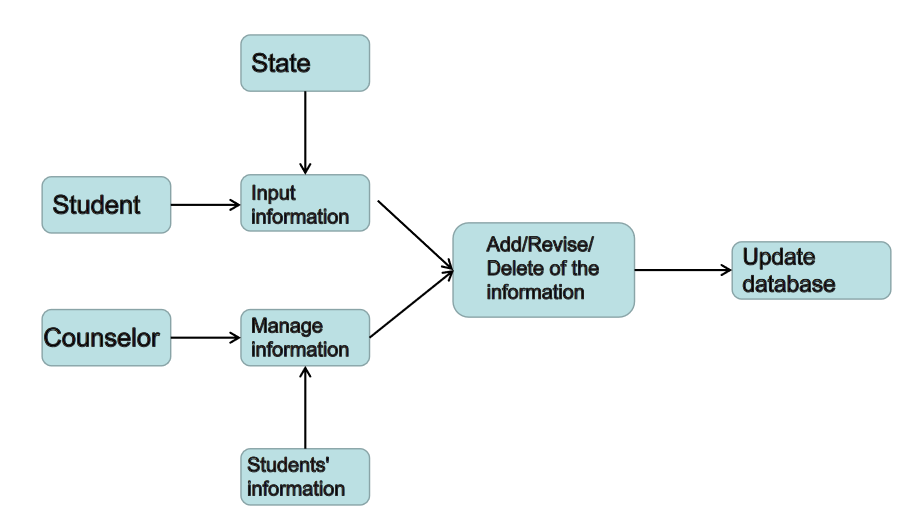


Figure 1.2

##### 3.2.4Data dictionary

**The explanation of the logical model of the system.**

Name: Students’ information

Description: The basic information of students. Including Id, Sex, College, Phone number, Native Place, Age, State.

Position: Database + Transaction

Name: Health Status Information

Description: The set of all the health status of students everyday. Including Id, Answer Date, Region, Touch History and The Answers.

Position: Database + Transaction

Name: Administrators’ information

Description: The basic information of Consolers. Including Id, Sex, Phone number, Department.

Position: Database

Name: Name

Description: The Set of all names of Students and Consolers. Including Id and Name.

Position: Database

Name: Question

Description: The Set of all questions. Including Id and question.

Position: Database

#### 3.3 Database Design

##### 3.3.1ER Diagram：

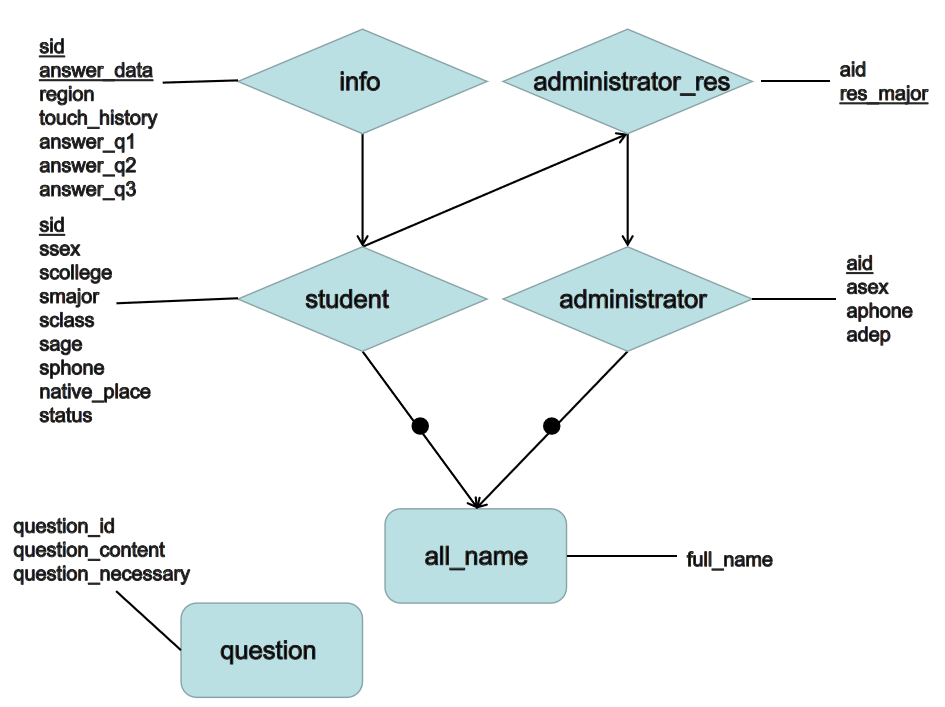


Figure 1.3

all\_name: Name

student: Students’ information

administrator: Administrators’ information

administrator\_res: Administrators’ responsible major information

info：Health Status Information

question: Question

Based on the consideration of the function and data volume of our database, we believe that the 2NF is most suitable for our current database relationship model. We think that such a database model better balances the problems of data redundancy and table complexity

