

福州大学

数据库程序设计实践报告

设计题目： 学生成绩管理系统

学院： 软件学院

专业： 软件工程

组员： 220800424 卢桂芳

时间：2010 年 12 月

目 录

1 使用说明文档	3
1.1 数据库设置	3
1.2 数据源的建立	3
1.3 应用程序使用	3
1.4 开发环境简介	3
2 需求分析说明书	4
2.1 项目背景与目的	4
2.2 项目功能说明	4
2.2.1 系统管理员模块	5
2.2.2 教师模块	5
2.2.3 学生模块	6
2.3 系统流程分析	6
3 概念结构设计	
3.1 E-R 图	8
3.2 关系模型	8
3.3 详细设计	8
3.3.1 基本表	9
3.3.2 视图表	11
3.3.3 基本表的建立	12
3.3.4 视图的创建	14
4 编程设计	16
4.1 改进的 model2 模式	16
4.2 sqlHelper 的实现	17
4.3 数据模型的实现	20
4.4 用户登录功能块	22
4.5 界面之间的传值问题	23
4.6 界面跳转问题	23

4.7 连接数据库异常处理问题	23
4.8 动态显示表格内容问题	23
5 用户使用说明书	25
5.1 用户登陆功能模块	25
5.2 学生用户	26
5.2.1 学生登录后的主界面	26
5.2.2 密码管理的界面及个人信息管理	27
5.2.3 成绩查询界面	28
5.3 教师用户	29
5.3.1 教师登录后的主界面	29
5.3.2 密码管理的界面及个人信息管理	30
5.3.3 成绩查询界面	30
5.3.4 成绩管理界面	32
5.4 管理员用户	34
5.4.1 管理员用户登录后的主界面	34
5.4.2 信息管理界面	34
5.4.2.1 学生管理	35
5.4.2.2 教师管理	37
5.4.2.3 课程管理	39
5.4.2.4 授课管理	40
5.4.2.5 选课管理	42
5.4.2.6 成绩管理	44
5.5 帮助信息	46
5.5.1 关于本系统	46
5.5.2 联系方式	47
6 总结	48
参考文献	49

Chapter1. 使用说明文档

1.1. 数据库设置:

本设计提供了 studentscore.mdf、studentscore_log.ldf 二个文件。因此可以选择添加 MDF 和 LDF 创建 studentscore 数据库。

1.2. 数据源的建立:

- 1) 执行“Microsoft SQL Server”|“查询分析器”命令;
- 3) 在查询窗口创建 studentscore 数据库, 在 studentscore 上创建表格。
- 3) 或者执行“Microsoft SQL Server”|“企业管理器”命令, 找到“控制台根目录”|“SQL 服务器组”|“(local) (Window NT)”|“数据库”, 点击右键, 选择“所以任务”|“附加数据库”, 添加要附加数据库的 MDF 文件, 即 studentscore.mdf、studentscore_log.ldf 即可。
- 4) 完成数据源 studentscore 的创建。并测试数据源。

1.3. 应用程序使用:

1) 进入 Iombozeclipse 后, 单击“file”, 在下拉菜单中选择“Import”, 找到 General 下的 Existing Projects into Workspace, 在 Select root directory 点击 Browse, 找到“学生成绩管理系统”的目录点击即可。

2) 在“学生成绩管理系统”目录下, 直接执行学生成绩管理系统.jar 也可运行。

3) 登陆帐号密码: 管理员: 000001 密码: 8888

教师: 220001 密码: 8888

学生: 220800424 密码: 8888

1.4. 开发环境简介:

操作系统: Windows Professional XP SP3

DBMS: Microsoft SQL Server 2000.

应用程序开发环境: Iombozeclipse

数据源: DBMS

Chapter2.需求分析说明书

2.1. 项目背景与目的:

现在我国的大中专院校的学生成绩管理水平普遍不高，有的还停留在纸介质基础上，这种管理手段已不能适应时代的发展，因为它浪费了了许多的的人力和物力。在当今信息时代这种传统的管理方法必然被计算机为基础的信息管理系统所代替。学生成绩管理系统的主要目的就是利用软件实现成绩的录入，查询，修改等功能，使教师对学生成绩的管理更加容易，提高工作效率，降低管理成本。

本课题的题目是：学生成绩管理系统。学生成绩的管理一般分为成绩查询和成绩维护两部分，在实际的教学过程中，学生主要对成绩进行查询，了解自己的成绩情况，而教师除了能够对成绩查询之外，还要对成绩进行录入，修改等操作。对于整个系统而言，为了便于对各种数据进行维护，还要增设管理员的权限，方便对学生和教师信息的维护，从而达到灵活的管理学生成绩的效果。

因此从安全角度考虑，有必要针对不同的用户对数据的访问与编辑进行控制。其中学生只能对成绩查询，不允许修改和删除成绩，教师可以录入成绩，并可以修改已有的成绩，同时，教师也应该能够对成绩进行查询，以便了解整个教学情况。管理员可以对整个系统进行维护，需要管理员进行维护可能因为3种情况：一种是用户忘记密码，这时需要管理员对密码进行重置。另一种情况是某些数据由于人为的原因造成错误，这时需要管理员强制进行修改。还有一种情况是对学生信息，课程信息和教师信息一些基本信息的添加与修改等。

