

# 《数据库原理及应用》 课程设计实验报告书

XX 大学计算机科学与技术学院

姓 名 banban

专 业 计 算 机 科 学 与 技 术

班 级 \_\_\_\_\_

学 号 \_\_\_\_\_

指导教师 \_\_\_\_\_

分 数	
-----	--

2022 年 6 月 12 日

## 摘 要

在当下，学校里的寝室管理系统十分重要，但许多学校的管理方式并不成熟，如果不及时改善管理，仅靠人工的方式去运作，不仅效率低下，而且错误率极高，安全性差。老师和学生服务体验会大大降低。所以，许多高校迫切需要一种能够满足这些需求的综合系统，通过计算机处理大量数据，保证高校寝室管理系统为师生带来方便。

该课题根据高校师生具体要求设计，基于 Python 和 Mysql 这两个工具的应用，完成了对用户信息的增、删、改、查功能和用户登陆、用户信息管理等的专门需求。系统运行结果证明，此寝室信息系统可以满足管理者的简单日常管理工作。该系统的核心部分是数据库的生成，利用 QT 完成了管理平台的页面设计，构造所有的类；使用 Mysql 数据库存储数据。

该系统的运行为学校带来寝室管理的准确、数据安全、节约人力、提高工作效率和质量、提高管理效率、降低错误发生几率等全方位的效益。

**关键词：**寝室管理系统，数据库，窗口应用，Python，PyQt。

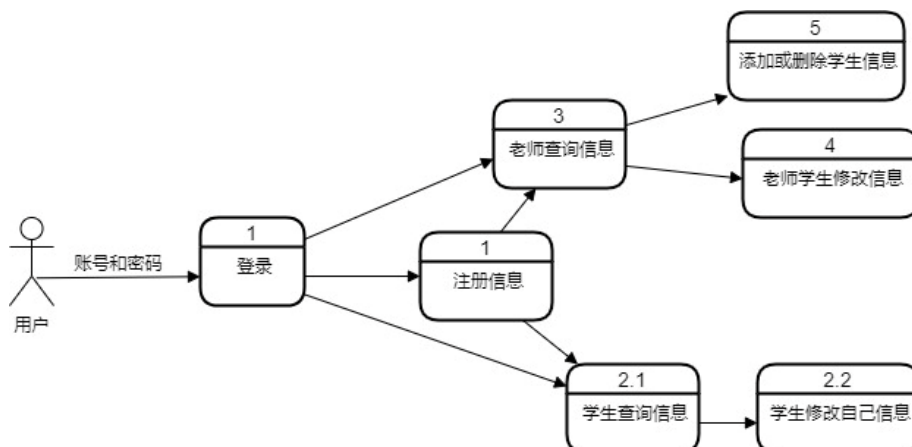
# 目录

1 系统需求分析与设计 .....	2
1.1 用户需求分析 .....	2
1.1.1 用户需求 .....	2
1.1.2 系统功能需求 .....	2
1.1.3 系统性能需求 .....	3
1.1.4 数据分析 .....	3
1.2 功能模块图及分模块功能描述 .....	4
1.2.1 系统的功能模块图 .....	4
1.2.2 系统功能模块简介 .....	4
2 数据库设计 .....	6
2.1 数据库概念设计（E-R 图） .....	6
2.1 数据库逻辑设计（关系模型） .....	6
3 系统实施 .....	8
3.1 建立数据库及其相关对象 .....	8
3.2 数据库连接 .....	12
3.3 主要模块实施 .....	12
3.3.1 登录模块的开发 .....	12
3.3.2 用户搜索模块的开发 .....	13
3.3.3 用户注册模块的开发 .....	15
3.4 系统测试 .....	16
3.4.1 软件测试的对象 .....	16
3.4.2 软件测试的结果 .....	20
4 系统说明 .....	21
4.1 开发环境 .....	21
4.2 系统安装、配置与发布应用程序的步骤 .....	21
总 结 .....	22
附录：部分源代码 .....	23

# 1 系统需求分析与设计

## 1.1 用户需求分析

### 1.1.1 用户需求



用户可以通过账号和密码，选择合适的身份进入系统里面，在系统里，老师用户可以查找每位学生用户的个人信息，并对信息进行修改，也可以添加学生用户信息，查找学生所在宿舍其他成员；学生用户可以查询自己的信息，可以查询自己室友的部分信息，也可以修改自己的信息。

### 1.1.2 系统功能需求

#### 1. 用户信息管理

老师用户可以在窗口界面中对自己学生用户信息进行修改，新增学生用户或删除学生用户。

#### 2. 修改信息

无论是学生用户还是老师用户，都可以在修改窗口界面中修改学生的个人信息。

#### 3. 注册管理

新用户可以在注册窗口界面中选择正确的身份，并填写相对应的个人信息，如果信息先前不存在数据库中，则可以注册。

### 1.1.3 系统性能需求

#### 1. 系统可靠性要求

系统应具有较高的稳定性。在用户使用时需要保证数据的正确性，数据的修改也需要实时进行，系统也需要能在何时何地都能运行起来。

#### 2. 系统响应时间需求

系统具有快速响应的特性，用户打开界面和提交事务的平均响应时间很短。用户进行实时查询业务操作的数据处理时间在 300ms 以内。

#### 3. 系统易用性要求

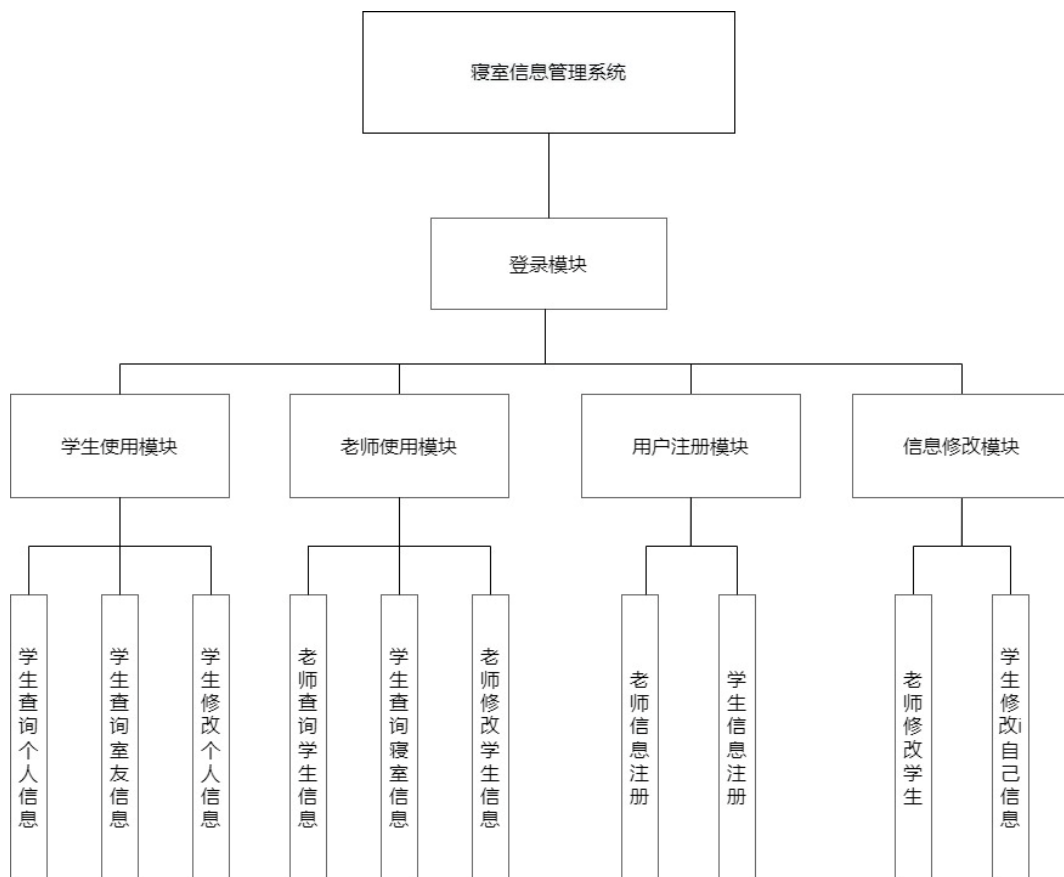
系统用户操作界面应操作简洁、易用。在后期维护的过程中，可以添加其他功能需求，更加方便用户使用。