##### 3.3.2Strcture:

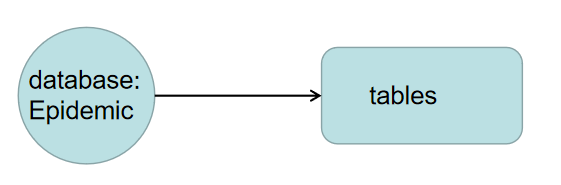


Figure 1.4

**Tables：**

**all\_name:**

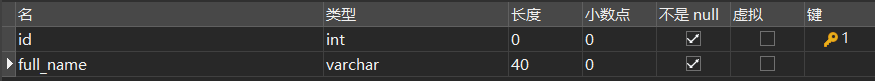


Figure 1.5.1

**student:**

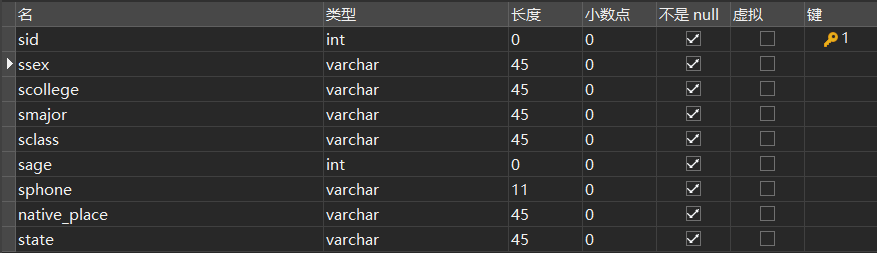


Figure 1.5.

**administrator:**



Figure 1.5.3

**Info:**

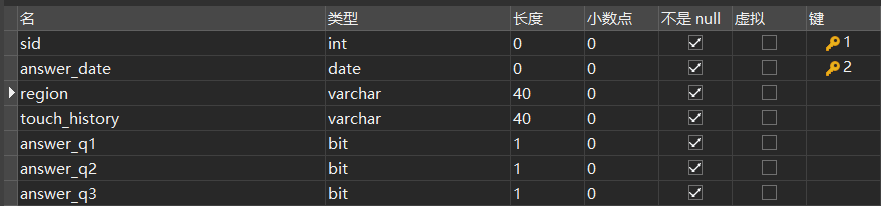


Figure 1.5.4

**administrator\_res:**

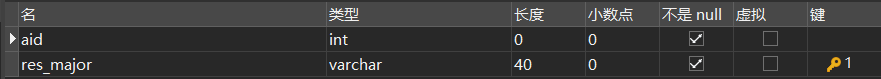


Figure1.5.5

**question:**

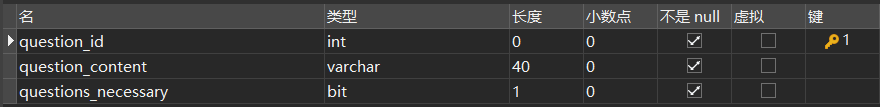


Figure 1.5.6

##### 3.3.3Sql queries:

Creat tables:

create table all\_name(

id int primary key,

full\_name varchar(40) not null

);

create table info

(

sid int,

answer\_date date,

region varchar(40) not null,

touch\_history varchar(40) not null,

answer\_q1 bit not null,#sql不支持bool型

answer\_q2 bit not null,

answer\_q3 bit not null,

constraint pk\_info primary key( sid,answer\_date)

);

create table questions(

question\_id int ,

question\_content varchar(40) not null,

questions\_necessary bit not null,

constraint pk\_questions primary key( question\_id )

);

create table Administrator

(

aid INT PRIMARY KEY,

asex VARCHAR(4) NOT NULL,

aphone VARCHAR(40) NOT NULL,

adep VARCHAR(40) NOT NULL,

constraint fk\_ad foreign key( aid ) references all\_name(id)

);

CREATE TABLE administrator\_res(

aid int NOT NULL,

res\_major varchar(40),

constraint pk\_ar PRIMARY KEY (res\_major)

) ;

CREATE TABLE student (

sid int,

ssex varchar(45) NOT NULL,

scollege varchar(45) NOT NULL,

smajor varchar(45) NOT NULL,#专业+年级

sclass varchar(45) NOT NULL,

sage int NOT NULL,#此处是年龄，给予保留

sphone VARCHAR(40) NOT NULL,

native\_place varchar(45) NOT NULL,

state varchar(45) NOT NULL,

constraint pk\_st PRIMARY KEY (sid),

constraint fk\_st foreign key( sid ) references all\_name(id)

) ;

We have inserted some data for testing, but we won’t show the query code here for there are too many of them. Here is the screen shot of part of the queries.