2.2 项目功能说明：

按照系统不同的用户角色，将整个系统划分为三大模块：系统管理员模块、教师模块、学生模块。以下是该系统的功能流程图：

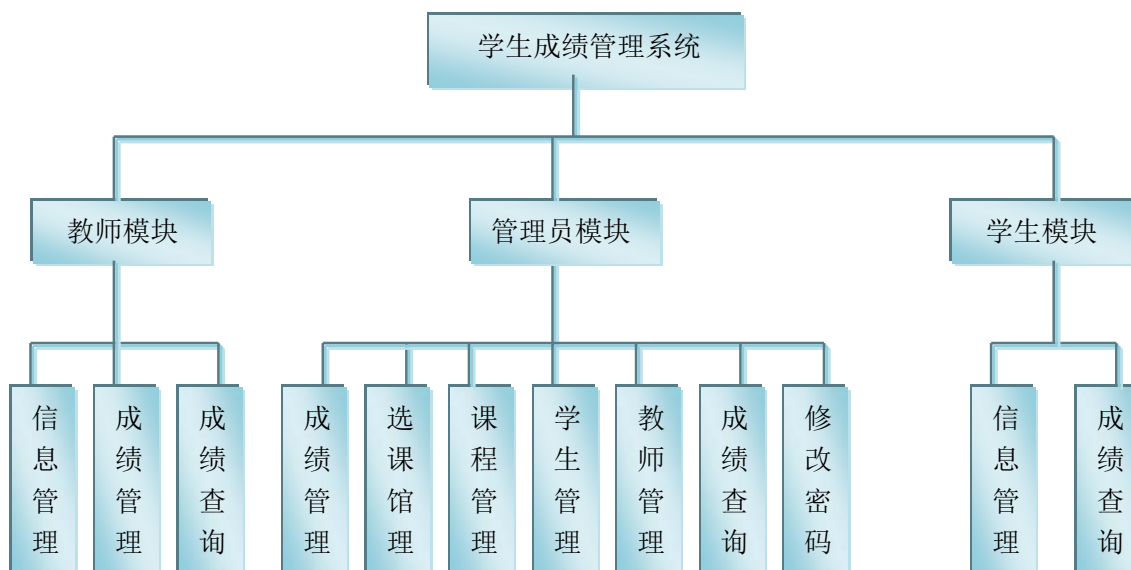


图 1.项目功能图

2. 2. 1 系统管理员模块

修改密码：管理员可以修改自己的登陆密码。

学生管理：能够对学生的一些基本信息进行添加，修改，删除等操作。学生的基本信息包括学生编号，姓名，性别，年龄，学院，专业，电话号码和地址等。

教师管理：能够对教师的一些基本信息进行添加，修改，删除等操作，教师的基本信息包括教师编号，教师姓名，性别，年龄，职称等。

课程管理：能够对课程的一些基本信息进行添加，修改，删除等操作，课程的基本信息包括课程编号，课程名称，该门课程的学时及学分。

授课管理：能够对教师授课的情况进行管理。包括添加，修改和删除等操作。当教师准备教授课程时要记录到数据库中，方便今后对课程、教师等有关信息的查询。

选课管理：能够对学生选课的情况进行管理。包括添加，修改和删除等操作，当学生选完课程时要记录到数据库中，方便今后对学生选课等有关信息的查询。

成绩管理：能够对成绩的情况进行管理，该部分是系统最主要的功能，当一门课的成绩结果公布时，要相应的记录到数据库中，这样，学生和教师才能够进行成绩的查询。

信息查询：能够按不同的查询条件进行学生成绩查询。

2. 2. 2 教师模块

信息管理：教师可以修改个人的信息已经登陆密码。

录入成绩：教师能够对所授课程的成绩进行录入操作。

成绩查询：教师能够查询所授课的班级的学生成绩。

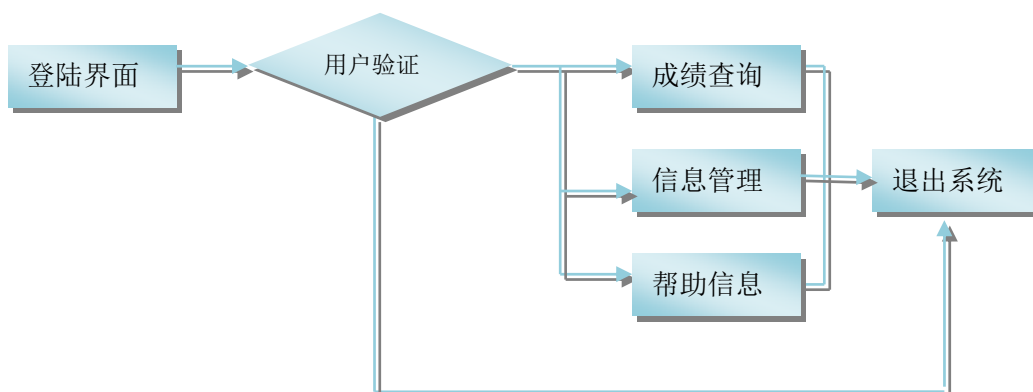
2. 2. 3 学生模块

信息管理：学生可以修改个人的信息已经登陆密码。

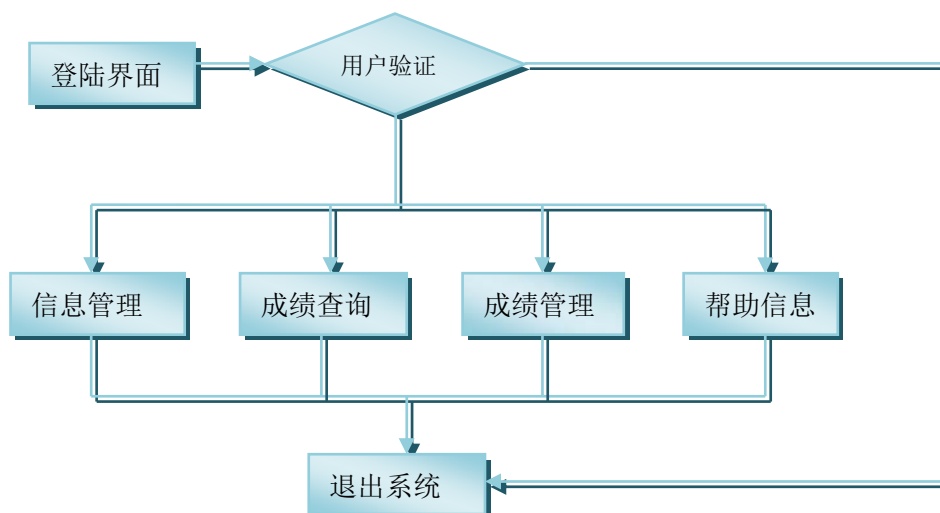
成绩查询：学生能够查询所授课的班级的学生成绩，但不能进行其他操作。

2. 3 系统流程分析

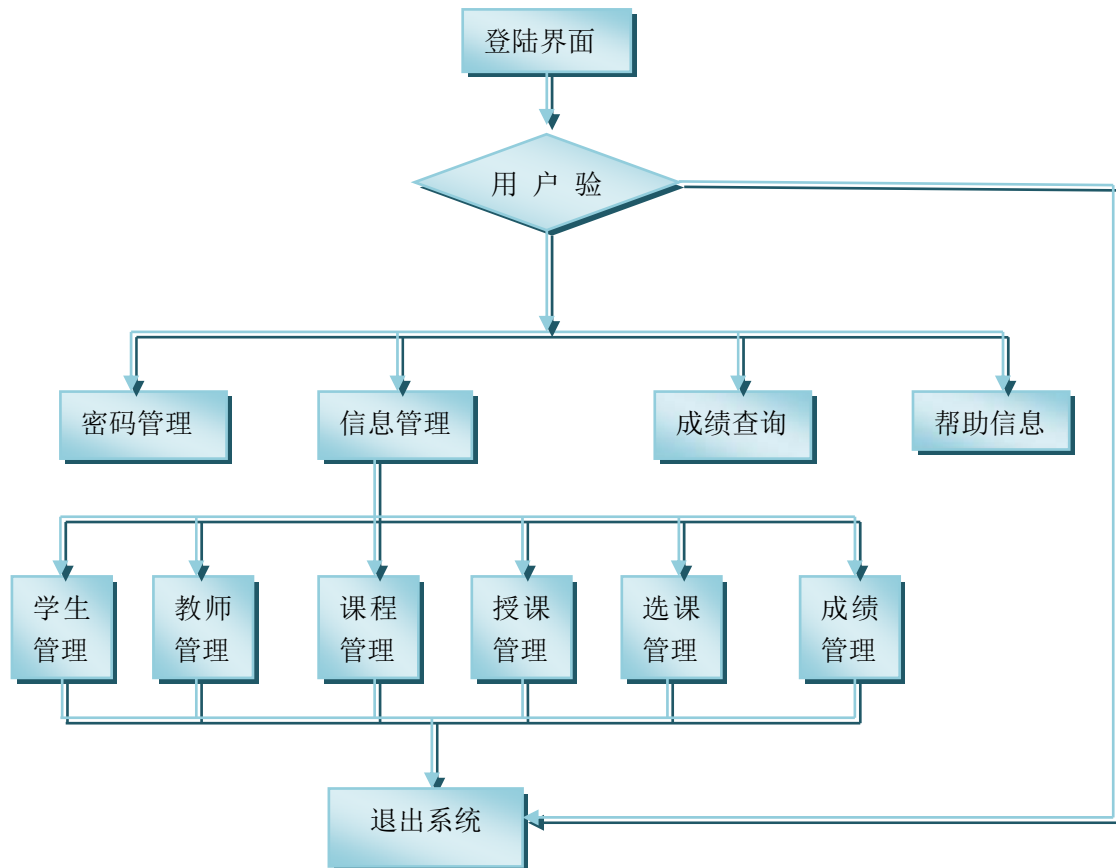
学生用户：



教师用户：



管理员用户：



chapter3. 概念结构设计

3.1.E-R 图

在以上的系统分析的基础上，设计出系统的 E-R 图，如图 2 所示。
 主要包括 3 个实体，分别是学生实体、管理管理员实体、教师实体、课程实体以及三个关系，即授课、选课和成绩关系。

