



東南大學

Southeast University

网络空间安全学院

暑期学校报告

课程名称: QT-C++语言课程设计

项目名称: 学生信息管理系统

年级专业: 20-网络空间安全

学生姓名: 周杰

学号: 57120204

日期 2022 年 8 月 28 日

学生信息管理系统

1. 需求分析

学生信息管理系统是一个学校里面十分重要的一个系统，对于学生个人查询个人信息，查询个人成绩，管理个人密码；对于管理员管理全体学生信息，群体学生成绩等等具有重要作用。此系统学生端与管理员端是互通的，管理员端可修改学生信息，一旦管理员端修改后，学生端就可查看自己被修改的信息。这样可以使两端信息统一，节省时间，提高工作效率。

本项目在分析资料的基础上，使用 QT Creator 为开发工具，以 SQLITE 为数据库后台，设计并实现了一个学生信息管理系统。此系统包括学生端，管理员端；学生端主要包括自我

信息展示，自我成绩展示，自我密码修改，教师端主要包括学生信息展示，添加学生，成绩展示，添加成绩，修改管理员密码，修改学生信息，修改学生密码功能。

学生信息管理是学校中十分重要的一项事务，学校学生信息管理一般十分庞大，涉及面广，信息冗杂，巨量。一般的手工管理是无法满足任务的。因此学生管理的全面信息化是十分重要的部分。学生信息管理系统是一项十分重要的项目，且具有现实意义。

学生信息管理系统具有如下几个特点：

- (1) 实现对数据库中学生的数据进行增、删、改、查等基本操作；
- (2) 实现信息录入，即学生录入，学生成绩录入。
- (3) 实现数据基本纠错功能。账号位数，是否为全数字的判定，密码位数判断，清除输入时的空格等等。

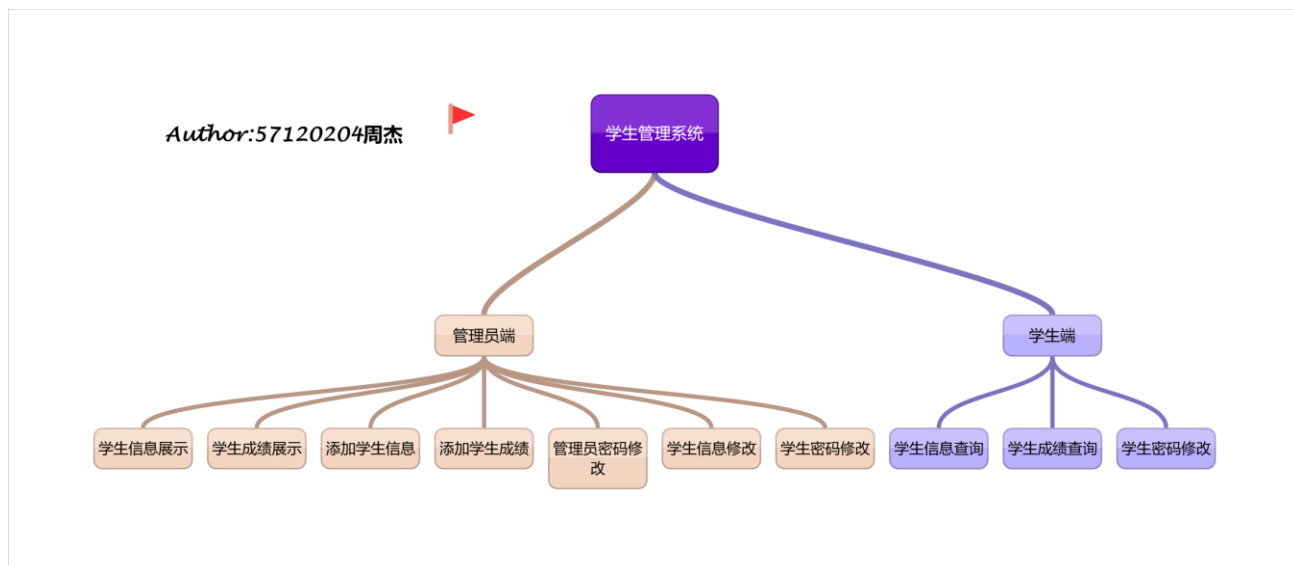
2. 概要设计

2.1 设计任务及要求

本次设计的主要功能要求：

- (1) 可以对学生信息的存入、取出等操作；
- (2) 学生信息的查询；
- (3) 学生以及管理员数据的添加，修改及删除操作。

2.2 总体方案设计



3. 详细设计 && 技术实现 && 软件测试

本系统的数据库名字为 custom2.db, 共有四张数据库表, 分别为 admintable, gradetable, studentpasstable, stutable (即管理员表, 成绩表, 学生密码表, 学生信息表)。