Figure 1.6.1

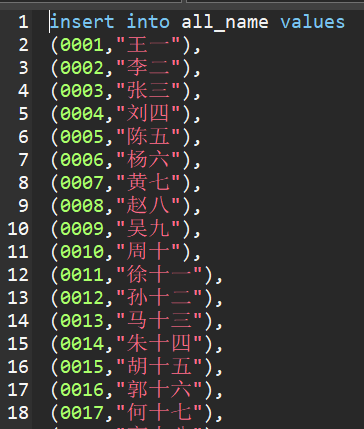


Figure 1.6.2

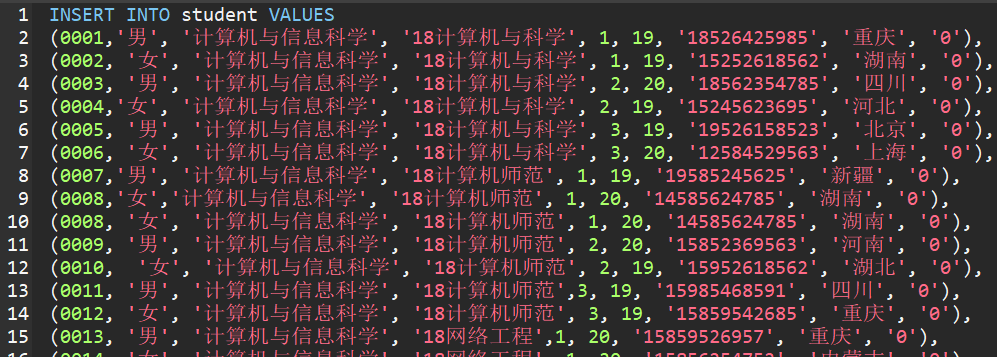


Figure 1.6.3

##### 3.3.4 Java code analysis

**UML reveal:**

The main interface, insert interface, search interface;

Father class: MyWindow class, MyPanel Class;

Coded by 曾令瀚.

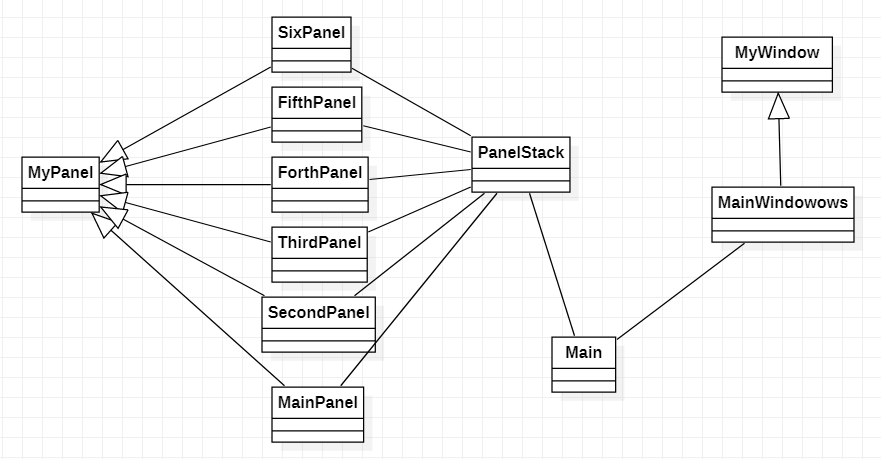


Figure 1.7.1

MyPanel: The father class of all the panels (a re-design of the class JPanel to make it easier to use).

MyWindow: The father class for the whole frame of the program (a re-design of the class JFrame to make it easier to use).

Main: The entrance to start the program.

MainPanel: The graphic interface design of the main interface of the program and its connection with the database.

SecondPanel: The graphic interface designs the function of insert students’ health status information of the program and its connection with the database.

ThirdPanel: The graphic interface designs the function of search students’ health status information by Id and revise them of the program and its connection with the database.

ForthPanel: The graphic interface designs the function of search students’ health status information by Id and revise them of the program and its connection with the database.

FifthPanel: The function of delete students’ health status information. (The delete function share the same interface with the function “search and revise”.

SixthPanel: The design of the back component for user to get back to the main frame.