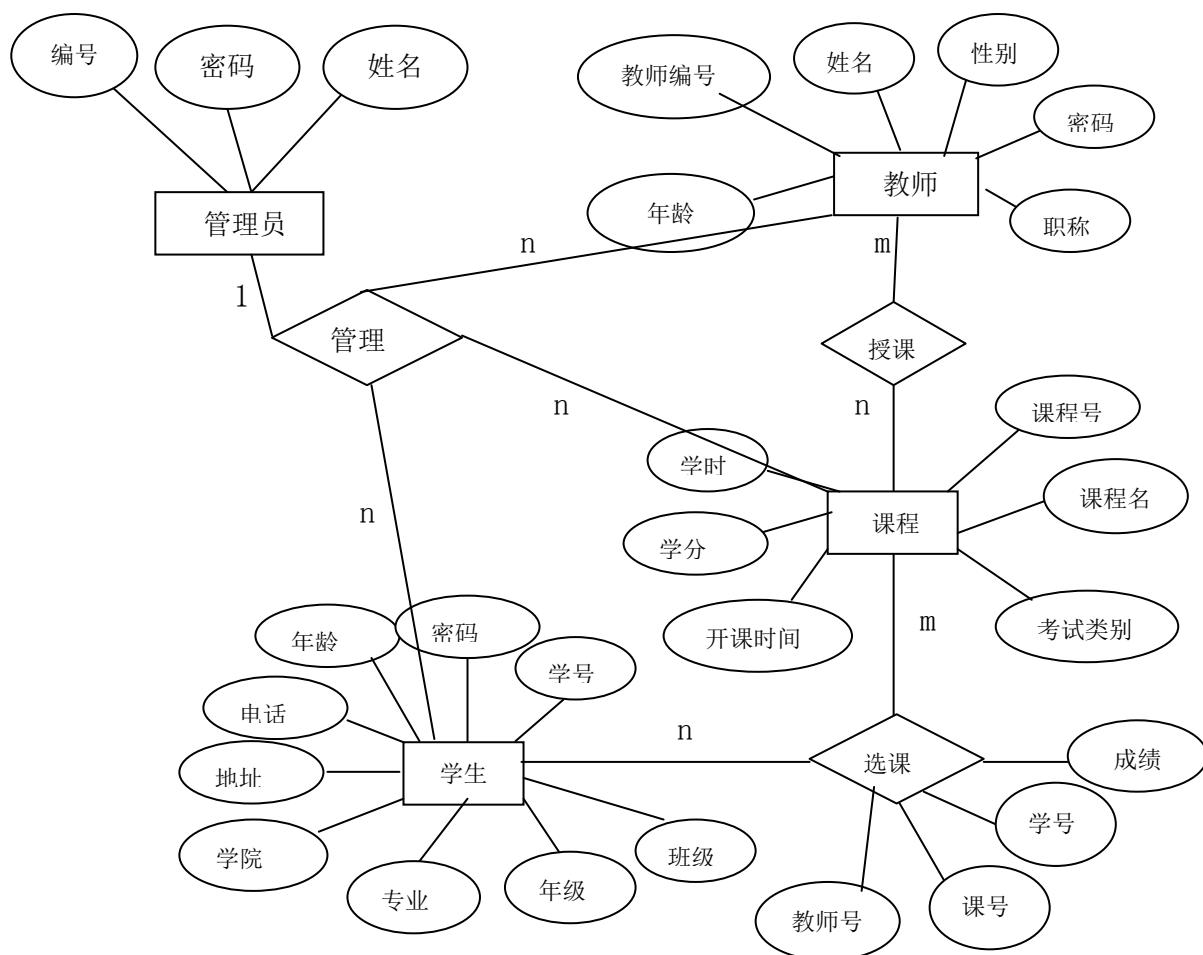


图 2 E-R 图

3.2 关系模型

管理员（管理员编号，姓名，密码）

教师（教师号，姓名，密码，性别，年龄，职称）

学生（学号，年级，班级，姓名，密码，学院，专业，性别，年龄，
 电话，地址）

课程（课程号，课程名，学时，学分，考试类别，开课时间）

授课（教师号，课程号）

选课/成绩（课程号，教师号，学号，成绩）

3.3. 详细设计

3.3.1 基本表

首先创建一个数据库，命名为：StudentScore。该数据库总共包含 6 个表。即管理员表、教师表、学生表、课程表、授课表、选课/成绩表。根据以上分析，其中成绩表通过学号与学生表联系起来，通过课程号与课程表联系起来；授课表通过课程号与课程表联系起来，通过教师号与教师表联系起来。选课表通过学号与学生表联系起来，通过课程号与课程表联系起来，通过教师号与教师表联系起来，各表的设置如表 1 至表 6 所示。

1、管理员信息表(administrator)

序号	字段名称	字段描述	类型	位数	属性
1	admin_no	管理员编号	char	20	非空、主键
2	admin_name	姓名	char	50	非空
3	admin_password	密码	char	20	非空

表 1

2、教师信息表(teacher)

序号	字段名称	字段描述	类型	位数	属性
1	Teacher_no	教师编号	char	20	非空、主键
2	Teacher_name	姓名	Char	20	非空
3	Teacher_password	密码	Char	20	非空
4	Teacher_sex	性别	Char	2	非空
5	Teacher_age	年龄	Int	3	非空
6	Teacher_level	职称	cher	20	非空

表 2

3、学生信息表(student)

序号	字段名称	字段描述	类型	位数	属性
1	Student_no	学号	char	20	非空、主键

2	Student_grade	年级	char	6	非空
3	Student_class	班级	char	6	非空
4	Student_name	姓名	char	20	非空
5	Student_password	密码	char	20	非空
6	Student_academy	学院	char	30	非空
7	Student_major	专业	char	30	非空
8	Student_sex	性别	char	2	非空
9	Student_age	年龄	int	3	非空
10	Student_phone	电话	char	15	
11	Student_address	地址	char	50	

表 3

4、课程信息表(course)

序号	字段名称	字段描述	类型	位数	属性
1	Course_no	课程号	char	20	非空、主键
2	Course_name	课程名	char	50	非空
3	Course_hour	学时	int	4	非空
4	Course_credit	学分	int	4	非空
5	Sytle	考试类别	Char	20	非空
6	Time	开课时间	Char	10	非空

表 4

5、授课(teach)

序号	字段名称	字段描述	类型	位数	属性
1	Teacher_no	教师号	char	20	非空、主键
2	Course_no	课程号	char	20	非空、主键

表 5

6、选课/成绩表(score)

序号	字段名称	字段描述	类型	位数	属性
1	Course_no	课程号	char	20	非空、主键
2	Teacher_no	开课老师号	Char	20	非空

3	student_no	学号	Char	20	非空、主键
4	Score	成绩	Int	4	

表 6

3.3.2 视图表

1、授课视图 (teachview)

序号	字段名称	字段描述	类型	位数	属性
1	Teacher_no	教师号	char	20	非空、主键
2	Teacher_name	教师名	Char	20	非空
3	Course_no	课程号	char	20	非空、主键
4	Course_name	课程名	Char	50	非空

表 7

2、选课视图 (selectcourseview)

序号	字段名称	字段描述	类型	位数	属性
1	Course_no	课程号	char	20	非空、主键
2	course_name	课程名	Char	50	非空
3	Teacher_no	开课老师号	Char	20	非空
4	Teacher_name	开课老师	char	20	非空
4	student_no	学号	Char	20	非空、主键
5	Student_name	学生名	Char	20	非空

表 8

3、成绩视图 (scoreview)

序号	字段名称	字段描述	类型	位数	属性
1	Course_no	课程号	char	20	非空、主键
2	Student_no	学号	char	20	非空、主键
3	Student_name	姓名	Char	20	非空
4	Course_name	课程名	Char	50	非空
5	credit	学分	Int	4	非空

6	Score	成绩	int	4	非空
7	Style	考试类别	char	20	非空
8	Time	开课时间	Char	10	非空
9	Teacher_no	开课老师号	Char	20	非空
10	Teacher_name	开课老师名	Char	20	非空

表 9

3.3.3 基本表的建立

-- 管理员信息表

```
create table administrator
(
admin_no char(20) primary key,
admin_name char(20) not null,
admin_password char(20) not null
);
```

-- 教师信息表

```
create table teacher
(
teacher_no char(20) primary key,
teacher_name char(20) not null,
teacher_password char(20) not null,
teacher_sex char(2) not null,
teacher_age smallint not null,
teacher_level char(20) not null
);
```

-- 学生信息表

```
create table student
(
student_no char(20) primary key,
student_grade char(6) not null,
student_class char(6) not null,
```

```
student_name char(20) not null,
student_password char(20) not null,
student_academy varchar(30) not null,
student_major varchar(30) not null,
student_sex char(2) not null,
student_age smallint not null,
student_phone char(15),
student_address varchar(50)
);
-- 课程信息表
create table course
(
course_no char(20) primary key,
course_name varchar(50) not null,
course_hour smallint not null,
course_credit smallint not null,
sytle char(20) not null,
time char(10) not null
);
-- 授课表
create table teach
(
teacher_no char(20) not null,
course_no char(20) not null,
foreign key (teacher_no) references teacher(teacher_no),
foreign key (course_no) references course(course_no)
);
-- 选课表
create table score
(
```

```
course_no char(20) not null,  
teacher_no char(20) not null,  
student_no char(20) not null,  
score int,  
primary key (course_no,student_no),  
foreign key (course_no) references course(course_no),  
foreign key (student_no) references student(student_no)  
);
```