所以 sql 语句均已植入代码, 会在程序第一次运行时创造数据库 custom2.db。创建 4 个表格, 基础表格如下:

网络空间安全学院 暑校报告

对象		admintable @main (2222) - 表	gradetable @main (2222)
开始事务		文本	筛选
		排序	导入
		导出	
		adid	adpass
		00001	12345678
		00002	12345678

图 3.1 admintable 表

gradetable @main (2222) - 表 - Navicat Premium	
文件 编辑 查看 表 收藏夹 工具 窗口 帮助	
连接 新建查询 表 视图 A-Z 索引 触发器 用户 查询 报表 备份 自动运行 模型	
mysql studentmanager	
2222	
main	
表	
admintable	
gradetable	
studentpasstable	
stutable	
视图	
A-Z 索引	
触发器	
查询	
报表	
备份	
对象	
admintable @main (2222) - 表	
gradetable @main (2222) - 表	
studentpasstable @main (2222) - 表	
stutable @main (2222) - 表	
开始事务	
文本	
筛选	
排序	
导入	
导出	
stdid	
grademath	
gradeen	
gradec	
000001	
84	
86	
72	
000011	
15	
45	
100	
000102	
30.5	
74	
99	

图 3.2 gradetable 表

mysql studentmanager	
2222	
main	
表	
admintable	
gradetable	
studentpasstable	
stutable	
视图	
A-Z 索引	
触发器	
对象	
admintable @main (2222) - 表	
gradetable @main (2222) - 表	
studentpasstable @main (2222) - 表	
stutable @main (2222) - 表	
开始事务	
文本	
筛选	
排序	
导入	
导出	
stdid	
spass	
000001	
12345678	
000102	
000000	
000011	
99999	

图 3.3 studentpasstable 表

mysql studentmanager	
2222	
main	
表	
admintable	
gradetable	
studentpasstable	
stutable	
视图	
A-Z 索引	
触发器	
对象	
admintable @main (2222) - 表	
gradetable @main (2222) - 表	
studentpasstable @main (2222) - 表	
stutable @main (2222) - 表	
开始事务	
文本	
筛选	
排序	
导入	
导出	
stdid	
sname	
ssex	
sage	
splace	
smajor	
000001	
张三	
男	
20	
四川省成都市郫都区	
计算机与软件工程学院	
000102	
李四	
男	
19	
上海市浦东新区	
管理学院会计学2班	
000011	
李娜	
女	
20	
北京市朝阳区	
理学院信息与计算科学	

图 3.4 stutable 表

数据库的创建

```
void LogScreen::createdatabase()
```

```

{

    //判断是否存在
    QString dbName="custom2.db";
    QFileInfo info(dbName);
    bool isExists=info.exists();
    QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");
    db.setHostName(dbHostName);           //选择本地主机
    db.setDatabaseName(dbName);           //设置数据源名称
    db.setUserName(dbusername);           //登录用户
    db.setPassword(dbpassword);           //密码
    bool isOpen=db.open();
    if(isOpen)
    {

        //不存在数据库，创建表
        if(!isExists)
        {
            QSqlQuery query(db);
            bool ok1=query.exec("CREATE TABLE admintable ( adid nchar
( 5 ) PRIMARY KEY NOT NULL, adpass nchar ( 20 ) NOT NULL );");
            bool ok2=query.exec("INSERT INTO admintable ( adid, adpass )
VALUES ( '00001', '12345678' ), ( '00002', '12345678' );");
            bool ok3=query.exec("CREATE TABLE studentpasstable ( stid
char ( 6 ) PRIMARY KEY NOT NULL, spass nchar ( 20 ) NOT NULL );");
            bool ok4=query.exec("INSERT INTO studentpasstable ( stid,
spass )"
                                "VALUES"
                                "( '000001', '12345678' ),"
                                "( '000102', '000000' ), "
                                "( '000011', '99999' ); ");
            bool ok5=query.exec("CREATE TABLE stutable ("
                                "stid char ( 6 ) PRIMARY KEY NOT NULL,"
                                "sname char ( 5 ) NOT NULL,"
                                "ssex nchar ( 1 ),"
                                "sage tinyint,"
                                "splace nchar ( 50 ),"
                                "smajor nchar ( 50 )"
                                ");");
            bool ok6=query.exec("INSERT INTO stutable ( stid, sname,
ssex, sage, splace, smajor )"
                                "VALUES"