The view interface;

Coded by 陈志宏：

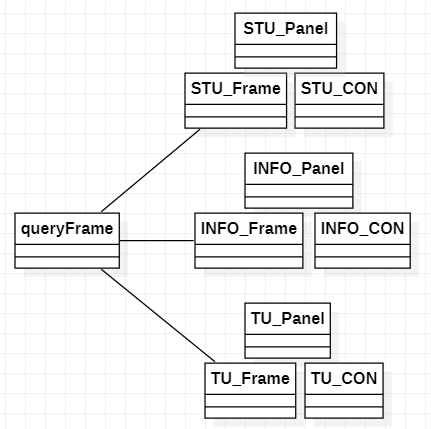


Figure 3.7.2

STU：To view the basic information of students.

INFO: To view the health status information of students.

TU: To view the basic information of consolers.

Frame: The graphical interface design of the frame.

Panel: The graphical interface design of the panels.

CON: The connection of the interface with the database

**The use of the JDBC(Java Database Connector)**

try {

System.out.println("调用next");

int rownum;

jdbc\_properties jdbc\_temp = new jdbc\_properties();// The class that connect to the attributes

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

Connection conn\_temp = DriverManager.getConnection(jdbc\_temp.JDBC\_URL, jdbc\_temp.JDBC\_USER,

jdbc\_temp.JDBC\_PASSWORD);

//Creat the SQL query String veriable

Statement stmt\_temp = conn\_temp.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE,

ResultSet.CONCUR\_UPDATABLE);

String query\_temp = "select all\_name.full\_name,info.\* from info,all\_name where info.sid=all\_name.id and answer\_date = \"2020-06-"

+ date + "\";";

ResultSet rs\_temp = stmt\_temp.executeQuery(query\_temp);

rs\_temp.last();

rownum = rs\_temp.getRow();// 注意

if (rownum - (first\_data + num\_data) >= 10) {

first\_data += 10;

} else if (first\_data < rownum - 10) {

first\_data += 10;

num\_data = rownum - first\_data;

num = num\_data;

} else {

first\_data = 0;

num\_data = 10;// 初始化

num = num\_data;

}

String query\_temp2 = "select all\_name.full\_name,info.\* from info,all\_name where info.sid=all\_name.id and answer\_date = \"2020-06-"

+ date + "\" LIMIT " + first\_data + "," + num\_data + ";";

ResultSet rs\_temp2 = stmt\_temp.executeQuery(query\_temp2);

int input = 0;// imput the position

while (rs\_temp2.next()) {// keep the number of the data

input\_date[input] = rs\_temp2.getString("info.answer\_date");

input\_sid[input] = rs\_temp2.getInt("info.sid");

input\_name[input] = rs\_temp2.getString("all\_name.full\_name");

input\_region[input] = rs\_temp2.getString("info.region");

input\_touch[input] = rs\_temp2.getString("info.touch\_history");

input\_q1[input] = rs\_temp2.getInt("info.answer\_q1");

input\_q2[input] = rs\_temp2.getInt("info.answer\_q2");

input\_q3[input] = rs\_temp2.getInt("info.answer\_q3");

input++;

}

} catch (Exception e) {

System.out.println(e.getMessage());

System.out.println("进步函数：数据库连接失败！");

}

}

### 4 Brief instructions for use

|  |  |
| --- | --- |
| Function：Insert information | |
| Functional block diagram    Figure 2.1.1 | |
| description | Insert new information of students’ health state, the system will only make an update on the database when the format of the information being inserted is correct. |

|  |  |
| --- | --- |
| Function：Search information | |
| Functional block diagram    Figure 2.1.2 | |
| description | Insert student’s Id to search his/her health status information. The system will prompt user that the id has no corresponding student and ask the user to insert again. The information will be shown if the Id is available, and user can revise that information by clicking the mouse. |

|  |  |
| --- | --- |
| Function：View information | |
| Functional block diagram    Figure 2.1.3 | |
| description | View all the students’ basic information |

|  |  |
| --- | --- |
| Function：Delete information | |
| Functional block diagram    Figure 2.1.4 | |
| description | Delete the chosen information |