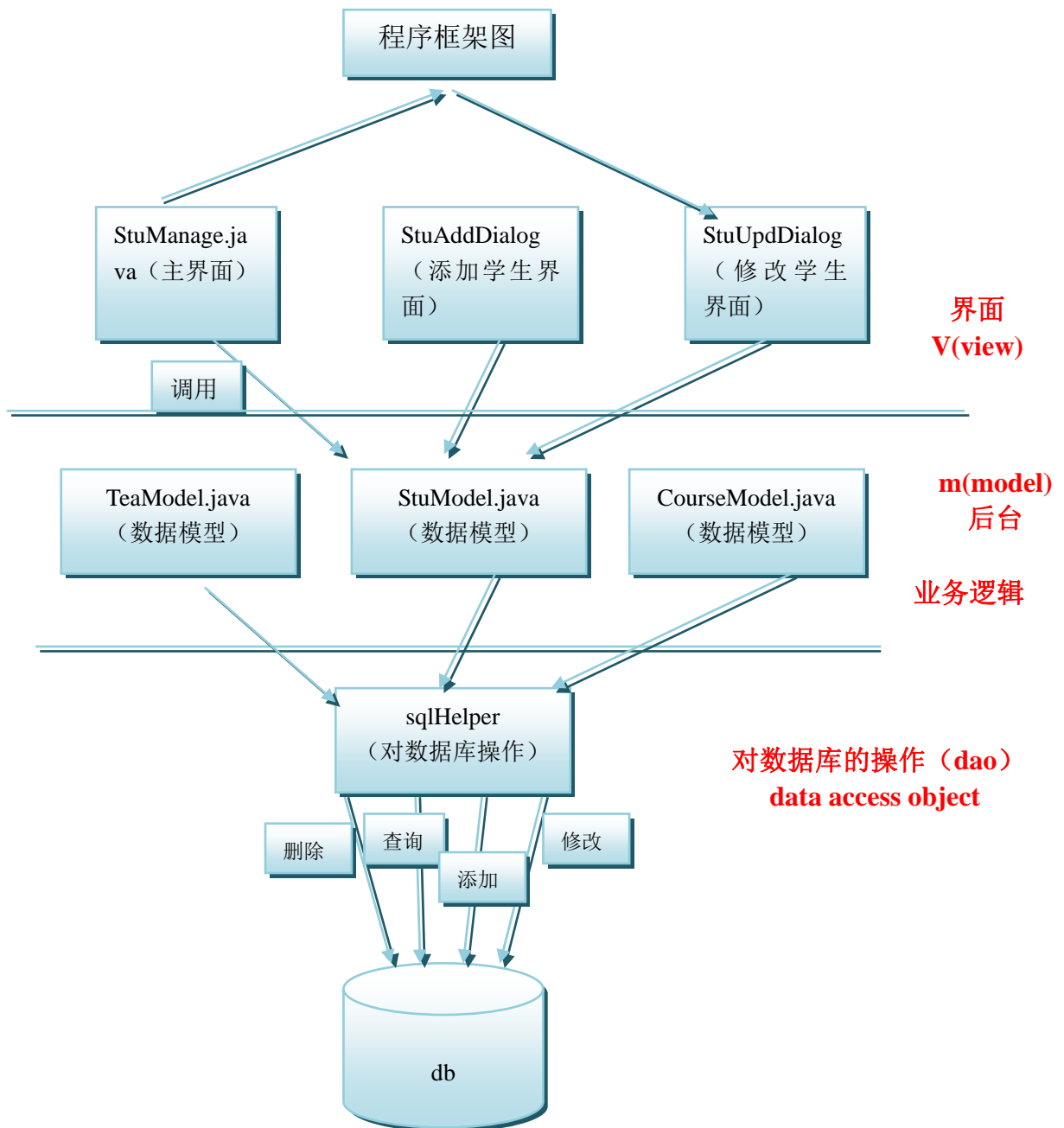
3.3.4. 视图的创建

```
CREATE VIEW teachview(teacher_no,teacher_name,course_no,course_name)  
AS  
SELECT  teach.teacher_no, teacher.teacher_name, teach.course_no,  
course.course_name,  
FROM    teach, teacher, course  
WHERE teach.teacher_no=teacher.teacher_no AND teach.course_no=  
course.course_no  
GO  
  
CREATE VIEW  
selectcourseview(course_no,course_name ,teacher_no,teacher_name,student_no,  
student_name)  
AS  
SELECT  score.course_no, course.course_name, score.teacher_no,  
teacher.teacher_name, score.student_no, student.student_name,  
FROM    score, teacher, course,student  
WHERE score.teacher_no=teacher.teacher_no AND score.course_no=  
course.course_no AND score.student_no=student.student_no  
GO  
  
CREATE VIEW scoreview(course_no, student_no,  
student_name ,course_name ,credit,score,style,time,teacher_no,teacher_name)  
AS
```

```
SELECT  score.course_no, score.student_no,  
student.student_name ,course.course_name ,course.credit,score.score,course.style,  
course.time,score.teacher_no,teacher.teacher_name  
FROM    score, teacher, course,student  
WHERE   score.teacher_no=teacher.teacher_no AND score.course_no=  
course.course_no AND score.student_no=student.student_no  
GO
```


Chapter4.编程设计

4.1 改进的 model2 模式



从上图中我们可以看到，对于 `StuManage.java` 部分只负责主界面，而不需要涉及数据库的操作，只需要调用数据模型 `stuModel.java`。

由于在学生管理系统中使用的数据表有多个，使用的数据模型也具有多个，为了使各个模型的增删改查处理更方便，于是将后台抽象为两层，一层是处理业务逻辑的 `model` 层，一层是对数据库的操作 `dao` 层，`dao` 层创建一个 `SqlHelp.java`，专门由来对数据库的操作类，这样，界面只调用数据模型，而数据模型调用 `SqlHelp`，使得每一层所完成的功能不一样。若以后需更改数据库的连接，也只需修改 `SqlHelp` 一处即可。这样改进后的 `model2` 模式不仅实现了功能模块和显示模块的分离，同时还提高了应用系统的可维护性、可扩展性、可移植性和组件的可复用性。

4.2 sqlHelper 的实现

```
public class SqlHelper {
// 定义操作数据库需要的东东
    PreparedStatement ps = null;
    Connection ct = null;
    ResultSet rs = null;
    //连接MySQL
    //String url="jdbc:mysql://localhost/studentscore";
    //String username="root";
    //String password="root";
    // String dirver = "com.mysql.jdbc.Driver";
    //连接SQL Server2000
    String url =
"jdbc:microsoft:sqlserver://127.0.0.1:1433;databaseName=studentsc
ore";
    String username = "sa";
    String password = "";
    String dirver = "com.microsoft.jdbc.sqlserver.SQLServerDriver";

//关闭数据库资源
    public void close()
    {
        try{
            if(rs != null) rs.close();
            if(ps != null) ps.close();
```

```
        if(ct != null) ct.close();
    }catch (Exception e){
        e.printStackTrace();
    }
}
//连接数据库
public ResultSet queryExectue(String sql)
{
    try
    {
        //1.加载驱动
        Class.forName(driver);
        //2.得到连接
        ct=DriverManager.getConnection(url,username,password);
        //3.创建ps
        ps = ct.prepareStatement(sql);
        //给ps的问号赋值
        rs=ps.executeQuery();
    }
    catch(Exception e){
        e.printStackTrace();
    }finally{
        //关闭???
        //this.close();
    }
    return rs;
}

//查询数据库操作
public ResultSet queryExectue(String sql,String []paras)
{
    try
    {
        //1.加载驱动
        Class.forName(driver);
        //2.得到连接
        ct=DriverManager.getConnection(url,username,password);
        //3.创建ps
```

```
        ps = ct.prepareStatement(sql);
        //给ps的问号赋值
        for(int i=0;i<paras.length;i++)
        {
            ps.setString(i+1, paras[i]);
        }
        rs=ps.executeQuery();
    }
    catch(Exception e){
        e.printStackTrace();
    }finally{
        //关闭???
        //this.close();
    }
    return rs;
}

//把增删改合在一起
public boolean updExecute(String sql,String []paras)
{
    boolean b = true;
    try
    {
        //1.加载驱动
        Class.forName(driver);
        //2.得到连接
        ct=DriverManager.getConnection(url,username,password);
        //3.创建ps
        ps = ct.prepareStatement(sql);
        //给ps的问号赋值
        for(int i=0;i<paras.length;i++)
        {
            ps.setString(i+1, paras[i]);
        }
        //4.执行操作
        if(ps.executeUpdate()!=1)
        {
            b = false;
        }
    }
```

```

    }
    catch(Exception e){
        b = false;
        e.printStackTrace();
    }finally{
        this.close();
    }
    return b;
}
}

```

4.3 数据模型的实现

```

class courseModel extends AbstractTableModel{
    //rowData 用来存放行数据
    //columnName 存放列名
    Vector rowData,columnNames;

    public boolean updScore(String sql,String []paras)
    {
        //创建SqlHelper (如果程序并发性不考虑, 可以把SqlHelper做成static, )
        SqlHelper sqlHelper = new SqlHelper();
        return sqlHelper.updExecute(sql, paras);
    }

    //查询本质就是初始化
    public void queryScore(String sql,String []paras)
    {
        SqlHelper sqlHelper = null;
        columnName = new Vector();
        columnName.add("课程号");
        columnName.add("课程名");
        columnName.add("学时");
        columnName.add("学分");
        columnName.add("考试类别");
        columnName.add("开课时间");
        rowData = new Vector();
        try
        {

```

```
        sqlHelper = new SqlHelper();
        ResultSet rs = sqlHelper.queryExectue(sql, paras);
        while(rs.next())
        {
            Vector hang = new Vector();
            hang.add(rs.getString(1));
            hang.add(rs.getString(2));
            hang.add(rs.getString(3));
            hang.add(rs.getString(4));
            hang.add(rs.getString(5));
            hang.add(rs.getString(6));
            rowData.add(hang);
        }
    }
    catch(Exception e){
        e.printStackTrace();
    }finally{
        sqlHelper.close();
    }
}

//得到共用多少列
public int getColumnCount() {
    return this.columnNames.size();
}

//得到共用多少行
public int getRowCount() {
    return this.rowData.size();
}

//得到某行某列的数据
public Object getValueAt(int row, int column) {
    return ((Vector)this.rowData.get(row)).get(column);
}

public String getColumnName(int arg0) {
    return (String)this.columnNames.get(arg0);
}
}
```

4.4 用户登录功能模块

在该功能模块中，创建一个SqlHelper对象，调用sqlHelper.queryExectue(sql, paras)方法，得到该用户的密码，再用pwd.equals(sql_pwd)判断输入的密码是否相同，相同则进入学生管理界面，不同则提示密码错误，并以catch (Exception ee)来处理输入的用户名是否存在。

```
String sql_pwd;
JButton obj = (JButton)e.getSource();
name = new String(xingming.getText()).trim();
pwd = new String(mima.getPassword()).trim();
if (obj == denglu)
{
    SqlHelper sqlHelper = null;
    if (name.length() == 0) {
        JOptionPane.showMessageDialog((Component) null, "请输入您的用户名!", "提示信息", JOptionPane.ERROR_MESSAGE);
        xingming.requestFocus();
    }
    else if (pwd.length() == 0) {
        JOptionPane.showMessageDialog((Component) null, "请输入您的密码!", "提示信息", JOptionPane.ERROR_MESSAGE);
        mima.requestFocus();
    }
    else {
        try {
            sqlHelper = new SqlHelper();
            String sql = "select * from administrator where
admin_no=?";

            String paras[]={name};
            ResultSet rs = sqlHelper.queryExectue(sql, paras);

            rs.next();
            sql_pwd = rs.getString(3).trim();
            if (!rs.next()) {
                if (pwd.equals(sql_pwd)) {
                    this.dispose();
                    Admin_manage manage = new Admin_manage();
                    manage.setVisible(true);
                }
                else {