### 1.1.4 数据分析

数据项名称	别名	类型	长度	取值	含义
工号	TNO	固定字符类型	10	9 位数字	教师的工号/账号
教师姓名	TNAME	可变字符类型	10	任意字符	教师的姓名
教师性别	TSEX	可变字符类型	5	男、女	教师的性别
教师密码	PASSWD	可变字符类型	20	任意字符	教师的密码
教师电话	PHONENUM	固定字符类型	15	11 位数字	教师的电话号码
教师备注	REMARKS	可变字符类型	50	任意字符	教师的备注
学号	SNO	固定字符类型	10	9 位数字	学生的学号/账号
学生姓名	SNAME	可变字符类型	10	任意字符	学生的姓名
学生性别	SSEX	可变字符类型	5	男、女	学生的性别
学生密码	PASSWD	可变字符类型	20	任意字符	学生的密码
学生电话	PHONENUM	固定字符类型	15	11 位数字	学生的电话号码
学生备注	REMARKS	可变字符类型	50	任意字符	学生的备注
楼栋号	FLOOR_NUM	可变字符类型	5	C1 - D2	学生居住的楼号
寝室号	DORMI_NUM	固定字符类型	5	3 位数字	学生所在的寝室号
床位	BED_NUM	固定字符类型	3	1、2、3、4	学生所在的床号
住宿费	MONEY	固定字符类型	10	数字	学生每年的住宿费
寝室居住备注	REMARKS	可变字符类型	50	任意字符	当前床号是否有人住

## 1.2 功能模块图及分模块功能描述

### 1.2.1 系统的功能模块图



### 1.2.2 系统功能模块简介

**学生使用模块：**学生使用模块是用来给学生用户查看修改数据的，包括对学生用户的学号、姓名、性别、联系方式、寝室信息（楼栋号，寝室号，床位号）、室友的查询和修改。

**老师使用模块：**老师使用模块是用来供老师用户查看学生信息、修改信息、删除学生信息模块的，包括查看学生学号、姓名、联系方式、寝室信息（楼栋号，寝室号，床位号）、每个寝室的成员部分信息。

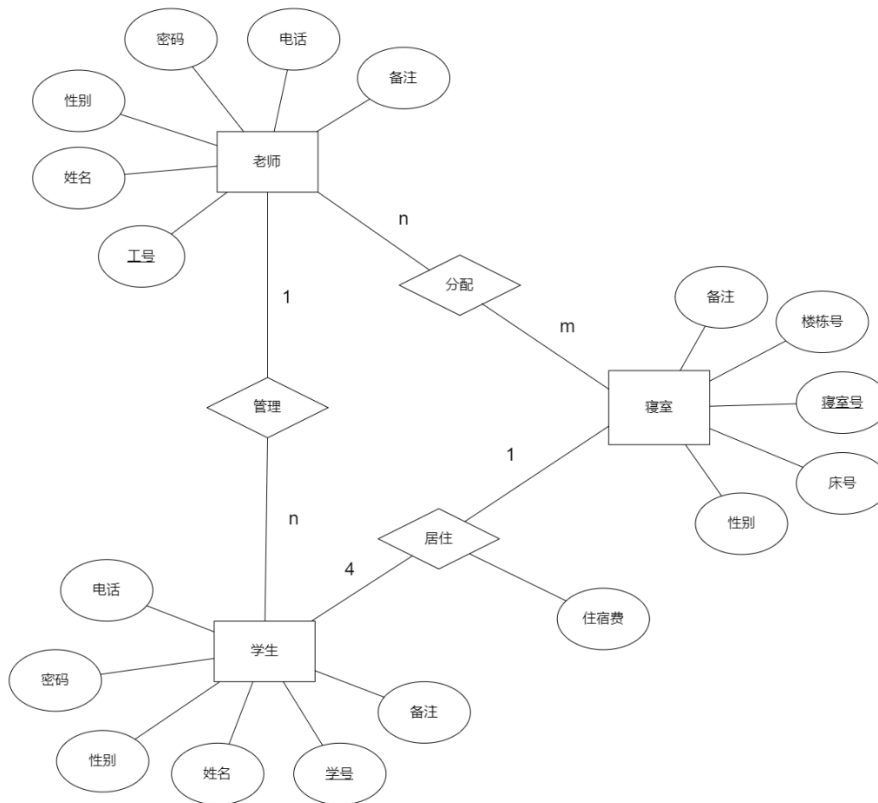
**用户注册模块：**用户注册模块是用来供新用户注册到系统的模块，包括老师用户和学生用户信息的注册，填写正确信息，且不与数据库数据冲突即可注册。

**信息修改模块：**信息修改模块是用来供用户修改信息的，包括查询信息，修改信息，

通过数据库触发器删除学生信息及其他信息。

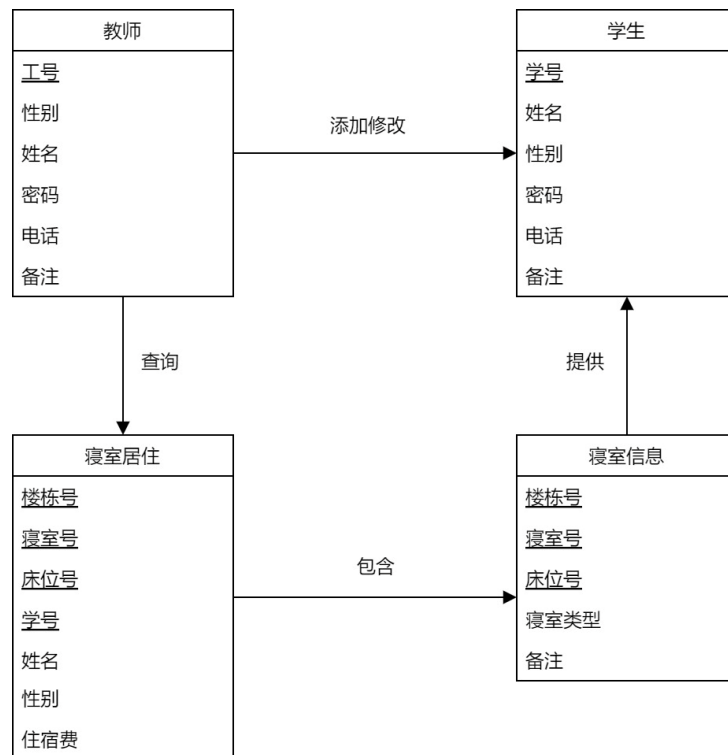
## 2 数据库设计

### 2.1 数据库概念设计（E-R 图）



### 2.1 数据库逻辑设计（关系模型）





### 初始设计:

老师 (工号, 老师姓名, 老师性别, 老师密码, 老师电话, 备注, 学生姓名, 学生性别, 学生电话)