### 5 Main interface screenshots

##### 5.1The main frame:

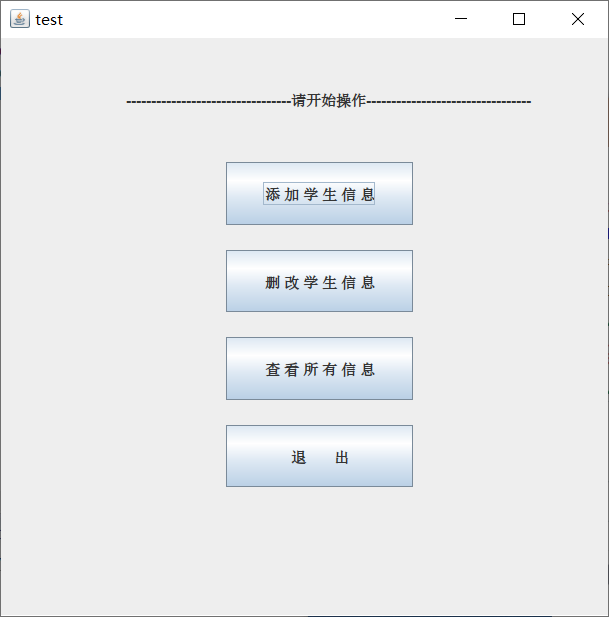


Figure 3.1

##### 5.2 Insert Frame

Prompt when sid is entered incorrectly Tips when the date is entered incorrectly

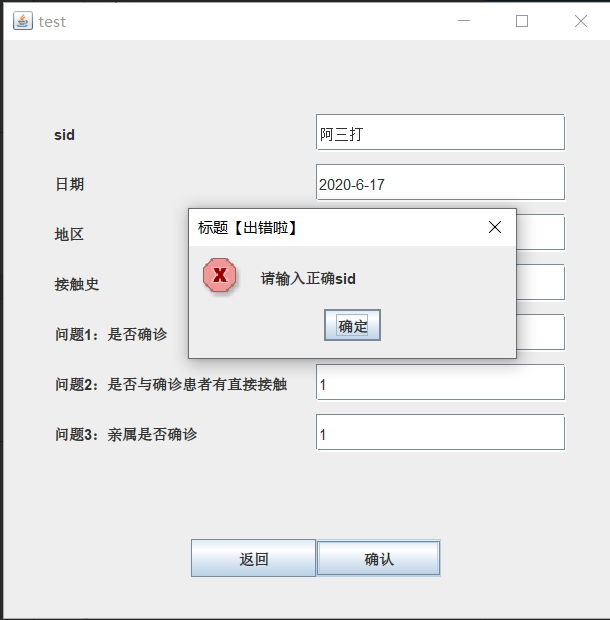
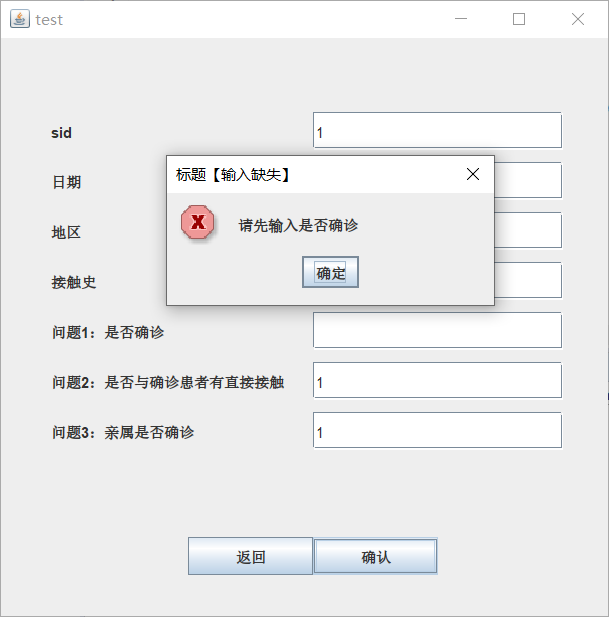