```

```

JOptionPane.showMessageDialog((Component)
null, "密码错误!", "提示信息", JOptionPane.ERROR_MESSAGE);
mima.requestFocus();
    }
}
} catch (Exception ee) {
    JOptionPane.showMessageDialog((Component) null, "
用户名不存在!", "提示信息", JOptionPane.ERROR_MESSAGE);
    xingming.requestFocus();
}finally{
    sqlHelper.close();
}
}
}

```

4.5 界面之间的传值问题

用户登陆成功后，可以修改个人密码。而在修改密码时，是要根据当前用户的用户名才能打开数据库中相应的用户表。这里就涉及到界面之间的值的传递。也就是从用户成功登陆后，必须要在他能操作的所有界面之间保存一个判别用户的值，即用户名。如果不能传递这样一个值的话，后面的相关界面则无法与数据库中对应的表相连。为此，可在用户的登陆功能定义一个静态变量，用来保存当前用户的用户名，这样，后面要打开数据库中的相关的表的时候，只要引用该静态变量即可。如在用户修改密码时就必须获得当前用户名才能打开相应用户的表，从而操作数据库。

4.6 界面跳转问题

用Java语言开发的界面不像网页那样，能方便的实现页面之间的跳转。为了实现界面之间的跳转，就必须用new语句初始化所要跳转到的界面的那个类。如从登陆界面跳转至管理员界面：

```

this.dispose();
Admin_manage manage1 = new Admin_manage();
manage1.setVisible(true);
this.dispose()即关闭当前界面，用setVisible(true)即显示另一新的界面。

```

4.7 连接数据库异常处理问题

在编写操作数据库的相关代码时，必须加上异常捕获语句，否则系统报错。

4.8 动态显示表格内容问题

在统计学生成绩模块中，必须要用到表格 JTable，但是因为数据库中的记录是动态的，经常变动的，所以无法确认表格的具体行数和列数。这就必须使用动态的方法生成相应的表格，即数据库中有多少条记录，表格就显示出多少条记录。这是一个相对较难处理的问题。以下是解决这个问题的相关代码。

```
CourseAddDialog sa = new CourseAddDialog(this, "添加课程", true);  
    //构建新的数据模型类，并更新  
    sm = new courseModel();  
    String[] paras2 = {"1"};  
    sm.queryScore("select * from course where l=?", paras2);  
    //更新JTable  
  
    table.setModel(sm);
```

只需构建新的数据模型，用 queryScore() 方法中传入相应的 SQL 语句，再用 table 的 setModel() 方法就可使查询的相关内容以表格的形式动态显示出。

Chapter5.用户使用说明书

5.1 用户登陆功能模块

用户登陆模块实现对用户合法性的检查、读取相应的用户权限，同时在登陆成功后根据用户的不同权限进入不同的操作界面。界面提供三种不同的用户选择，即管理员、教师、学生，用户根据需要进行相应的身份登录。如下图所示：



点击相应的身份后进入登陆界面，如下图的学生登录窗口（其他两种用户的登录窗口与学生的登录窗口一样！）：



若输入的用户名和用户密码正确，进入相应的用户界面，若不正确，则弹出密码错误或用户名不存在的提示信息。点击取消回到开始界面。

5.2 学生用户

5.2.1 学生登录后的主界面

本系统学生用户的权限只有对用户自己的密码管理（修改），成绩查询和帮助功能。



5.2.2 密码管理的界面及个人信息管理



在此界面修改密码。

个人信息

学号

220800424

班级

04

密码

8888

专业

软件工程

年龄

21

地址

福建

年级

08

姓名

卢桂芳

学院

软件学院

性别

女

电话

13645086173

修改

取消

在此界面修改个人信息

5.2.3 成绩查询界面

学生成绩查询功能，只能提供学生查询自己有修的课程，可以根据不同的查询方法，或是全部查询（单击全部按钮），或是查找单科成绩（根据课程号获是课程名），或是部分查询（按照学分，不同的考试类别等）。

测试数据学生学号 220800424，该学生修了八门课程

查询全部

成绩查询

课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师				
1	...	220800...	卢桂芳	...	数据库	4	90	正常考试...		220001	...	邓新国	...
2	...	220800...	卢桂芳	...	汉字与中...	1	88	正常考试...		220006	...	杨艳华	...
3	...	220800...	卢桂芳	...	高数	5	80	正常考试...		220004	...	黄岩	...
4	...	220800...	卢桂芳	...	操作系统	3	85	正常考试...		220002	...	江兰芳	...
5	...	220800...	卢桂芳	...	数据结构	4	88	正常考试...		220001	...	邓新国	...
6	...	220800...	卢桂芳	...	英语	4	78	正常考试...		220005	...	游帆	...
7	...	220800...	卢桂芳	...	java	3	83	正常考试...		220003	...	张舒	...
8	...	220800...	卢桂芳	...	计算机网...	4	74	课程实践...		220005	...	游帆	...

查询类型

课程号

课程号

课程名

学分

成绩

考试类别

开课时间

查询

全部

用课程号查找单科成绩：

成绩查询

课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师				
3	...	220800...	卢桂芳	...	高数	5	80	正常考试...		220004	...	黄岩	...

查询类型

课程号

数值

3

查询

5.3 教师用户

5.3.1 教师登录后的主界面

教师登录！

本人信息管理 学生成绩管理 信息查询 帮助

学生成绩管理系统



密码管理



成绩查询



成绩管理



帮助

本系统教师用户的权限学生用户本身的密码管理（与学生用户的一样！），

对选修自己所开课程的学生的成绩查询，成绩管理。

5.3.2 密码管理的界面及个人信息管理



密码修改

用户名: 220800424

密码:

在此界面修改密码。



个人信息

编号	220001	姓名	邓新国
密码	8888	性别	男
年龄	40	职称	讲师

在此查看个人信息。

5.3.3 成绩查询界面

教师成绩查询功能，只能提供教师查找有选修自己课程的学生成绩。提供多种查询方法，可以查询全部学生的全部成绩（单击全部按钮），查询某个学生的成绩（按学生学号或学生姓名查询），查询某个科目所有学生的成绩（按课程号或课程名查询），查找某个分数的所有学生（按分数查询），还要按学分查询等多种方式（如图）。

查询全部：

成绩查询

查询类型: 课程号 数值: 查询

课程号
课程名
学号
姓名
学分
成绩
考试类别
开课时间

序号	课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师
1	1	...	220800...	阮茂芳	数据库	4	83	正常考...	220001...	邓新国
2	1	...	220800...	胡雪钦	数据库	4	80	正常考...	220001...	邓新国
3	1	...	220800...	李碧玉	数据库	4	85	正常考...	220001...	邓新国
4	1	...	220800...	陈雅莉	数据库	4	80	正常考...	220001...	邓新国
5	1	...	220800...	卢桂芳	数据库	4	90	正常考...	220001...	邓新国
6	5	...	220800...	阮茂芳	数据结构	4	85	正常考...	220001...	邓新国
7	5	...	220800...	胡雪钦	数据结构	4	85	正常考...	220001...	邓新国
8	5	...	220800...	李碧玉	数据结构	4	80	正常考...	220001...	邓新国
9	5	...	220800...	陈雅莉	数据结构	4	80	正常考...	220001...	邓新国
10	5	...	220800...	卢桂芳	数据结构	4	88	正常考...	220001...	邓新国

排序

最大值

最小值

平均值

全部

返回

按课程后查询某个课程的所有学生成绩：

成绩查询

查询类型: 课程号 数值: 1 查询

序号	课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师
1	1	...	220800...	阮茂芳	数据库	4	83	正常考...	220001...	邓新国
2	1	...	220800...	胡雪钦	数据库	4	80	正常考...	220001...	邓新国
3	1	...	220800...	李碧玉	数据库	4	85	正常考...	220001...	邓新国
4	1	...	220800...	陈雅莉	数据库	4	80	正常考...	220001...	邓新国
5	1	...	220800...	卢桂芳	数据库	4	90	正常考...	220001...	邓新国