学生 (学号, 学生姓名, 学生性别, 学生密码, 学生电话, 备注, 老师姓名, 老师电话)

寝室居住 (楼栋号, 寝室号, 床位号, 寝室类型, 备注, 老师姓名, 老师电话)

寝室信息 (楼栋号, 寝室号, 床位号, 学号, 姓名, 性别, 住宿费, 电话, 备注)

### 经过规范化后:

老师 (工号, 姓名, 性别, 密码, 电话, 备注)

学生 (学号, 姓名, 性别, 密码, 电话, 备注)

寝室居住 (楼栋号, 寝室号, 床位号, 寝室类型, 备注)

寝室信息 (楼栋号, 寝室号, 床位号, 学号, 姓名, 性别, 住宿费)

## 3 系统实施

### 3.1 建立数据库及其相关对象

#### 1. 数据库表目录：

teacher_info	教师信息表
student_info	学生信息表
dormitory	寝室基本表
dormitory_info	寝室居住表

#### 2. 数据库结构表：

teacher_info: 教师信息表		
表说明：记录每一个老师的个人信息		
字段	类型	注释
TNO	CHAR(10)	主键，教师的工号/账号
TNAME	VARCHAR(10)	教师的姓名
TSEX	VARCHAR(5)	教师的性别
PASSWD	VARCHAR(20)	教师的密码
PHONENUM	CHAR(15)	教师的电话号码
REMARKS	VARCHAR(50)	教师的备注

student_info: 学生信息表		
表说明：记录每一位学生的个人信息		
字段	类型	注释
SNO	CHAR(10)	主键，学生的学号/账号
SNAME	VARCHAR(10)	学生的姓名
SSEX	VARCHAR(5)	学生的性别
PASSWD	VARCHAR(20)	学生的密码
PHONENUM	CHAR(15)	学生的电话号码
REMARKS	VARCHAR(50)	学生的备注

dormitory: 寝室信息表		
表说明：记录每一个寝室的信息		
字段	类型	注释
FLOOR_NUM	VARCHAR(5)	关联主键，楼栋号码
DORMI_NUM	VARCHAR(5)	关联主键，寝室号码
BED_NUM	CHAR(5)	关联主键，床位号

SEX	CHAR(3)	男或女寝室
REMARKS	VARCHAR(50)	当前床号是否有人住

dormitory_info: 寝室居住表		
表说明：记录居住在寝室的每个人		
字段	类型	注释
FLOOR_NUM	VARCHAR(5)	关联主键，学生居住的楼号
DORMI_NUM	CHAR(5)	关联主键，学生所在的寝室号
BED_NUM	CHAR(3)	关联主键，学生所在的床号
SNO	CHAR(15)	外键，居住学生的学号
SNAME	VARCHAR(10)	居住学生的姓名
SSEX	VARCHAR(5)	居住学生的性别
MONEY	CHAR(10)	学生住宿费

### 3. Sql 语句:

创建表:

```

/*=====*/
/* Table: " teacher_info " 老师信息 */
/*=====*/
CREATE TABLE IF NOT EXISTS teacher_info
(
    TNO      CHAR(10)      PRIMARY KEY,          -- 账号/工号
    TNAME    VARCHAR(10),  -- 姓名
    TSEX     VARCHAR(5)    DEFAULT '男' CHECK(TSEX IN ('男','女')), -- 性别
    PASSWD   VARCHAR(20)   NOT NULL,             -- 密码
    PHONENUM CHAR(15),     -- 电话
    REMARKS  VARCHAR(50)   DEFAULT '无'          -- 备注
);

/*=====*/
/* Index: " student_info " 学生信息 */
/*=====*/
CREATE TABLE IF NOT EXISTS student_info
(
    SNO      CHAR(15)      PRIMARY KEY,          -- 账号/学号
    SNAME    VARCHAR(10),  -- 姓名
    SSEX     VARCHAR(5)    DEFAULT '男' CHECK(SSEX IN ('男','女')), -- 性别
    PASSWD   VARCHAR(20)   NOT NULL,             -- 密码
    PHONENUM CHAR(15),     -- 电话
    REMARKS  VARCHAR(50)   DEFAULT '无'          -- 备注
);

```

注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

```
);

/*=====*/
/* Index: " dormitory " 寝室基本信息 */
/*=====*/
CREATE TABLE IF NOT EXISTS dormitory
(
    FLOOR_NUM    VARCHAR(5),                -- 楼栋号
    SEX          VARCHAR(5)    DEFAULT '男' CHECK(SEX IN ('男','女')), -- 寝室类型
    DORMI_NUM    CHAR(5),                -- 寝室号
    BED_NUM      CHAR(3)    CHECK(BED_NUM IN ('1','2','3','4')), -- 床位
    REMARKS      VARCHAR(50)    DEFAULT '无',                -- 备注
    PRIMARY KEY(FLOOR_NUM , DORMI_NUM , BED_NUM)
);

/*=====*/
/* Index: " dormitory_info " 寝室居住信息 */
/*=====*/
CREATE TABLE IF NOT EXISTS dormitory_info
(
    FLOOR_NUM    VARCHAR(5),                -- 楼栋号
    DORMI_NUM    CHAR(5),                -- 寝室号
    BED_NUM      CHAR(3)    CHECK(BED_NUM IN ('1','2','3','4')), -- 床位
    SNO          CHAR(15),                -- 学号
    SNAME        VARCHAR(10),            -- 姓名
    SSEX         VARCHAR(5)    DEFAULT '男' CHECK(SSEX IN ('男','女')), -- 性别
    MONEY        CHAR(10)    DEFAULT '1000',                -- 住宿费
    PRIMARY KEY(FLOOR_NUM , DORMI_NUM , BED_NUM , SNO),
    FOREIGN KEY(SNO) REFERENCES student_info(SNO)
);
```

### 创建触发器和存储过程:

```
/*-----*/
/*-----老师删除学生信息触发器-----*/
/*-----*/
CREATE TRIGGER tea_del_stu
BEFORE DELETE
ON student_info
FOR EACH ROW
BEGIN
    UPDATE dormitory a CROSS JOIN dormitory_info b SET REMARKS = '无'
        WHERE a.FLOOR_NUM = b.FLOOR_NUM AND a.DORMI_NUM = b.DORMI_NUM
        AND a.BED_NUM = b.BED_NUM AND b.SNO = old.SNO;
    DELETE FROM dormitory_info WHERE SNO = old.SNO;
```