Figure 3.2.1

Prompt when entering sid not in record Prompt when input is missing



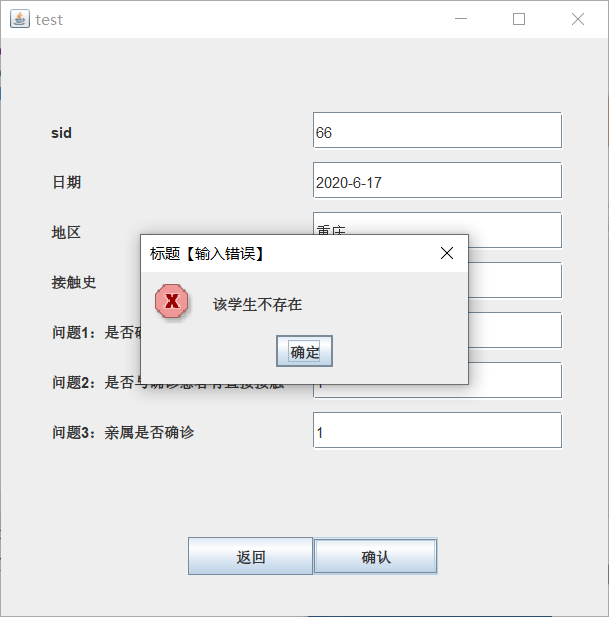


Figure 3.2.2

Prompt when question format input is not Boolean Inserted successfully and back

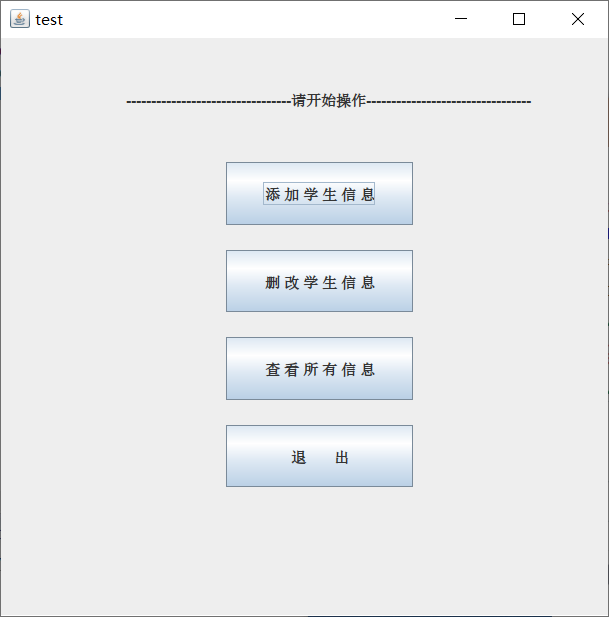
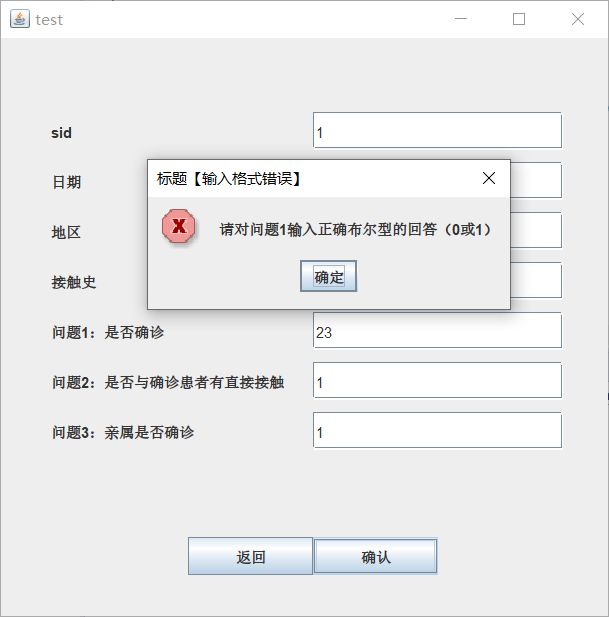