排序

最大值

最小值

平均值

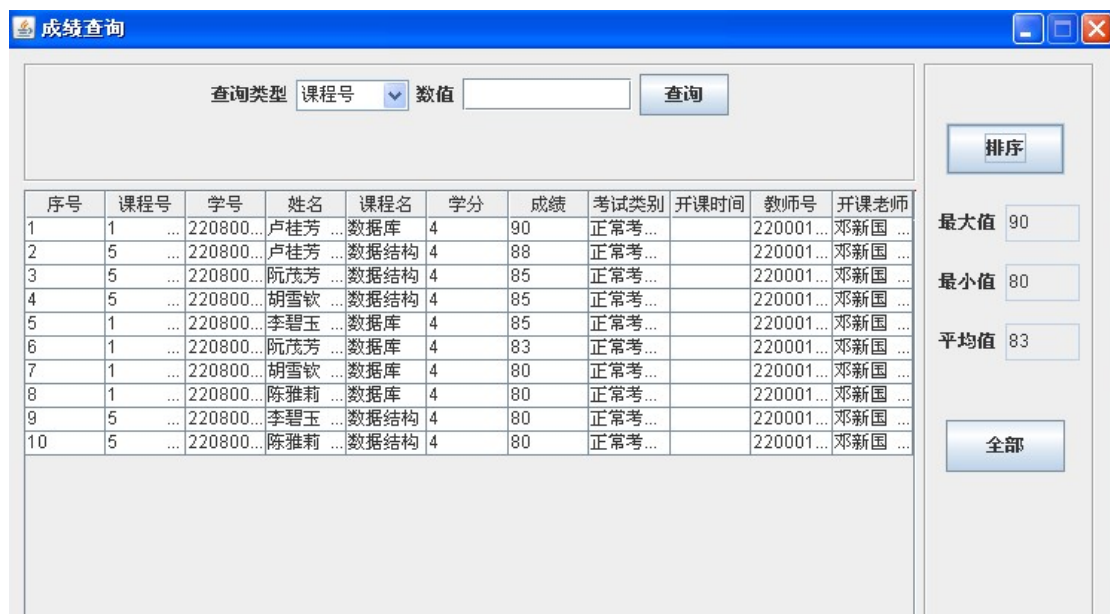
全部

按学生学号查询某个学生的成绩：



序号	课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师
1	1	220800...	李碧玉	数据库	4	85	正常考...		220001...	邓新国
2	5	220800...	李碧玉	数据结构	4	80	正常考...		220001...	邓新国

同时，本功能还提供先将成绩按从大到小排序后查询（默认按学号从小到大输出），输出最大值、最小值和平均值，如图：



序号	课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师
1	1	220800...	卢桂芳	数据库	4	90	正常考...		220001...	邓新国
2	5	220800...	卢桂芳	数据结构	4	88	正常考...		220001...	邓新国
3	5	220800...	阮茂芳	数据结构	4	85	正常考...		220001...	邓新国
4	5	220800...	胡雪钦	数据结构	4	85	正常考...		220001...	邓新国
5	1	220800...	李碧玉	数据库	4	85	正常考...		220001...	邓新国
6	1	220800...	阮茂芳	数据库	4	83	正常考...		220001...	邓新国
7	1	220800...	胡雪钦	数据库	4	80	正常考...		220001...	邓新国
8	1	220800...	陈雅莉	数据库	4	80	正常考...		220001...	邓新国
9	5	220800...	李碧玉	数据结构	4	80	正常考...		220001...	邓新国
10	5	220800...	陈雅莉	数据结构	4	80	正常考...		220001...	邓新国

5.3.4 成绩管理界面

教师成绩管理功能，只能提供教师管理有选修自己课程的学生成绩。管理学生成绩的同时，也能查看学生成绩。提供多种查看方法，可以查询全部学生的全部成绩（单击全部按钮），查询某个学生的成绩（按学生学号或学生姓名查询），查询某个科目所有学生的成绩（按课程号或课程名查询），

查找某个分数的所有学生（按分数查询），还要按学分查询等多种方式（如图）。（注：此处的学生成绩查询与成绩查询功能几乎一样！只是没有提供成绩排序，成绩的最大值，最小值和平均值）

成绩查询

查询类型

课程名

数值

查询

课程号

学号

姓名

课程名

学分

成绩

考试类别

开课时间

教师号

开课老师

1	...	220800...	阮茂芳	...	83	正常考试...		220001	邓新国	...
1	...	220800...	胡雪钦	...	80	正常考试...		220001	邓新国	...
1	...	220800...	李碧玉	...	85	正常考试...		220001	邓新国	...
1	...	220800...	陈雅莉	...	80	正常考试...		220001	邓新国	...
1	...	220800...	卢桂芳	...	90	正常考试...		220001	邓新国	...
5	...	220800...	阮茂芳	...	85	正常考试...		220001	邓新国	...
5	...	220800...	胡雪钦	...	85	正常考试...		220001	邓新国	...
5	...	220800...	李碧玉	...	80	正常考试...		220001	邓新国	...
5	...	220800...	陈雅莉	...	80	正常考试...		220001	邓新国	...
5	...	220800...	卢桂芳	...	88	正常考试...		220001	邓新国	...

添加

修改

删除

全部

返回

添加学生成绩和修改学生成绩：（添加修改学生成绩，修改学生成绩需要选中该学生后单击修改，若没有选中，则会弹出提示窗口）

添加成绩

课程号

1

学号

220800407

课程名

数据库

姓名

阮茂芳

学分

4

成绩

80

考试类别

正常考试

开课时间

老师号

220001

开课老师

邓新国

添加

取消

修改成绩

课程号

1

学号

220800409

课程名

数据库

姓名

胡雪钦

学分

4

成绩

80

考试类别

正常考试

开课时间

老师号

220001

开课老师

邓新国

修改

取消

提示窗口：



5.4 管理员用户

5.4.1 管理员用户登录后的主界面

本系统管理员用户的权限最多，包括其本身的密码管理（与学生用户一样！），对所有学生成绩的查询，对学生用户，教师用户等信息管理。



5.4.2 信息管理界面

信息管理包括：学生信息管理，教师信息管理，课程信息管理，授课信息管理，选课信息管理，学生成绩管理。



5.4.2. 1 学生管理

该功能用于添加新学生信息，修改学生信息和删除学生信息，可以根据不同的查询方式对学生信息进行查询，既可以单个查询（按学号方式查询），还可以部分查询（年级、班级、姓名、学院等方式查询），全部显示学生信息（点击全部按钮）。

学生管理

查询类型

学号

数值

查询

学号

年级

班级

姓名

学院

专业

性别

年龄

电话

地址

学号	年级	班级	姓名	学院	专业	性别	年龄	电话	地址		
220800...	08	04	阮茂为	...	软件学院	软件工程	女	21		福建	
220800...	08	04	胡雪莉	...	软件学院	软件工程	女	22	189504...	福建	
220800...	20	04	李碧兰	...	软件学院	软件工程	女	20		福建	
220800...	08	04	陈雅莉	...	8888	软件学院	软件工程	女	21		福建
220800...	08	04	卢桂芳	...	8888	软件学院	软件工程	女	21	136450...	福建
220900...	09	03	陆建新	...	8888	软件学院	软件工程	男	20		福建

添加

修改

删除

全部

返回

添加学生信息和修改学生信息的界面(注:修改学生信息时需要选中一行,否则会弹出提示框,删除学生信息的时候也要选中一行)

添加学生

学号

220900301

年级

09

班级

03

姓名

卢建兴

密码

8888

学院

软件学院

专业

软件工程

性别

男

年龄

20

电话

地址

福建

添加

取消

修改学生

学号

220800418

年级

08

班级

04

姓名

陈雅莉

密码

8888

学院

软件学院

专业

软件工程

性别

女

年龄

21

电话

地址

福建

修改

取消

提示窗口:



查询方式有多种选择，点击下拉式菜单选择其中一种，在右边数值文本框上选择你所要查询的信息，点击查询即可查到你所要的信息：

查询类型

学号

数值

查询

学号	姓名	学院	专业	性别	年龄	电
阮茂芳	...	软件学院	软件工程	女	21	
胡雪颖	...	软件学院	软件工程	女	22	18950
李碧玉	...	软件学院	软件工程	女	20	
陈雅莉	...	软件学院	软件工程	女	21	

5.4.2. 2 教师管理

教师管理的基本界面、实现的功能等与学生管理的几乎一样！功能有添加新的教师信息，修改教师信息和删除教师信息，可以根据不同的查询方式对教师信息进行查询，既可以单个查询（按教师编号方式查询），还可以部分查询（姓名、职称等方式查询），全部显示教师信息（点击全部按钮）。

教师管理

查询类型

编号

数值

查询

编号

姓名

性别

年龄

职称

密码

性别

年龄

职称

220001	邓新国	8888	男	40	副教授
220002	江兰芳	8888	女	30	讲师
220003	张舒	8888	女	30	讲师
220004	黄岩	8888	男	35	讲师
220005	游帆	8888	男	30	讲师
220006	杨艳华	8888	女	40	讲师