注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

```
END;
/

/*-----*/
/*-----老师更改学生寝室触发器-----*/
/*-----*/
CREATE TRIGGER tea_set_stu
AFTER UPDATE
ON dormitory_info
FOR EACH ROW
BEGIN
    UPDATE dormitory a SET REMARKS = '无'
        WHERE a.FLOOR_NUM = old.FLOOR_NUM AND a.DORMI_NUM = old.DORMI_NUM
        AND a.BED_NUM = old.BED_NUM;
    UPDATE dormitory a SET REMARKS = '有'
        WHERE a.FLOOR_NUM = new.FLOOR_NUM AND a.DORMI_NUM = new.DORMI_NUM
        AND a.BED_NUM = new.BED_NUM;
END;
/

/*-----*/
/*-----老师添加学生信息的存储过程-----*/
/*-----*/
CREATE PROCEDURE tea_add_stupro
(
    IN ADD_SNO CHAR(15),
    IN ADD_SNAME VARCHAR(10),
    IN ADD_SSEX VARCHAR(5),
    IN ADD_PASSWD VARCHAR(20),
    IN ADD_PHONENUM CHAR(15),
    IN ADD_FLOOR_NUM VARCHAR(5),
    IN ADD_DORMI_NUM CHAR(5),
    IN ADD_BED_NUM CHAR(3)
)
BEGIN
    INSERT INTO student_info(SNO, SNAME, SSEX, PASSWD, PHONENUM)
        VALUES(ADD_SNO, ADD_SNAME, ADD_SSEX, ADD_PASSWD, ADD_PHONENUM);
    INSERT INTO dormitory_info(FLOOR_NUM, DORMI_NUM, BED_NUM, SNO, SNAME,
        SSEX) VALUES(ADD_FLOOR_NUM, ADD_DORMI_NUM, ADD_BED_NUM, ADD_SNO,
        ADD_SNAME, ADD_SSEX);
    UPDATE dormitory a CROSS JOIN dormitory_info b SET REMARKS = '有'
        WHERE a.FLOOR_NUM = b.FLOOR_NUM AND a.DORMI_NUM = b.DORMI_NUM AND
        a.BED_NUM = b.BED_NUM AND b.SNO = ADD_SNO;
END;
```

注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

/

## 3.2 数据库连接

使用 pymysql 库进行数据库连接

```
import pymysql
conn = pymysql.connect(
    host='localhost',
    port=3306,
    user='root',
    password="000000",
    db="dormitory-manager"
)
cur = conn.cursor()
sql = f"SELECT * FROM student_info"
cur.execute(sql)
```

## 3.3 主要模块实施

### 3.3.1 登录模块的开发

```
def retranslateUi(self, Dialog_login):
    _translate = QtCore.QCoreApplication.translate
    Dialog_login.setWindowTitle(_translate("Dialog_login", "登录界面"))
    self.label_subtitle_2.setText(_translate("Dialog_login", "登录界面"))
    self.label_title_main.setText(_translate("Dialog_login", "学生寝室管理系统"))
    self.radioButton_stu.setText(_translate("Dialog_login", "学生"))
    self.radioButton_tea.setText(_translate("Dialog_login", "老师"))
    self.pushButton_registered.setText(_translate("Dialog_login", "注册"))
    self.pushButton_login.setText(_translate("Dialog_login", "登录"))
    self.label_acc.setText(_translate("Dialog_login", "账号: "))
    self.label_pd.setText(_translate("Dialog_login", "密码: "))
```

结果如下：



### 3.3.2 用户搜索模块的开发

```
/*===== 学生搜索界面 =====*/
def retranslateUi(self, Stu_search):
    _translate = QtCore.QCoreApplication.translate
    Stu_search.setWindowTitle(_translate("Stu_search", "学生搜索界面"))
    item = self.tableWidget.verticalHeaderItem(0)
    item.setText(_translate("Stu_search", "1"))
    item = self.tableWidget.verticalHeaderItem(1)
    item.setText(_translate("Stu_search", "2"))
    item = self.tableWidget.verticalHeaderItem(2)
    item.setText(_translate("Stu_search", "3"))
    item = self.tableWidget.verticalHeaderItem(3)
    item.setText(_translate("Stu_search", "4"))
    item = self.tableWidget.horizontalHeaderItem(0)
    item.setText(_translate("Stu_search", "1"))
    item = self.tableWidget.horizontalHeaderItem(1)
    item.setText(_translate("Stu_search", "2"))
    item = self.tableWidget.horizontalHeaderItem(2)
    item.setText(_translate("Stu_search", "3"))
    item = self.tableWidget.horizontalHeaderItem(3)
    item.setText(_translate("Stu_search", "4"))
    self.label_title_main.setText(_translate("Stu_search", "学生寝室管理系统"))
    self.label_subtitle_2.setText(_translate("Stu_search", "学生查询界面"))
    self.label_role.setText(_translate("Stu_search", "当前身份: 学生"))
    self.label_title_3.setText(_translate("Stu_search", "学生信息查询"))
    self.pushButton_sea_xs.setText(_translate("Stu_search", "信息查询"))
    self.pushButton_sea_xs_2.setText(_translate("Stu_search", "信息修改"))
    self.pushButton_sea_qs.setText(_translate("Stu_search", "本寝室查询"))
    self.pushButton_back.setText(_translate("Stu_search", "返回"))
```

注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

```
/*===== 老师搜索界面 =====*/
def retranslateUi(self, Tea_search):
    _translate = QtCore.QCoreApplication.translate
    Tea_search.setWindowTitle(_translate("Tea_search", "教师搜索界面"))
    item = self.tableWidget.horizontalHeaderItem(0)
    item.setText(_translate("Tea_search", "学号"))
    item = self.tableWidget.horizontalHeaderItem(1)
    item.setText(_translate("Tea_search", "姓名"))
    item = self.tableWidget.horizontalHeaderItem(2)
    item.setText(_translate("Tea_search", "性别"))
    item = self.tableWidget.horizontalHeaderItem(3)
    item.setText(_translate("Tea_search", "电话"))
    item = self.tableWidget.horizontalHeaderItem(4)
    item.setText(_translate("Tea_search", "楼栋号"))
    item = self.tableWidget.horizontalHeaderItem(5)
    item.setText(_translate("Tea_search", "寝室号"))
    item = self.tableWidget.horizontalHeaderItem(6)
    item.setText(_translate("Tea_search", "床号"))
    item = self.tableWidget.horizontalHeaderItem(7)
    item.setText(_translate("Tea_search", "住宿费"))
    item = self.tableWidget.horizontalHeaderItem(8)
    item.setText(_translate("Tea_search", "备注"))
    self.label_title_main.setText(_translate("Tea_search", "学生寝室管理系统"))
    self.label_subtitle_2.setText(_translate("Tea_search", "老师查询界面"))
    self.label_role.setText(_translate("Tea_search", "当前身份: 老师"))
    self.label_title_3.setText(_translate("Tea_search", "学生信息查询"))
    self.label_name.setText(_translate("Tea_search", "姓名: "))
    self.label_acc.setText(_translate("Tea_search", "学号: "))
    self.pushButton_sea_xs.setText(_translate("Tea_search", "查询"))
    self.pushButton_sea_xs_2.setText(_translate("Tea_search", "修改"))
    self.label_title_4.setText(_translate("Tea_search", "寝室查询"))
    self.label_floor.setText(_translate("Tea_search", "楼栋号: "))
    self.comboBox_floor.setItemText(0, _translate("Tea_search", "C1"))
    self.comboBox_floor.setItemText(1, _translate("Tea_search", "C2"))
    self.comboBox_floor.setItemText(2, _translate("Tea_search", "D1"))
    self.comboBox_floor.setItemText(3, _translate("Tea_search", "D2"))
    self.label_qsh.setText(_translate("Tea_search", "寝室号: "))
    self.pushButton_sea_qs.setText(_translate("Tea_search", "查询"))
    self.pushButton_back.setText(_translate("Tea_search", "返回"))
```