```

```
        "( '000001', '张三', '男', '20', '四川省成都市  
郫都区', '计算机与软件工程学院计算机科学与技术2班' ),"  
        "( '000102', '李四', '男', '19', '上海市浦东新  
区', '管理学院会计学2班' ),"  
        "( '000011', '李娜', '女', '20', '北京市朝阳区  
, '理学院信息与计算科学1班' );");  
    bool ok7 = query.exec("CREATE TABLE gradetable ("  
        "stid nchar ( 6 ) NOT NULL,"  
        "grademath numeric ( 4, 1 ) NOT NULL,"  
        "gradeen numeric ( 4, 1 ) NOT NULL,"  
        "gradec numeric ( 4, 1 )"  
        ");");  
    bool ok8 = query.exec("INSERT INTO gradetable ( stid,  
grademath, gradeen, gradec )"  
        "VALUES"  
        "( '000001', '84', '86', '72' ),"  
        "( '000011', '15', '45', '100' ),"  
        "( '000102', '30.5', '74', '99' )");  
  
    if(ok1 && ok2 && ok3 && ok4 && ok5 && ok6 && ok7 && ok8)  
        qDebug()<<"创建表成功";  
    else qDebug()<<"创建表失败";  
}  
else qDebug()<<"表已存在";  
}  
}
```

一些公有函数的实现:

①去除输入输出时的空格

```
QString removeSpace(QString str)  
{  
    QString ans = "";  
    for(int i = 0; i < str.length(); i++)  
    {  
        if(str[i] != ' ')  
            ans+=str[i];  
        else continue;  
    }  
    return ans;  
}
```

②检查学生账号是否正确

```
bool isProperId(QString str)  
{  
    if(str.length() != 6)  
        return false;
```

```

else
{
    QByteArray ba = str.toLatin1();//QString 转换为 char*
    const char *s = ba.data();
    while(*s && *s >= '0' && *s <= '9') s++;

    if (*s)
    { //不是纯数字
        return false;
    }
    else
    { //纯数字
        return true;
    }
}
}

③检查密码是否正确
bool isProperPassword(QString str)
{
    //密码位数必须要大于等于 6 位，小于等于 20 位
    if(str.length()>=6 && str.length()<=20)
        return true;
    else return false;
}

```

3. 1 登陆界面

3.1.1在登陆界面进行登录



图 3.5 登陆界面

“清空”按钮的实现:

```
void LogScreen::on_pushButton_clear_clicked()
```

```

{
    ui->lineEdit_id->setText("");
    ui->lineEdit_password->setText("");
}

“登录”按钮的实现：
void LogScreen::on_pushButton_log_clicked()
{
    QString getId = removeSpace(ui->lineEdit_id->text());
    QString getPass = removeSpace(ui->lineEdit_password->text());
    bool flag_pos;//管理员为1，学生为0

    if(ui->radioButton->isChecked())
        flag_pos = 1;
    else flag_pos = 0;
    if(checkpassword(flag_pos, getId, getPass) && getId != "")
    {
        userLogId = getId;

        qDebug() << "userLogId1:" << userLogId;
        this->hide();
        if(flag_pos == 1)
        {
            AdminScreen * adminscreen = new AdminScreen;
            adminscreen->show();
            connect(adminscreen, &AdminScreen::backToLoginScreen, [=]() {
                this->show();
                adminscreen->hide();
            });
        }
        else
        {
            StuScreen * stuscreen = new StuScreen;
            stuscreen->show();
            connect(stuscreen, &StuScreen::backToLoginScreen, [=]() {
                this->show();
                stuscreen->hide();
            });
        }
    }
    else
    {