Figure 3.2.3

##### 5.3Delete/Revise Frame

Prompt when no Id found Prompt when Id was inserted in the Wrong Format

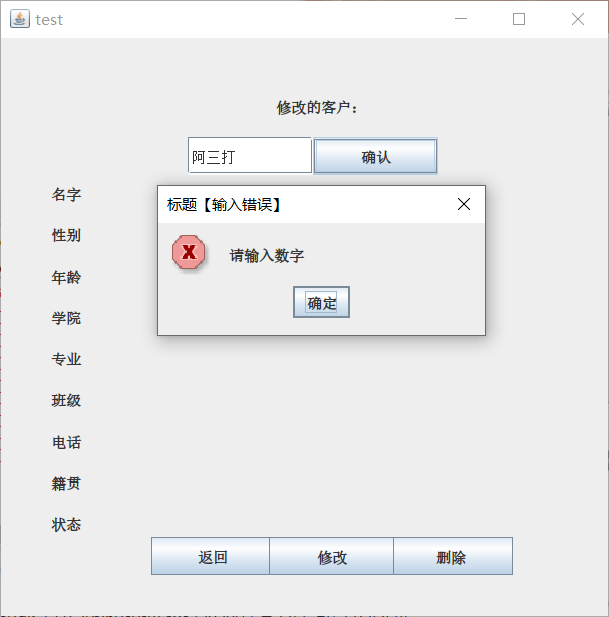




Figure 3.3.1

Corresponding person found

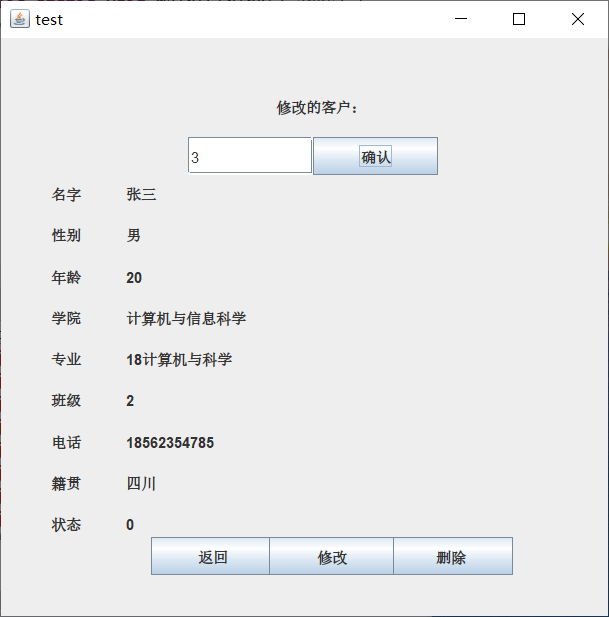


Figure 3.3.2

Revise his/her phone number



Figure 3.3.3

##### 5.4 View information

View frame

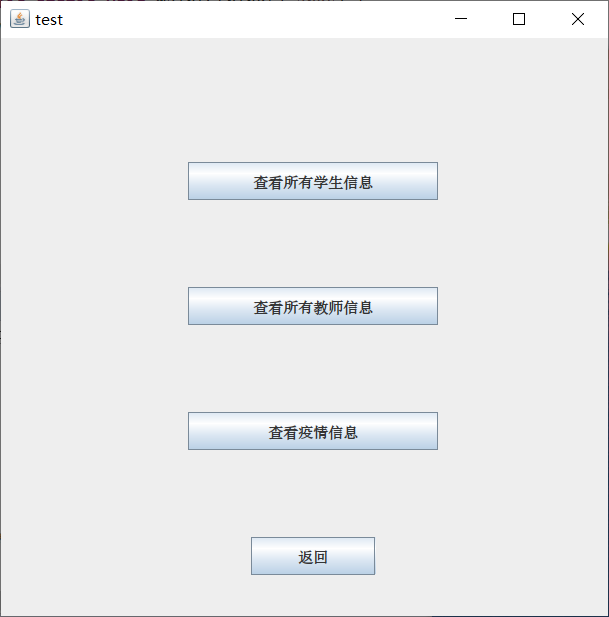


Figure 3.4.1

View all the students’ basic information



Figure 3.4.2

View students’ health status information

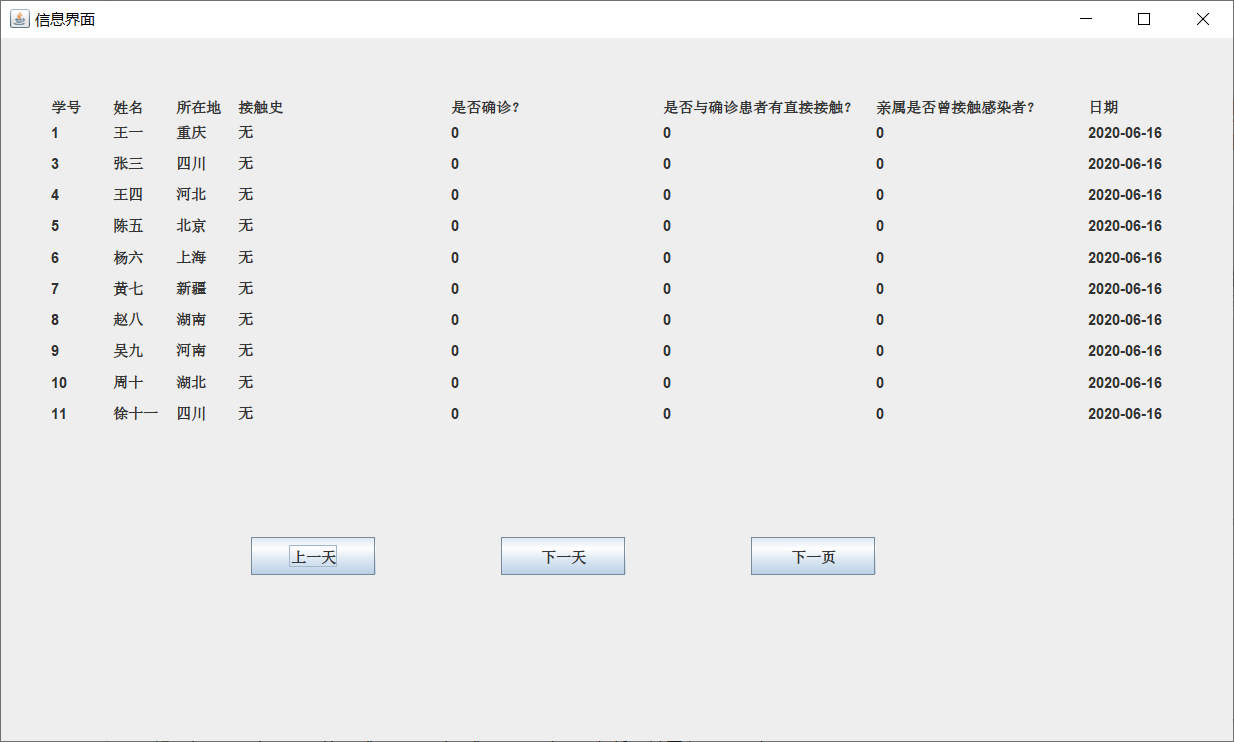


Figure 3.4.3

View consolers’ information

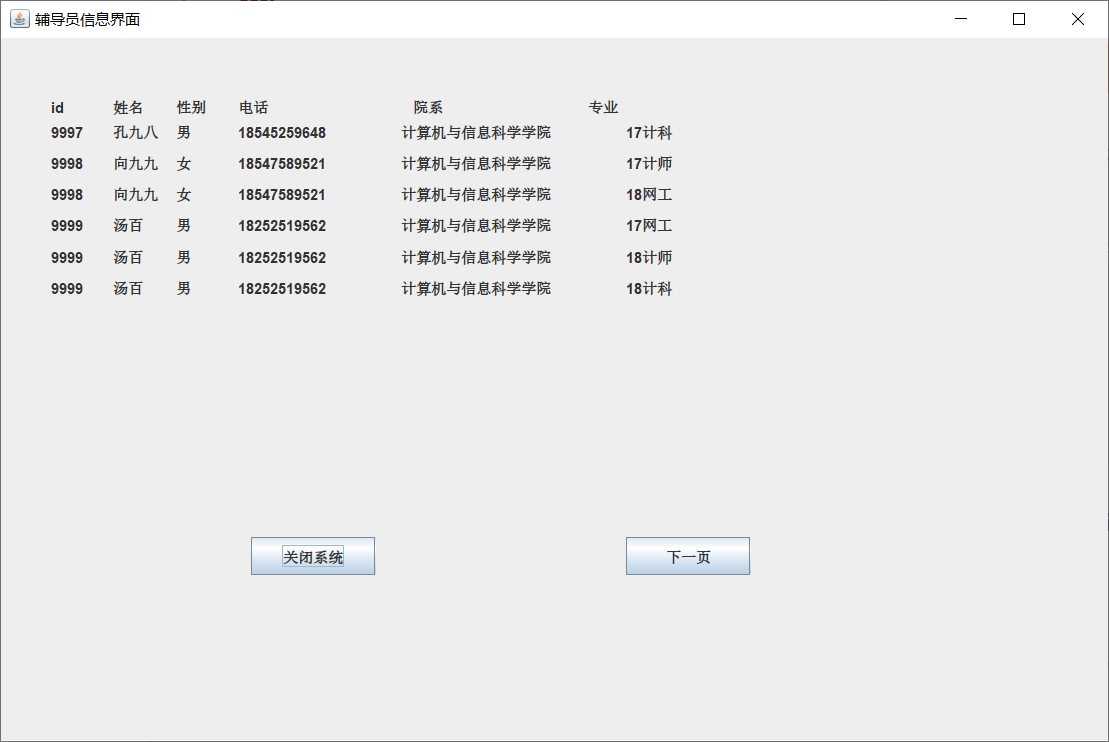


Figure 3.4.4

### 6 Summary

In general, this experiment has been very fruitful overall. We have implemented a simple outbreak student management system, which mainly realizes the functions of student information management and collection, from needs analysis, interface construction, beautification, to database connection, forms, The establishment of stored procedures and stored procedures. A series of process steps have given us a deeper understanding of the MySQL database and a deeper understanding of object-oriented programming and graphical interface design.

As for our database design, we applied the design theory of the database in the textbook to our works, and considered the integrity and security of the database. We also properly dealt with the redundancy and complexity of the data model. As for our program, We used Java's AWT to write a simple graphical interface, and used JDBC to connect our program to the back-end database. Although our graphical interface is not very beautiful, but it has achieved our core requirements for database operations, and has made the interface clean and tidy, ensuring the efficiency of the program.

Finally, this course design is a group work of a group of three of us. From the initial project total design to the division of specific content, our teamwork ability has been well trained. During our work, we encountered various problems, but we actively discussed, consulted a lot of information, and also asked our teachers about our problems. Fortunately, we solved all the problems and completed the course design assignment. We have learned a lot from this course design, and of course we are aware of our shortcomings. I believe that through this course design and future study, in the upcoming graduation design, there will be a greater breakthrough and make what you deserve.

### 7 References

[1] 王丽艳;数据库原理及应用(SQL Server 2012);人民邮电出版社;2018-3.

[2] 传智播客高教产品研发部;Java基础入门;清华大学出版社;2016-10.