添加

修改

删除

全部

返回

添加教师信息和修改教师信息的界面：（注：修改教师信息时需要选中一行，否则会弹出提示框，删除教师信息的时候也要选中一行）

添加教师

编号

姓名

密码

性别

年龄

职称

添加

取消

修改教师

编号

姓名

密码

性别

年龄

职称

修改

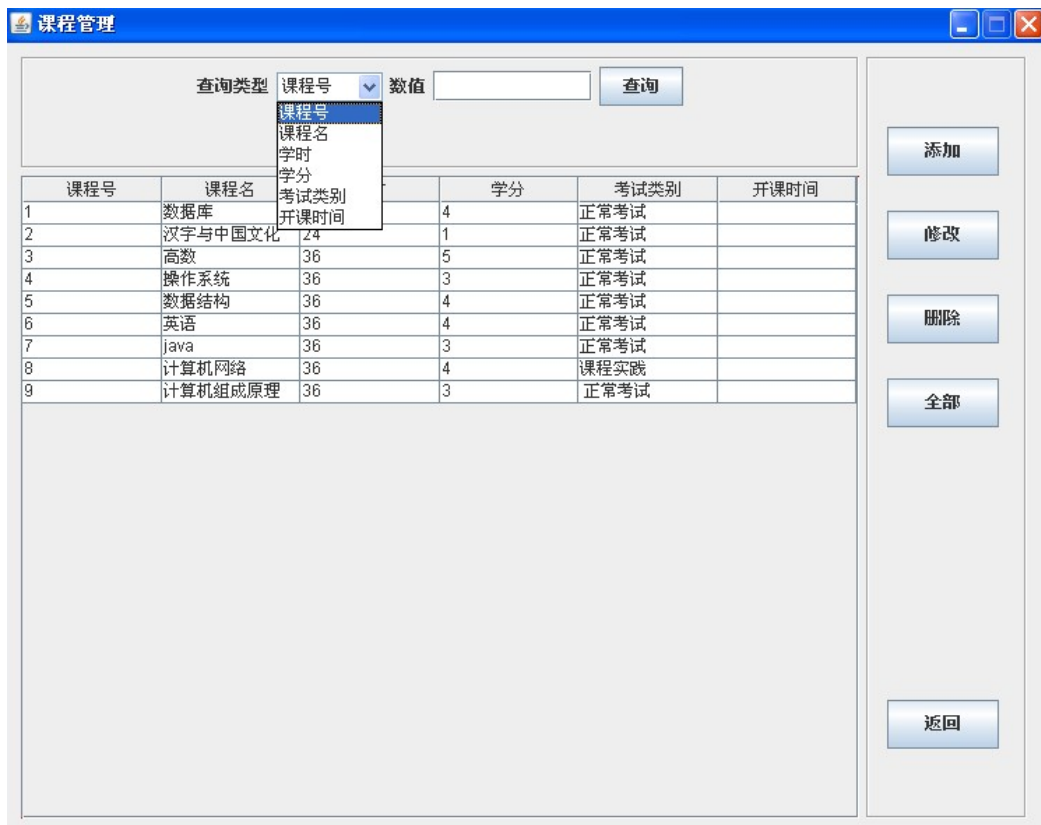
取消

提示窗口：



5.4.2.3 课程管理

该功能用于添加新课程，修改已有的课程信息和删除课程信息。同时也可以对课程信息的查询，单个查询（按课程号方式查询），还可以部分查询（课程名、学时等方式查询），全部显示课程信息（点击全部按钮）。



添加课程信息和修改课程信息界面（注：修改课程信息时需要选中一行，否则会弹出提示框，删除课程信息的时候也要选中一行）

添加课程

课程号

课程名

学时

学分

考试类别

开课时间

添加

取消

修改课程

课程号

2

课程名

汉字与中国文化

学时

24

学分

1

考试类别

正常考试

开课时间

修改

取消

提示窗口：

消息

i

请选择一行

确定

5.4.2. 4 授课管理

该功能是针对教师开课信息的管理。用于添加授课、修改授课，删除授课，授课信息查询，同样可以选择不同的方式查询（教师号、教师名、课程号、课程名）

授课管理

查询类型 教师号 数值 查询

教师号
教师名
课程号
课程名

教师号	教师名	课程号	课程名
220001	邓新国	1	数据库
220001	邓新国	5	数据结构
220002	江兰芳	4	操作系统
220002	江兰芳	9	计算机组成原理
220003	张舒	7	java
220004	黄岩	3	高数
220005	游帆	6	英语
220005	游帆	8	计算机网络
220006	杨艳华	2	汉字与中国文化

添加

修改

删除

全部

返回

查询某个老师的授课信息：

查询类型 教师名 数值 张舒 查询

教师号	教师名	课程号	课程名
220003	张舒	7	java

添加授课和修改授课信息的界面（注：修改授课信息时需要选中一行，否则会弹出提示框，删除授课信息的时候也要选中一行），其中教师号必须是在教师管理中已经存在的，课程号是在课程管理中已经存在的，若不存在，则添加和修改都不成功，需要返回的相应的管理中进行添加。

The image shows two side-by-side windows from a software application. The left window is titled '添加授课' (Add Teaching) and contains fields for '教师号' (Teacher ID) with a dropdown menu showing '220001', '课程号' (Course ID) with a dropdown menu showing '1', '教师名' (Teacher Name) with a text box containing '邓新国', and '课程名' (Course Name) with a text box containing '数据库'. At the bottom are '添加' (Add) and '取消' (Cancel) buttons. The right window is titled '修改授课' (Modify Teaching) and contains fields for '教师号' (Teacher ID) with a dropdown menu showing '220002', '课程号' (Course ID) with a dropdown menu showing '4', '教师名' (Teacher Name) with a text box containing '江兰芳', and '课程名' (Course Name) with a text box containing '操作系统'. At the bottom are '修改' (Modify) and '取消' (Cancel) buttons.

提示窗口：

The image shows a small message dialog box titled '消息' (Message). It contains an information icon (a lowercase 'i' inside a circle) and the text '请选择一行' (Please select a row). At the bottom is a '确定' (OK) button.

5.4.2.5 选课管理

该功能是针对所有学生选课信息的管理。用于添加选课、包括修改选课，删除选课，选课信息查询，同样可以选择不同的方式查询（课程号、课程名、教师号、授课教师、学生学号、学生名）

选课管理

查询类型: 课程号 数值: 查询

课程号
课程名
教师号
选课老师
学号
学生名

课程号	课程名	教师号	选课老师	学号	学生名
1	数据库	220001	邓新国	220800407	阮茂芳
1	数据库	220001	邓新国	220800409	胡雪钦
1	数据库	220001	邓新国	220800411	李碧玉
1	数据库	220001	邓新国	220800418	陈雅莉
1	数据库	220001	邓新国	220800424	卢桂芳
2	汉字与中国文化	220006	杨艳华	220800407	阮茂芳
2	汉字与中国文化	220006	杨艳华	220800409	胡雪钦
2	汉字与中国文化	220006	杨艳华	220800411	李碧玉
2	汉字与中国文化	220006	杨艳华	220800418	陈雅莉
2	汉字与中国文化	220006	杨艳华	220800424	卢桂芳
3	高数	220004	黄岩	220800407	阮茂芳
3	高数	220004	黄岩	220800409	胡雪钦
3	高数	220004	黄岩	220800411	李碧玉
3	高数	220004	黄岩	220800418	陈雅莉
3	高数	220004	黄岩	220800424	卢桂芳
4	操作系统	220002	江兰芳	220800407	阮茂芳
4	操作系统	220002	江兰芳	220800409	胡雪钦
4	操作系统	220002	江兰芳	220800411	李碧玉
4	操作系统	220002	江兰芳	220800418	陈雅莉
4	操作系统	220002	江兰芳	220800424	卢桂芳
5	数据结构	220001	邓新国	220800407	阮茂芳
5	数据结构	220001	邓新国	220800409	胡雪钦
5	数据结构	220001	邓新国	220800411	李碧玉
5	数据结构	220001	邓新国	220800418	陈雅莉
5	数据结构	220001	邓新国	220800424	卢桂芳
6	英语	220005	游帆	220800407	阮茂芳
6	英语	220005	游帆	220800409	胡雪钦
6	英语	220005	游帆	220800411	李碧玉

添加 修改 删除 全部 返回

查询某个学生的选课情况:

选课管理

查询类型: 学号 数值: 220800418 查询

课程号	课程名	老师号	选课老师	学号	学生名
1	数据库	220001	邓新国	220800418	陈雅莉
2	汉字与中国文化	220006	杨艳华	220800418	陈雅莉
3	高数	220004	黄岩	220800418	陈雅莉
4	操作系统	220002	江兰芳	220800418	陈雅莉
5	数据结构	220001	邓新国	220800418	陈雅莉
6	英语	220005	游帆	220800418	陈雅莉
7	java	220003	张舒	220800418	陈雅莉
8	计算机网络	220005	游帆	220800418	陈雅莉
9	计算机组成原理	220002	江兰芳	220800418	陈雅莉

添加 修改 删除 全部

添加选课和修改选课信息的界面（注：修改选课信息时需要选中一行，否则会弹出提示框，删除选课信息的时候也要选中一行），其中课程号是在课程管理中已经存在的，教师号必须是在教师管理中已经存在的，学号是学生管理中已经存在的。若不存在，则添加和修改都不成功，需要返回的相应的管理中进行添加。