```

```
        QMessageBox::information(this, "提示", "登陆失败, 可能的原因:\n 1. 账号错误\n 2. 密码错误\n 若忘记密码, 请及时联系学校管理员修改");
    }
```

3.2 学生界面

3.2.1

学生界面展示



图 3.6 学生界面

3.2.2 信息查询按钮的实现

```
void StuScreen::on_pushButton_selectInfo_clicked()
{
    this->hide();
    stuinfoshowself->show();
}
```

修改密码, 成绩查询实现方法同理, 此处省略。

3.2.3 返回按钮的实现

① 点击返回按钮, 发射信号

```
void StuScreen::on_pushButton_back_clicked()
{
    emit this->backToLoginScreen();
}
```

② 登陆界面接收信号

```
connect(stuscreen, &StuScreen::backToLoginScreen, [=]() {
    this->show();
    stuscreen->hide();
});
```

3.2.4 界面的实现


```
void StuScreen::initScreen()
{
    //获取姓名
    QString sname;
    QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");
    db.setHostName(dbHostName);           //选择本地主机
    db.setDatabaseName("custom2.db");      //设置数据源名称
    db.setUserName(dbusername);           //登录用户
    db.setPassword(dbpassword);           //密码
    bool isOpen = db.open();
    if(isOpen)
    {
        qDebug() << "打开了数据库";
        QSqlQuery query(db);

        bool ok = query.exec("select sname from stutable where
stdid='"+userLogId+" '");
        qDebug() << "OK=" << ok;
        query.next();
        sname = query.value(0).toString();

    }
    else qDebug() << "查询姓名失败";
    qDebug() << "学生姓名为: " << sname;
    setWindowTitle("学生界面");
    setFixedSize(500, 333);
    ui->lineEdit_welcome->setStyleSheet("border :1px ;background-color:
rgba(0,0,0,0);color:#10E0F0");
    ui->lineEdit_welcome->setText("欢迎"+ sname +"同学");
    ui->lineEdit_welcome->setFont(QFont("Timers",28,QFont::Bold));
    setWindowIcon(QIcon(":/imag/icon.jpg"));

    ui->pushButton_modifyPass->setStyleSheet("border :1px ;background-col
or: rgba(0,0,0,100);color:#F2390E;height:60px");

    ui->pushButton_selectGrade->setStyleSheet("border :1px ;background-co
lor: rgba(0,0,0,100);color:#F2390E;height:60px");

    ui->pushButton_selectInfo->setStyleSheet("border :1px ;background-col
or: rgba(0,0,0,100);color:#F2390E;height:60px");
    ui->pushButton_back->setStyleSheet("border :1px ;background-color:
rgba(0,0,0,100);color:#F2390E;height:60px");
```

```

        ui->pushButton_modifyPass->setFont(QFont("微软雅黑", 11, QFont::Bold));
        ui->pushButton_selectGrade->setFont(QFont("微软雅黑", 11, QFont::Bold));
        ui->pushButton_selectInfo->setFont(QFont("微软雅黑", 11, QFont::Bold));
        ui->pushButton_back->setFont(QFont("微软雅黑", 11, QFont::Bold));
    }

```

3.2.5 学生个人信息展示



图 3.6 学生个人信息展示

代码实现(只展示主要部分)

① 从数据库获取

```

QSqlQuery query(db);
    query.exec("select sname, ssex, sage, splace, smajor from stutable
where stid='"+userLogId+"'");
    query.next();
    //转换为无空格 str
    sname = removeSpace(query.value(0).toString());
    ssex = removeSpace(query.value(1).toString());
    sage = removeSpace(query.value(2).toString());
    splace = removeSpace(query.value(3).toString());
    smajor = removeSpace(query.value(4).toString());

```

② 展示

```

ui->lineEdit_age->setText(sage);
ui->lineEdit_id->setText(userLogId);
ui->lineEdit_major->setText(smajor);
ui->lineEdit_place->setText(splace);
ui->lineEdit_sex->setText(ssex);
ui->lineEdit_name->setText(sname);