结果如下:

注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>





### 3.3.3 用户注册模块的开发

```
def retranslateUi(self, Register):
    _translate = QtCore.QCoreApplication.translate
    Register.setWindowTitle(_translate("Register", "注册界面"))
    self.pushButton_back.setText(_translate("Register", "返回"))
    self.pushButton_next.setText(_translate("Register", "确认"))
    self.label_acc.setText(_translate("Register", "工号/学号: "))
    self.label_name.setText(_translate("Register", "姓名: "))
    self.label_sex.setText(_translate("Register", "性别: "))
    self.radioButton_man.setText(_translate("Register", "男"))
    self.radioButton_won.setText(_translate("Register", "女"))
    self.label_phone.setText(_translate("Register", "电话: "))
    self.label_floor.setText(_translate("Register", "楼栋号: "))
    self.comboBox_floor.setItemText(0, _translate("Register", "C1"))
    self.comboBox_floor.setItemText(1, _translate("Register", "C2"))
    self.comboBox_floor.setItemText(2, _translate("Register", "D1"))
    self.comboBox_floor.setItemText(3, _translate("Register", "D2"))
    self.label_qsh.setText(_translate("Register", "寝室号: "))
    self.label_bed.setText(_translate("Register", "床位: "))
    self.comboBox_bed.setItemText(0, _translate("Register", "1"))
    self.comboBox_bed.setItemText(1, _translate("Register", "2"))
    self.comboBox_bed.setItemText(2, _translate("Register", "3"))
    self.comboBox_bed.setItemText(3, _translate("Register", "4"))
    self.label_role.setText(_translate("Register", "身份: "))
    self.radioButton_stu.setText(_translate("Register", "学生"))
    self.radioButton_tea.setText(_translate("Register", "老师"))
    self.label_pd.setText(_translate("Register", "密码: "))
    self.label_pd_b.setText(_translate("Register", "再次输入: "))
    self.label_bz.setText(_translate("Register", "备注: "))
```

注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

```
self.label_subtitle.setText(_translate("Register", "注册界面"))
self.label_title_main.setText(_translate("Register", "学生寝室管理系统"))
```

结果如下：



3.4 系统测试

3.4.1 软件测试的对象

以用户的相关操作，设计测试方式如下：

- 1. 登录界面，用户可以选择两种不同的身份进行登录，不存在的学生用户登录时会登录成功；使用不存在的用户登录不成功。如果密码不正确，会提醒用户重新输入。



- 1. 登录学生查询页面，点击“信息查询”，即可查到自己的个人信息，点击“本寝室查

询”，可查找室友信息。



2. 修改学生信息，根据输入的学号和姓名修改学生的寝室信息。如果床位已经有人使用会修改不成功，并提醒用户修改其他床位。



3. 老师查询界面，可以根据学号姓名查询所有学生信息，也可以根据寝室信息去查找每个寝室的居住使用信息。

教师搜索界面

### 学生寝室管理系统

老师查询界面

	学号	姓名	性别	电话	楼栋号	寝室号
1	187045...	红好成	男	138907...	C1	101
2						
3						
4						
5						
6						

当前身份: 老师

学生信息查询

学号: 187045019

姓名: 红好成

寝室查询

楼栋号: C1

寝室号:

教师搜索界面

### 学生寝室管理系统

老师查询界面

	楼栋号	寝室号	床号	学号	姓名	性别
1	C1	101	1	187045...	红好成	男
2	C1	101	2	187045...	衡马坚	男
3	C1	101	3	187045...	单于...	男
4	C1	101	4	未入住	未入住	未入住
5						
6						

当前身份: 老师

学生信息查询

学号: 187045019

姓名: 红好成

寝室查询

楼栋号: C1

寝室号: 101

4. 学生注册界面, 如果输入账号与数据库有相同, 会提醒用户, 如果前后两次密码输入不一样, 会警告用户重新输入。如果床位有人使用, 当注册时再次使用会提示用户有人使用此床位。将必要信息填写完整后, 用户就可以注册成功了。

注册界面

提醒

学号账号存在!

OK

工号/学号: 187045019

姓名: 红好成

性别: ☒ 男 ☐ 女

电话: 1389074952

楼栋号: C1

寝室号: 101

床位: 1

身份: ☒ 学生 ☐ 老师

密码: ●●●

再次输入: ●●●

备注:

注册界面

提醒

床位已有人使用!

OK

工号/学号: 111111111

姓名: 红好成

性别: ☒ 男 ☐ 女

电话: 1389074952

楼栋号: C1

寝室号: 101

床位: 1

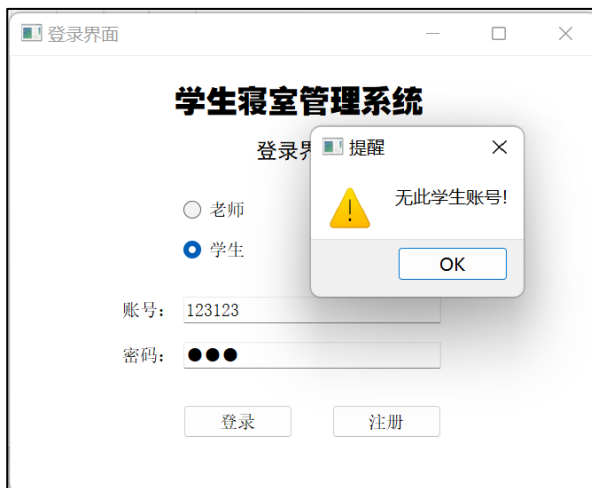
身份: ☒ 学生 ☐ 老师

密码: ●●●

再次输入: ●●●

备注:





### 3.4.2 软件测试的结果

测试程序运行顺序和结果如下：



注意不要雷同

banban

<https://github.com/dream4789/Computer-learning-resources.git>

## 4 系统说明

### 4.1 开发环境

本系统的是在 Python 平台下使用 QT 开发的，系统的后台数据库为 Mysql 数据库；因此在使用本系统前，应先安装 QT 框架和 Mysql 数据库，否则该系统无法运行。

Python 版本：3.9.10

PyQt5 版本：5.15.2

Mysql 版本：8.0.28

### 4.2 系统安装、配置与发布应用程序的步骤

系统的原文件直接复制到机器上后，在数据库管理系统上建立一个名为 dormitory-manager 的数据库后，创建所需要的四个基本表，之后在 python 里运行 dormitory\_qt\_master.py，系统便可以运行。

## 总 结

在整个数据库课程设计的过程中，我用了十天时间自学了 python 与数据库的连接方法，以及如何使用 QT 绘制窗体，如何设计窗口运行程序，使自己对 python 和数据库之间的关系有了进一步的了解，提升了自己的数据库设计水平。

在本次寝室管理课程设计中，通过设计了一个简单的窗口界面系统，解决了 Python 与数据库的连接问题，编写窗口应用程序的问题。但其中仍有未解决的问题，例如对整体框架的完善和窗口之间的关联性。日后会对系统的框架进行进一步的完善，以提高系统的可扩展性和可维护性。通过本次课程设计，也体会到了软件开发过程的艰辛以及辛苦，感受到了程序与数据库之间的关系，对编程的兴趣有了进一步的提高。