课程号	课程名	老师号	开课老师	学号	学生名
3	高数	220004	黄岩	220800409	胡雪钦

课程号	课程名	老师号	开课老师	学号	学生名
8	计算机网络	220005	游帆	220800418	陈雅莉

提示信息：

消息

请选择一行

确定

5.4.2.6 成绩管理

该功能用于添加、修改已修课程的学生成绩（权限是所有学生），也可按照不同方式查询成绩单个查询（按学号方式查询），还可以部分查询（课程名、学分等方式查询），全部显示学生信息（点击全部按钮）。

成绩查询

查询类型: 开课老师 数值: 查询

课程号 学号 姓名 课程名 学分 成绩 考试类别 开课时间 教师号 开课老师

1	220800...	阮茂芳	成绩	83	正常考...	220001	邓新国
1	220800...	胡雪钦	成绩	80	正常考...	220001	邓新国
1	220800...	李碧玉	成绩	85	正常考...	220001	邓新国
1	220800...	陈雅莉	成绩	80	正常考...	220001	邓新国
1	220800...	卢桂芳	成绩	90	正常考...	220001	邓新国
2	220800...	阮茂芳	汉字与...	80	正常考...	220006	杨艳华
2	220800...	胡雪钦	汉字与...	90	正常考...	220006	杨艳华
2	220800...	李碧玉	汉字与...	81	正常考...	220006	杨艳华
2	220800...	陈雅莉	汉字与...	85	正常考...	220006	杨艳华
2	220800...	卢桂芳	汉字与...	88	正常考...	220006	杨艳华
3	220800...	阮茂芳	高数	85	正常考...	220004	黄岩
3	220800...	胡雪钦	高数	70	正常考...	220004	黄岩
3	220800...	李碧玉	高数	90	正常考...	220004	黄岩
3	220800...	陈雅莉	高数	70	正常考...	220004	黄岩
3	220800...	卢桂芳	高数	80	正常考...	220004	黄岩
4	220800...	阮茂芳	操作系统	80	正常考...	220002	江兰芳
4	220800...	胡雪钦	操作系统	81	正常考...	220002	江兰芳
4	220800...	李碧玉	操作系统	82	正常考...	220002	江兰芳
4	220800...	陈雅莉	操作系统	83	正常考...	220002	江兰芳
4	220800...	卢桂芳	操作系统	85	正常考...	220002	江兰芳
5	220800...	阮茂芳	数据结构	85	正常考...	220001	邓新国
5	220800...	胡雪钦	数据结构	85	正常考...	220001	邓新国
5	220800...	李碧玉	数据结构	80	正常考...	220001	邓新国
5	220800...	陈雅莉	数据结构	80	正常考...	220001	邓新国
5	220800...	卢桂芳	数据结构	88	正常考...	220001	邓新国
6	220800...	阮茂芳	英语	80	正常考...	220005	游帆
6	220800...	胡雪钦	英语	78	正常考...	220005	游帆
6	220800...	李碧玉	英语	83	正常考...	220005	游帆
6	220800...	陈雅莉	英语	80	正常考...	220005	游帆
6	220800...	卢桂芳	英语	78	正常考...	220005	游帆

添加 修改 删除 全部 返回

查询某个课程的所有学生成绩：（课程号为六）

成绩查询

查询类型: 课程号 数值: 6 查询

课程号 学号 姓名 课程名 学分 成绩 考试类别 开课时间 教师号 开课老师

6	220800...	阮茂芳	英语	4	80	正常考试...	220005	游帆
6	220800...	胡雪钦	英语	4	78	正常考试...	220005	游帆
6	220800...	李碧玉	英语	4	83	正常考试...	220005	游帆
6	220800...	陈雅莉	英语	4	80	正常考试...	220005	游帆
6	220800...	卢桂芳	英语	4	78	正常考试...	220005	游帆

添加 修改 删除 全部

查询某个学分的所有课程的学生成绩：（3 学分）

成绩查询

查询类型

学分

数值

3

查询

课程号	学号	姓名	课程名	学分	成绩	考试类别	开课时间	教师号	开课老师
4	220800...	阮茂芳	操作系统	3	80	正常考试...		220002	江兰芳
4	220800...	胡雪钦	操作系统	3	81	正常考试...		220002	江兰芳
4	220800...	李碧玉	操作系统	3	82	正常考试...		220002	江兰芳
4	220800...	陈雅莉	操作系统	3	83	正常考试...		220002	江兰芳
4	220800...	卢桂芳	操作系统	3	85	正常考试...		220002	江兰芳
7	220800...	阮茂芳	java	3	80	正常考试...		220003	张舒
7	220800...	胡雪钦	java	3	80	正常考试...		220003	张舒
7	220800...	李碧玉	java	3	81	正常考试...		220003	张舒
7	220800...	陈雅莉	java	3	82	正常考试...		220003	张舒
7	220800...	卢桂芳	java	3	83	正常考试...		220003	张舒

添加

修改

删除

全部

添加成绩和修改成绩信息的界面（注：修改成绩信息时需要选中一行，否则会弹出提示框，删除信息的时候也要选中一行），其中课程号是在课程管理中已经存在的，学号是学生管理中已经存在的。而且只能添加学生有选的课程的成绩。若不存在，则添加和修改都不成功。添加成绩只能选择课程号、学号和成绩，修改成绩只能修改成绩，其他信息均不能修改

添加成绩

课程号

1

学号

220800409

课程名

数据库

姓名

胡雪钦

学分

4

成绩

考试类别

正常考试

开课时间

老师号

220001

开课老师

邓新国

添加

取消

修改成绩

课程号

1

学号

220800418

课程名

数据库

姓名

陈雅莉

学分

4

成绩

80

考试类别

正常考试

开课时间

老师号

220001

开课老师

邓新国

修改

取消

5.5 帮助信息

5.5.1 关于本系统



5.5.2 联系方式



Chapter6 总结

实践活动结束了，站在实践阶段的末端，回顾实践过程的点点滴滴，虽然说不上千辛万苦，但是毕竟我们为此付出了诸多的心血。当实践正式开始时我三个便闪电组成实践小分队为接下来的实践活动做好准备。组队结束后当天晚上我们就召开了小组会议就有关实践的内容进行讨论。我们将整个实践环节分为三个阶段。第一个阶段为需求分析期，为期 5 天，首先我们就实践要用到的知识各自进行相应的学习，学习期间保持良好的互动，之后一起讨论本程序的项目功能设计和数据库设计。第二个阶段为编程期，为期 15 天，这期间我们进行细致的分工，一人负责数据库的建立和连接，一人负责总界面的实现，一个负责增删改查等操作的实现，在这期间，三人都保持了良好的互动与合作。第三个阶段为完善期，在这个阶段我们就完成的程序进行测试，对程序的漏洞进行修补改进，尽量将程序做到尽善尽美。做好了程序，剩下就只有程序文档了，我们也将文档的内容进行分块，分工共同来完成。

在本次数据库课程设计实践中，虽然顺利完成了整个学生成绩管理系统。但是系统中仍然存在一些问题，比如，在管理员和教师成绩录入模块中，学生成绩只能一条一条录入，毫无疑问，这样做效率很低。要提高效率，就必须做到成绩的成批录入，如按班级录入。另一方面，整个系统的界面设计不够美观，这主要是因为对 GUI 的布局处理的不够好。

在整个系统的开发设计过程，我们从中学到了许多东西。最重要的一点就是我们认识到了亲自动手好处！以前总是觉得自己学得不够好，掌握得不够深，不相信自己能开发出类似的系统。其实书本上的东西觉得再好，但也只有通过实践，亲自动手做过以后才能真正的掌握，遇到问题才知道如何下手。在开发过程中要不断的调整自己的思维，要善于变通，不能钻牛角尖，否则将陷入死胡同。遇到问题，要及时地查阅资料，如上网到专门的技术论坛询问等，在这次开发过程中，我们从网上获得了极大的帮助，几个技术难点都是通过上网查资料解决的。

另外一点很重要，在开发系统之前必须确保做好充分的需求分析，否则后面的开发工作难以进行。一个完善的系统应该有一个合理的后台数据库，所以，我们必须把数据库设计放在一个很重要的位置，不断地将数据库完善以后才能开始编写程序。开发过程中最忌一边写程序一边修改数据库，这样开发出来的系统极易瘫痪。

系统开发的过程是一个团队合作、沟通、互相协助的过程。团队合作、默契沟通以及相互协助是开发成功的保证。团队成员的任务分工要明确。在本次开发过程中，本组成员之间的配合比较默契。

总之，实践才能出真知，只有多动手，才能掌握得更多。这是本次课设中我体会最深的！以后我们将会花更多的时间和精力在这方面。相信我们一定可以做一个优秀的软件开发者的！

参考文献

- [1] Cay S. Horstmann Cray Cornell 《Java 核心技术》(第八版). 机械工业出版社, 2010. 2
- [2] 宋坤等 《Visual C++数据库系统开发案例精选》. 人民邮电出版社, 2006. 8
- [3] 侯晓强等 《精通 Java2》. 科学出版社, 2003. 11
- [4] 廖望等 《SQL Server2000 案例教程》. 冶金工业出版社, 2004. 4
- [5] 王珊 萨师煊 《数据库系统概论》(第四版). 高等教育出版社, 2006. 5