```

3.2.6 学生个人成绩查询



图 3.7 学生个人成绩查询

主要代码实现:

```
if(isOpen)
{
    QSqlQuery query(db);
    query.exec("select * from gradetable where
stdid='"+userLogId+"'");
    query.next();
    math = removeSpace(query.value(1).toString());
    English = removeSpace(query.value(2).toString());
    Cplusplus = removeSpace(query.value(3).toString());

    avgGrade=(math.toFloat()+english.toFloat()+cplusplus.toFloat())/3;
}
else qDebug()<<"打开数据库失败";
ui->lineEdit->setText(math);
ui->lineEdit_2->setText(english);
ui->lineEdit_3->setText(cplusplus);
ui->lineEdit_4->setText(convert(avgGrade));
```

3.2.7 个人密码修改



图 3.8 学生个人密码修改

主要代码实现

```
if(isOpen)
{
    QSqlQuery query(db);
    query.exec("select spass from studentpasstable where
stid='"+userLogId+"'");
    query.next();
    QString oldP = query.value(0).toString();
    if(oldP == getOldPass)
    {
        query.exec("update studentpasstable set spass ='"+ newPass
+"'' where stid='"+ userLogId +"'");
        query.next();
        QMessageBox::information(this,"提示","修改密码成功!");
    }

    else QMessageBox::critical(this,"警告","初始密码输入错误! \n
请重新输入! \n 若忘记密码, 请联系学校管理员修改");
}

else qDebug()<<"数据库打开失败, 修改失败!";
```

3.3 管理员界面

3.3.1 管理员界面展示（各按钮实现与上学生界面基本一致，此处省略）



图 3.9 管理员界面展示

初始化界面代码：

① 标题程序框大小的实现

```
void AdminScreen::iniScene()
{
    setWindowIcon(QIcon(":/imag/icon.jpg"));
    setWindowTitle("管理员操作界面");
    setFixedSize(500, 333);
}
```

② 背景的实现

```
void AdminScreen::paintEvent(QPaintEvent *event)
{
    QPainter painter(this);

    painter.drawPixmap(0, 0, 500, 333, QPixmap(":/imag/background4.jpg"));
}
```

3.3.2 学生信息展示

	学号	姓名	性别	年龄	籍贯	专业
1	000001	张三	男	20	四川省成都市郫都区	计算机与软件工程学院计算机科学与技术2班
2	000102	李四	男	19	上海市浦东新区	管理学院会计学2班
3	000011	李娜	女	20	北京市朝阳区	理学院信息与计算科学1班
4	571211	陈六	女	17	安徽省安庆市怀宁县	理学院化学2班

图 3.10 学生信息展示

界面 tableview 实现

```
qDebug() << "database open success!";
model = new QSqlTableModel(this, db);
model->setTable("stutable"); //表名
model->setHeaderData(0, Qt::Horizontal, QVariant("学号")); //表
头

model->setHeaderData(1, Qt::Horizontal, ("姓名"));
model->setHeaderData(2, Qt::Horizontal, QVariant("性别"));
model->setHeaderData(3, Qt::Horizontal, QVariant("年龄"));
model->setHeaderData(4, Qt::Horizontal, QVariant("籍贯"));
model->setHeaderData(5, Qt::Horizontal, QVariant("专业"));
model->setEditStrategy(QSqlTableModel::OnManualSubmit); //修
改模式 All changes will be cached in the model until either submitAll()
or revertAll() is called.
model->select();
ui->tableView->setModel(model);
// ui->tableView->setColumnWidth(0, 80);
// ui->tableView->setColumnWidth(1, 80);
// ui->tableView->setColumnWidth(2, 50);
// ui->tableView->setColumnWidth(3, 50);
// ui->tableView->setColumnWidth(4, 200);
// ui->tableView->setColumnWidth(5, 300);
ui->tableView->resizeColumnsToContents();
ui->tableView->resizeRowsToContents();
确定/刷新按钮实现
```

```
void AllStuInfo::on_pushButton_3_clicked()
{
```

```
    model->submitAll();
    model->select();
}
```

取消按钮实现

```
void AllStuInfo::on_pushButton_cancel_clicked()
{
```

```
    model->revertAll();
}
```

添加按钮实现

```
void AllStuInfo::on_pushButton_add_clicked()
{
```

```
    model->insertRow(model->rowCount());
}
```