当然，在设计的过程中也存在很多问题，这次课程设计是对我的学习态度的一次检验。对于这次我的第一大心得体会就是作为一名计算机技术人员，要求具备的首要素质绝对应当是严谨，需要有足够的综合应用的本事和自学能力，感激教师的教诲以及同学的帮忙，是他们让我有了一个更好的认识。



## 附录：部分源代码

```

#####
#####-----登录-----#####
#####

class Main_login(QLineEdit, login.Ui_Dialog_login):
    line_acc = 0
    line_pd = ''
    RButton_role = ''

    def __init__(self):
        super().__init__()
        self.setupUi(self)

        IntValidator = QIntValidator(1, 999999999, self)
        self.lineEdit_acc.setValidator(IntValidator)
        self.lineEdit_acc.textChanged.connect(self.acc_input)
        self.lineEdit_pd.setEchoMode(QLineEdit.Password)
        self.lineEdit_pd.textChanged.connect(self.pd_input)
        self.radioButton_stu.toggled.connect(lambda: self.role_input(self.radioButton_stu))
        self.radioButton_tea.toggled.connect(lambda: self.role_input(self.radioButton_tea))
        self.pushButton_registered.clicked.connect(self.register_info)
        self.pushButton_login.clicked.connect(self.login_info)

    def acc_input(self):
        Main_login.line_acc = self.lineEdit_acc.text()

    def pd_input(self):
        Main_login.line_pd = self.lineEdit_pd.text()

    def register_info(self):
        ui_reg.show()

    def role_input(self, rolebtn):
        if rolebtn.text() == "学生":
            if rolebtn.isChecked():
                Main_login.RButton_role = rolebtn.text()

        if rolebtn.text() == "老师":
            if rolebtn.isChecked():
                Main_login.RButton_role = rolebtn.text()

    def login_info(self):
        conn = pymysql.connect(
            host='localhost',
            port=3306,

```

```
        user='root',
        password="000000",
        db="dormitory-manager"
    )
    cur = conn.cursor()
    if Main_login.RButton_role == '':
        QMessageBox.warning(self, "提醒", "请输入角色!", QMessageBox.Ok)
    else:
        if Main_login.line_acc == 0 or Main_login.line_pd == '':
            QMessageBox.warning(self, "提醒", "请输入账号或密码!", QMessageBox.Ok)
        else:
            if Main_login.RButton_role == '学生':
                sql = f"SELECT SNO FROM student_info"
                cur.execute(sql)
                data = cur.fetchall()
                data_list = list(chain.from_iterable(data))

                if Main_login.line_acc not in data_list:
                    QMessageBox.warning(self, "提醒", "无此学生账号!", QMessageBox.Ok)
                else:
                    sql = f"SELECT SNAME FROM student_info " \
                        f"where SNO = '{Main_login.line_acc}';"
                    cur.execute(sql)
                    data_name = cur.fetchall()[0][0]

                    sql = f"SELECT PASSWD FROM student_info " \
                        f"where SNO = '{Main_login.line_acc}';"
                    cur.execute(sql)
                    data = cur.fetchall()
                    if data[0][0] == Main_login.line_pd:
                        print('学生登陆成功!')
                        self.stu_login()
                        global line_sno
                        global line_name
                        line_sno = Main_login.line_acc
                        line_name = data_name
                    else:
                        QMessageBox.critical(self, "错误", "密码错误!")
            else:
                sql = f"SELECT TNO FROM teacher_info"
                cur.execute(sql)
                data = cur.fetchall()
                data_list = list(chain.from_iterable(data))

                if Main_login.line_acc not in data_list:
```

```

        QMessageBox.warning(self, "提醒", "无此老师账号!", QMessageBox.Ok)
    else:
        sql = f"SELECT PASSWD FROM teacher_info " \
              f"where TNO = '{Main_login.line_acc}'"
        cur.execute(sql)
        data = cur.fetchall()
        if data[0][0] == Main_login.line_pd:
            print('老师登录成功!')
            self.tea_login()
        else:
            QMessageBox.critical(self, "错误", "密码错误!")

    cur.close()
    conn.close()

def stu_login(self):
    ui_stu.show()

def tea_login(self):
    ui_tea.show()

###-----###
###-----老师搜索-----###
###-----###

class Teacher_Search(QWidget, tea_search.Ui_Tea_search):
    line_sno = 0
    line_name = ''
    line_floor = 'C1'
    line_dormi = 0

    def __init__(self):
        super().__init__() # 继承父类里的函数
        self.setupUi(self) # 初始化页面
        self.pushButton_sea_xs.clicked.connect(self.find_stu)
        self.pushButton_sea_xs_2.clicked.connect(self.alter_stu)
        self.pushButton_sea_qs.clicked.connect(self.find_dormi)
        self.pushButton_back.clicked.connect(self.back_login)
        self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)
        self.lineEdit_qsh.textChanged.connect(self.qsh_input)
        self.comboBox_floor.currentIndexChanged.connect(self.floor_combox)
        IntValidator = QIntValidator(1, 999999999, self)
        self.lineEdit_sno.setValidator(IntValidator) # 限制输入大小
        self.lineEdit_sno.textChanged.connect(self.sno_input)
        self.lineEdit_name.textChanged.connect(self.name_input) # 输入结束时触发

def sno_input(self):