删除按钮实现

```
void AllStuInfo::on_pushButton_delete_clicked()
{
```

```
    int selectRow = ui->tableView->currentIndex().row();
```

```
    if(selectRow >= 0)
```

```
    {
```

```
        int del = QMessageBox::question(this, "提示", "是否要删除此学生  
信息\n删除后数据无法恢复, 请慎重考虑!", QMessageBox::Yes |  
QMessageBox::No, QMessageBox::Yes);
```

```
        if(del == QMessageBox::Yes)
```

```
        {
```

```
            model->removeRow(selectRow);
```

```
            model->submitAll();
```

```
        }
```

```
        else return;
```

```
    }
```

```
}
```

升序降序按钮实现

```
void AllStuInfo::on_pushButton_asc_clicked()
{
```

```
    model->sort(0, Qt::AscendingOrder);
}
```

```
void AllStuInfo::on_pushButton_desc_clicked()
{
```

```
    model->sort(0, Qt::DescendingOrder);
}
```


按钮样式及初始化的实现

```
void AllStuInfo::initScreen()
{
    setFixedSize(839, 527);
    setWindowIcon(QIcon(":/imag/icon.jpg"));
    setWindowTitle("学生信息展示");
    ui->tableView->setStyleSheet("border :1px ;background-color:
    rgba(0, 0, 0, 0)");
    ui->widget_btn->setStyleSheet("border:0px;background-color:
    rgba(0, 0, 0, 50);");

    ui->pushButton_3->setStyleSheet("height:40px;background-color:#A06091
    ");

    ui->pushButton_add->setStyleSheet("height:40px;background-color:#A060
    91");

    ui->pushButton_asc->setStyleSheet("height:40px;background-color:#A060
    91");

    ui->pushButton_back->setStyleSheet("height:40px;background-color:#A06
    091");

    ui->pushButton_cancel->setStyleSheet("height:40px;background-color:#A
    06091");

    ui->pushButton_delete->setStyleSheet("height:40px;background-color:#A
    06091");

    ui->pushButton_desc->setStyleSheet("height:40px;background-color:#A06
    091");
}
```

3.3.3 添加学生界面

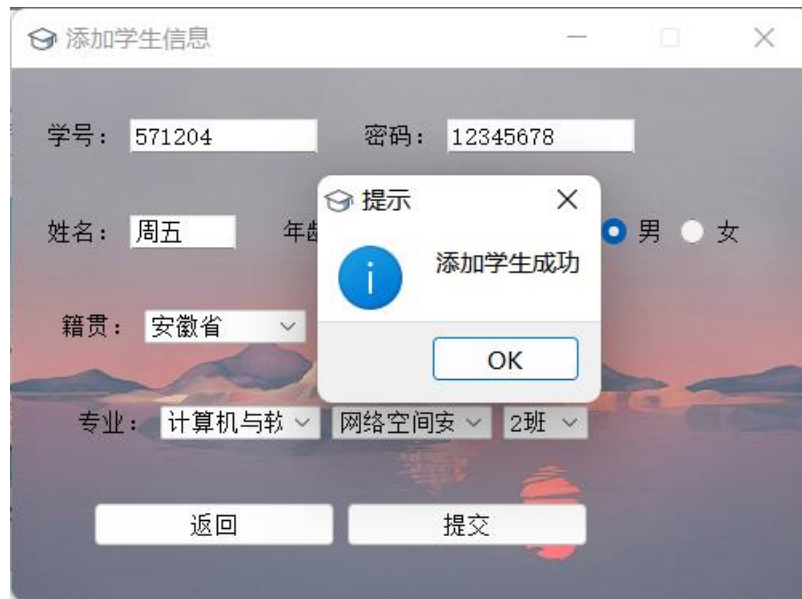


图 3.11 添加学生信息界面

添加学生实现代码

```
if(isOpen)
{
    QSqlQuery query(db);
    query.exec("select count(*) from stutable where
stid='"+sid+"'");
    query.next();
    int num;
    num = query.value(0).toInt();
    if(num)
    {
        qDebug()<<"账号已经存在";
        QMessageBox::critical(this,"警告","此学生已经存在!\n 请重新输入");
        return false;
    }
    else
    {
        query.exec("insert into studentpasstable(stid,spass)"
        "values ('"+ sid + "', '"+ spass + "')");
        query.exec("INSERT into
stutable(stid,sname,splace,ssex,smajor,sage) "
        "values ('"+ sid + "', '"+ sname + "', '"+ splace
+ "', '"+ ssex + "', '"+ smajor + "', '"+ sage + "')");
        QMessageBox::information(this,"提示","添加学生成功");
    }
}
```