```

```
Teacher_Search.line_sno = self.lineEdit_sno.text() # 等价于返回值（文本框输入内容）

def name_input(self):
    Teacher_Search.line_name = self.lineEdit_name.text() # 文本框输入内容

def qsh_input(self):
    Teacher_Search.line_dormi = self.lineEdit_qsh.text()

def floor_combox(self):
    Teacher_Search.line_floor = self.comboBox_floor.currentText()

def find_stu(self):
    self.tableWidget.clearContents()
    self.tableWidget.setHorizontalHeaderLabels(["学号", "姓名", "性别", "电话", "楼栋号", "寝室号", "床号", "住宿费", "备注"])
    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password="000000",
        db="dormitory-manager")
    cur = conn.cursor()

    if Teacher_Search.line_sno == 0:
        QMessageBox.warning(self, "警告", "账号为空!", QMessageBox.Ok)
    else:
        sql = f"SELECT SNO FROM student_info;"
        cur.execute(sql)
        data_list = cur.fetchall()
        data_list_sno = list(chain.from_iterable(data_list))

        if Teacher_Search.line_sno in data_list_sno:
            if Teacher_Search.line_sno == 0 or Teacher_Search.line_name == '':
                QMessageBox.warning(self, "提醒", "信息请输入完整!", QMessageBox.Ok)
            else:
                sql = f"SELECT SNAME FROM student_info WHERE " \
                    f"SNO = '{Teacher_Search.line_sno}'"
                cur.execute(sql)
                data = cur.fetchall()
                if data[0][0] != Teacher_Search.line_name:
                    QMessageBox.warning(self, "提醒", "学号或姓名错误!", QMessageBox.Ok)
                else:
                    sql = f"SELECT a.SNO, a.SNAME, a.SSEX, PHONENUM, " \
                        f"FLOOR_NUM, DORMI_NUM, BED_NUM, MONEY, REMARKS " \
                        f"FROM student_info a CROSS JOIN dormitory_info b " \
```

```

        f"WHERE a.SNO = b.SNO AND a.SNO = '{Teacher_Search.line_sno}';"
    cur.execute(sql)
    data = cur.fetchall()
    x = 0
    for i in data:
        y = 0
        for j in i:
            self.tableWidget.setItem(x, y, QTableWidgetItem(str(data[x][y])))
            y = y + 1
        x = x + 1
    else:
        QMessageBox.warning(self, "提醒", "账号不存在!", QMessageBox.Ok)

    cur.close()
    conn.close()

def alter_stu(self):
    ui_add.show()

def find_dormi(self):
    self.tableWidget.clearContents()
    self.tableWidget.setHorizontalHeaderLabels(["楼栋号", "寝室号", "床号", "学号", "姓名", "性别", "", "", "", ""])
    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password="000000",
        db="dormitory-manager")
    cur = conn.cursor()
    sql = f"SELECT DISTINCT DORMI_NUM FROM dormitory;"
    cur.execute(sql)
    data = cur.fetchall()
    data_list = list(chain.from_iterable(data))
    if Teacher_Search.line_dormi not in data_list:
        QMessageBox.warning(self, "提醒", "信息不正确!", QMessageBox.Ok)
    elif Teacher_Search.line_dormi == 0:
        QMessageBox.warning(self, "提醒", "信息请输入完整!", QMessageBox.Ok)
    else:
        sql = f"SELECT a.FLOOR_NUM, a.DORMI_NUM, a.BED_NUM, IFNULL(b.SNO, '未入住') SNO, " \
            f"IFNULL(b.SNAME, '未入住') SNAME, IFNULL(b.SSEX, '未入住') SSEX " \
            f"FROM dormitory a LEFT JOIN dormitory_info b ON a.FLOOR_NUM = b.FLOOR_NUM " \
            f"AND a.DORMI_NUM = b.DORMI_NUM AND a.BED_NUM = b.BED_NUM " \
            f"WHERE a.FLOOR_NUM = '{Teacher_Search.line_floor}' AND a.DORMI_NUM = " \
            f'{Teacher_Search.line_dormi}';"

```

```

        cur.execute(sql)
        data = cur.fetchall()
        x = 0
        for i in data:
            y = 0
            for j in i:
                self.tableWidget.setItem(x, y, QTableWidgetItem(str(data[x][y])))
                y = y + 1
            x = x + 1
        cur.close()
        conn.close()

    def back_login(self):
        ui_tea.close()

###-----###
###-----学生搜索-----###
###-----###

class Student_Search(QWidget, stu_search.Ui_Stu_search):
    line_floor = ''
    line_dormi = 0
    def __init__(self):
        super().__init__() # 继承父类里的函数
        self.setupUi(self) # 初始化页面
        self.pushButton_sea_xs.clicked.connect(self.find_stu)
        self.pushButton_sea_xs_2.clicked.connect(self.alter_stu)
        self.pushButton_sea_qs.clicked.connect(self.find_dormi)
        self.pushButton_back.clicked.connect(self.back_login)
        self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)

    def find_stu(self):
        self.tableWidget.clearContents()
        self.tableWidget.setHorizontalHeaderLabels(["学号", "姓名", "性别", "电话", "楼栋号", "寝室号", "床号", "住宿费", "备注"])
        conn = pymysql.connect(
            host='localhost',
            port=3306,
            user='root',
            password="000000",
            db="dormitory-manager")
        cur = conn.cursor()
        sql = f"SELECT * FROM stu_view WHERE SNO = '{line_sno}';"
        cur.execute(sql)
        data = cur.fetchall()

```

```
x = 0
for i in data:
    y = 0
    for j in i:
        self.tableWidget.setItem(x, y, QTableWidgetItem(str(data[x][y])))
        y = y + 1
    x = x + 1
cur.close()
conn.close()

def alter_stu(self):
    ui_add.show()

def find_dormi(self):
    self.tableWidget.clearContents()
    self.tableWidget.setHorizontalHeaderLabels(["楼栋号", "寝室号", "床号", "学号", "姓名", "性别", "", "", "", ""])
    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password="000000",
        db="dormitory-manager")
    cur = conn.cursor()
    sql = f"SELECT FLOOR_NUM, DORMI_NUM FROM dormitory_info WHERE SNO = '{line_sno}';"
    cur.execute(sql)
    data = cur.fetchall()
    Student_Search.line_floor = data[0][0]
    Student_Search.line_dormi = data[0][1]
    sql = f"SELECT a.FLOOR_NUM, a.DORMI_NUM, a.BED_NUM, IFNULL(b.SNO, '未入住') SNO, " \
        f"IFNULL(b.SNAME, '未入住') SNAME, IFNULL(b.SSEX, '未入住') SSEX " \
        f"FROM dormitory a LEFT JOIN dormitory_info b ON a.FLOOR_NUM = b.FLOOR_NUM " \
        f"AND a.DORMI_NUM = b.DORMI_NUM AND a.BED_NUM = b.BED_NUM " \
        f"WHERE a.FLOOR_NUM = '{Student_Search.line_floor}' " \
        f"AND a.DORMI_NUM = '{Student_Search.line_dormi}';"
    cur.execute(sql)
    data = cur.fetchall()
    x = 0
    for i in data:
        y = 0
        for j in i:
            self.tableWidget.setItem(x, y, QTableWidgetItem(str(data[x][y])))
            y = y + 1
        x = x + 1
```

```

        cur.close()
        conn.close()

    def back_login(self):
        ui_stu.close()

###-----###
###-----注册-----###
###-----###

class Register(QWidget, register.Ui_Register):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.line_acc = 0
        self.line_name = ''
        self.Rbutton_sex = ''
        self.Rbutton_role = ''
        self.line_phone = 0
        self.line_pd1 = 0
        self.line_pd2 = 0
        self.line_mk = ''

        self.line_floor = 'C1'
        self.line_dormi = 0
        self.line_bed = '1'

        self.lineEdit_acc.textChanged.connect(self.acc_input)
        self.lineEdit_name.textChanged.connect(self.name_input)
        self.lineEdit_phone.textChanged.connect(self.phone_input)
        self.lineEdit_pd_1.textChanged.connect(self.pd1_input)
        self.lineEdit_pd_2.editingFinished.connect(self.pd2_input)
        self.lineEdit_qsh.textChanged.connect(self.qsh_input)
        self.radioButton_man.toggled.connect(lambda: self.sex_input(self.radioButton_man))
        self.radioButton_won.toggled.connect(lambda: self.sex_input(self.radioButton_won))
        self.radioButton_stu.toggled.connect(lambda: self.role_input(self.radioButton_stu))
        self.radioButton_tea.toggled.connect(lambda: self.role_input(self.radioButton_tea))
        self.comboBox_floor.currentIndexChanged.connect(self.floor_combox)
        self.comboBox_bed.currentIndexChanged.connect(self.bed_combox)
        self.pushButton_next.clicked.connect(self.next_click)
        self.pushButton_back.clicked.connect(self.back_click)

    def acc_input(self):
        self.line_acc = self.lineEdit_acc.text()

    def name_input(self):