}

3.3.4 学生成绩展示界面



图 3.12 学生成绩展示

TableView 代码实现

```
if(isOpen)
{
    qDebug() << "database open success!";

    QString cmd = "select stutable.stid, sname, grademath, gradeen,
gradec, gradec+gradeen+grademath from stutable, gradetable where
stutable.stid=gradetable.stid";
    proxyModel = new QSortFilterProxyModel(this);
    model->setQuery(cmd);
    proxyModel->setSourceModel(model);
    proxyModel->setHeaderData(0, Qt::Horizontal, tr("学号"));
    proxyModel->setHeaderData(1, Qt::Horizontal, tr("姓名"));
    proxyModel->setHeaderData(2, Qt::Horizontal, tr("数学"));
    proxyModel->setHeaderData(3, Qt::Horizontal, tr("英语"));
    proxyModel->setHeaderData(4, Qt::Horizontal, tr("C++"));
    proxyModel->setHeaderData(5, Qt::Horizontal, tr("总分"));

    ui->tableView->setModel(proxyModel);
    ui->tableView->show();
    ui->tableView->setColumnWidth(0, 80);
    ui->tableView->setColumnWidth(1, 60);
    ui->tableView->setColumnWidth(2, 50);
    ui->tableView->setColumnWidth(3, 50);
    ui->tableView->setColumnWidth(4, 50);
    ui->tableView->setColumnWidth(5, 50);
}
```

```
}
```

选择排序方式与升降序实现

```
void AllStuGrades::on_comboBox_activated(int index)
{
    num = index;
    //QDebug() << num;
}
```

```
void AllStuGrades::on_pushButton_asc_clicked()
{
    proxyModel->sort(num, Qt::AscendingOrder);
    qDebug() << num;
}
```

```
void AllStuGrades::on_pushButton_desc_clicked()
{
    proxyModel->sort(num, Qt::DescendingOrder);
    qDebug() << num;
}
```

刷新按钮实现

```
void AllStuGrades::on_pushButton_select_clicked()
{
    initScreen();
    studentGradesView();
}
```

添加学生成绩界面



图 3.13 添加学生成绩

添加学生成绩代码实现

```
void AddStuGrades::setGrades()
{
    QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");
    db.setHostName(dbHostName);           //选择本地主机
    db.setDatabaseName("custom2.db");      //设置数据源名称
    db.setUserName(dbusername);           //登录用户
    db.setPassword(dbpassword);           //密码
    db.open();
    QSqlQuery query(db);
    query.exec("insert into gradetable(stdid, grademath, gradeen, gradeec)
values ('"+ stdid + "','"+ math + "','"+ english + "','"+ cpp + "')");
    query.next();

    QMessageBox::information(this, "提示", "添加成绩成功");
}
```

3.3.5 修改管理员密码



图 3.14 修改管理员密码界面

修改管理员密码实现

```
QSqlQuery query(db);
query.exec("select adpass from admintable where
adid='"+userLogId+"'");
query.next();
QString oldP = query.value(0).toString();
QDebug()<<"获取到的密码是:"<<oldP;
if(oldP == getOldPass)
{
    query.exec("update admintable set adpass ='"+ newPass +"'
where adid='"+ userLogId +"'");
}
```

```

        query.next();
        QMessageBox::information(this, "提示", "密码修改成功!");
    }
    else QMessageBox::critical(this, "警告", "初始密码输入错误! \n
    请重新输入!");

```

3.3.6 修改学生信息

图 3.15 修改学生信息界面

修改学生信息代码实现

① 按学号查询

```

 QSqlQuery query(db);
    query.exec("select count(*) from stutable where
    stid='"+sid+"'");
    query.next();
    int num=query.value(0).toInt();
    if(!num)
    {
        QMessageBox::information(this, "提示", "无此学生. \n 请重新
    输入!");
        return;
    }
    else
    {
        query.exec("select sname, ssex, sage, splace, smajor from
    stutable where stid='"+sid+"'");
        query.next();
        sname = removeSpace(query.value(0).toString());
        ssex = removeSpace(query.value(1).toString());
        sage = removeSpace(query.value(2).toString());
        splace = removeSpace(query.value(3).toString());
    }

```

```

        smajor = removeSpace(query.value(4).toString());
    }

}

```

```

ui->lineEdit_age->setText(sage);
ui->lineEdit_name->setText(sname);
if(ssex == "男")
    ui->radioButton_man->setChecked(true);
else ui->radioButton_woman->setChecked(true);
ui->lineEdit_showMajor->setText(smajor);
ui->lineEdit_showPlace->setText(splace);

```

② 按修改后保存学生信息

```

void ModifyStuInfo::setInformation()
{
    sname = ui->lineEdit_name->text();
    sage = ui->lineEdit_age->text();
    splace = ui->lineEdit_showPlace->text();
    smajor = ui->lineEdit_showMajor->text();
    if(ui->radioButton_man->isChecked())
        ssex = "男";
    else ssex = "女";
    QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");
    db.setHostName(dbHostName);
    db.setDatabaseName("custom2.db");
    db.setUserName(dbusername);
    db.setPassword(dbpassword);
    bool isOpen = db.open();
    if(isOpen)
    {
        QSqlQuery query(db);
        query.exec("update stutable set
sname=' "+sname+"', ssex=' "+ssex+"', sage=' "+sage+"', "
                "splace=' "+splace+"', smajor=' "+smajor+" where
stid=' "+sid+" " ");
        query.next();
        qDebug() << sname << sage << splace << smajor << ssex;
        QMessageBox::information(this, "消息", "修改成功!");
    }
}

```

3.3.7 修改学生密码



图 3.16 修改学生密码界面

修改学生密码实现

① 查询此人的姓名

```
if(isOpen)
{
    QSqlQuery query(db);
    QString stuid = ui->lineEdit_stuid->text();
    int n;
    query.exec("select count(*) from stutable where
stid='"+stuid+"'");
    query.next();
    n = query.value(0).toInt();
    if(n)
    {
        query.exec("select sname from stutable where
stid='"+stuid+"'");
        query.next();
        ui->lineEdit_sname->setText(query.value(0).toString());
    }
    else
    {
        QMessageBox::information(this,"提示","无此学生\n 请检查学
号是否正确\n 请重新输入");
        ui->lineEdit_stuid->setText("");
        ui->lineEdit_sname->setText("");
    }
}
```

② 修改密码的实现

```
if(isOpen)
```

```
{
    QSqlQuery query(db);
    query.exec("update studentpasstable set spass='"+getPassword
+"'' where stid='"+stuid+"''");
    query.next();
    QMessageBox::information(this,"提示","密码修改成功!");
}
```

4. 课程总结

本系统学生信息管理的要求，因为自己设计，为了简单高效，使用的轻量级的数据库 Sqlite、，使用 C++语言，Qt 框架，完成学生管理系统的需求分析，系统的数据库设计和实现，系统的表单设计，主控程序设计，并由此设计了数据库结构和应用程序。系统运行结果证明，我设计的学生信息管理系统可以满足学生查询个人信息，修改个人密码；管理员对学生信息的查询，修改，添加，删除，对学生成绩的查询，添加等等。

因为完全没有数据库知识基础，不断地从网上查资料，学操作，写这个数据库大约 2 周左右，也做了两个版本，数据库由 SQL server 到 Sqlite，从第一版杂乱的布局，到第二版添加许多的背景图，修改字体，修改按钮样式等等。在完成这个项目的时候，也遇见了很多问题，比如无法查看 sqlite 的表，后来知道了 Navicat Premium 这个软件，以及后面遇到的界面数里的选项顺序要和代码中的 switch-case 的顺序对应，否则在籍贯选择的时候可能会出现安徽省-南京市乌龙；如何把登录界面的账号传给后面的界面，确认操作用户，最后通过 extern 解决，还是对 c++一些关键字不熟悉导致的；也有在成绩表添加成绩后如何刷新，查询很久没能解决最后想到重新加载等等。

总的来说，这次编程经历让我深刻体会到：不要一开始就胆怯害怕，千里之行始于足下。当迈出了第一步你会发现并没有你想想的那么难。学习编程不懂的，要学会查文档，网上查资料。自学是很重要的能力。