```



```
self.line_name = self.lineEdit_name.text()

def sex_input(self, sexbtn):
    if sexbtn.text() == "男":
        if sexbtn.isChecked():
            self.Rbutton_sex = sexbtn.text()

    if sexbtn.text() == "女":
        if sexbtn.isChecked():
            self.Rbutton_sex = sexbtn.text()

def role_input(self, rolebtn):
    if rolebtn.text() == "学生":
        if rolebtn.isChecked():
            self.lineEdit_qsh.setText('')
            self.lineEdit_qsh.setReadOnly(False)
            self.Rbutton_role = rolebtn.text()

    if rolebtn.text() == "老师":
        if rolebtn.isChecked():
            str_hint = '无需填写'
            self.lineEdit_qsh.setText(str_hint)
            self.lineEdit_qsh.setReadOnly(True)
            self.Rbutton_role = rolebtn.text()

def phone_input(self):
    self.line_phone = self.lineEdit_phone.text()

def pd1_input(self):
    self.line_pd1 = self.lineEdit_pd_1.text()

def pd2_input(self):
    self.line_pd2 = self.lineEdit_pd_2.text()
    if self.line_pd1 != self.line_pd2:
        QMessageBox.critical(self, "错误", "密码不一致!")

def qsh_input(self):
    self.line_dormi = self.lineEdit_qsh.text()

def floor_combox(self):
    self.line_floor = self.comboBox_floor.currentText()

def bed_combox(self):
    self.line_bed = self.comboBox_bed.currentText()
```

```

def next_click(self):
    self.line_mk = self.textEdit_bz.toPlainText()
    if self.Rbutton_role == '':
        QMessageBox.warning(self, "提醒", "请选择角色!", QMessageBox.Ok)
    else:
        if self.Rbutton_role == '学生':
            if self.line_acc == '' or self.line_name == '' or self.line_pd2 == 0 or \
                self.Rbutton_sex == '' or self.line_phone == 0 or self.line_pd1 == 0 \
                or self.line_dormi == 0:
                QMessageBox.warning(self, "提醒", "学生信息不全!", QMessageBox.Ok)
            else:
                # 存入学生信息
                conn = pymysql.connect(
                    host='localhost',
                    port=3306,
                    user='root',
                    password="000000",
                    db="dormitory-manager")
                cur = conn.cursor()
                sql = f"SELECT SNO FROM student_info"
                cur.execute(sql)
                data = cur.fetchall()
                data_list = list(chain.from_iterable(data))
                if self.line_acc in data_list:
                    QMessageBox.warning(self, "提醒", "学号账号存在!", QMessageBox.Ok)
                else:
                    sql = f"SELECT FLOOR_NUM, DORMI_NUM, BED_NUM FROM dormitory WHERE
REMARKS = '无';"
                    cur.execute(sql)
                    data_list = cur.fetchall()
                    a = [self.line_floor, self.line_dormi, self.line_bed]
                    a = tuple(a)

                    if a in data_list:
                        sql = f"CALL
tea_add_stupro('{self.line_acc}','{self.line_name}','{self.Rbutton_sex}'," \
                    f"'{self.line_pd2}','{self.line_phone}','{self.line_floor}'"\
                    f"'{self.line_dormi}','{self.line_bed}','{self.line_mk}');"
                        cur.execute(sql)
                        conn.commit()
                        print('学生注册成功', sql)
                        QMessageBox.information(self, "注册", "学生注册成功!",
QMessageBox.Yes)
                    else:
                        QMessageBox.warning(self, "提醒", "床位已有人使用!", QMessageBox.Ok)

```

```

        cur.close()
        conn.close()

    else:
        if self.line_acc == '' or self.line_name == '' or self.line_phone == 0 or \
            self.Rbutton_sex == '' or self.line_pd1 == 0 or self.line_pd2 == 0:
            QMessageBox.warning(self, "提醒", "老师信息不全!", QMessageBox.Ok)
        else:
            # 存入老师信息
            conn = pymysql.connect(
                host='localhost',
                port=3306,
                user='root',
                password="000000",
                db="dormitory-manager")

            cur = conn.cursor()
            sql = f"SELECT TNO FROM teacher_info"
            cur.execute(sql)
            data = cur.fetchall()
            data_list = list(chain.from_iterable(data))
            if self.line_acc in data_list:
                QMessageBox.warning(self, "提醒", "账号存在!", QMessageBox.Ok)
            else:
                QMessageBox.information(self, "注册", "老师注册成功!", QMessageBox.Yes)
                sql = f"INSERT INTO teacher_info(TNO, TNAME, TSEX, " \
                    f"PASSWD, PHONENUM, REMARKS) VALUE " \
                    f"('{self.line_acc}','{self.line_name}','{self.Rbutton_sex}'," \
                    f"'{self.line_pd2}','{self.line_phone}', '{self.line_mk}');"
                cur.execute(sql)

            cur.close()
            conn.close()

    def back_click(self):
        ui_reg.close()

###-----###
###-----学生修改-----###
###-----###

class Student_change(QWidget, add_info.Ui_change_info):
    def __init__(self):
        super().__init__() # 继承父类里的函数
        self.setupUi(self) # 初始化页面

        self.line_sno = 0
        self.line_name = ''
        self.line_floor = 'C1'

```

```

self.line_dormi = 0
self.line_bed = '1'

self.lineEdit_sno.textChanged.connect(self.sno_input)
self.lineEdit_name.textChanged.connect(self.name_input)
self.pushButton_find.clicked.connect(self.find_stu)
self.pushButton_save.clicked.connect(self.save_stu)
self.pushButton_back.clicked.connect(self.back_add)
self.comboBox_floor.currentIndexChanged.connect(self.floor_combox)
self.comboBox_bed.currentIndexChanged.connect(self.bed_combox)
self.lineEdit_qsh.textChanged.connect(self.qsh_input)
self.tableWidget.setEditTriggers(QAbstractItemView.NoEditTriggers)

def find_stu(self):
    self.tableWidget.clearContents()

    self.tableWidget.setHorizontalHeaderLabels(["学号", "姓名", "性别", "电话", "楼栋号", "寝室号", "床号", "住宿费", "备注"])

    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password="000000",
        db="dormitory-manager")
    cur = conn.cursor()

    sql = f"SELECT SNO FROM student_info;"
    cur.execute(sql)
    data_list = cur.fetchall()
    data_list_sno = list(chain.from_iterable(data_list))
    if self.line_sno in data_list_sno:
        if self.line_sno != 0:
            sql = f"SELECT SNAME FROM student_info WHERE SNO = '{self.line_sno}'"
            cur.execute(sql)
            data_name = cur.fetchall()
            if self.line_sno == 0 or self.line_name == '':
                QMessageBox.warning(self, "提醒", "学生信息不全!", QMessageBox.Ok)
            elif self.line_name != data_name[0][0]:
                QMessageBox.warning(self, "提醒", "账号或姓名不正确!", QMessageBox.Ok)
            else:
                sql = f"SELECT a.SNO, a.SNAME, a.SSEX, PHONENUM, FLOOR_NUM, " \
                    f"DORMI_NUM, BED_NUM, MONEY, REMARKS " \
                    f"FROM student_info a CROSS JOIN dormitory_info b " \
                    f"WHERE a.SNO = b.SNO AND a.SNO = '{self.line_sno}';"
                cur.execute(sql)
                data = cur.fetchall()
                x = 0

```

```
        for i in data:
            y = 0
            for j in i:
                self.tableWidget.setItem(x, y, QTableWidgetItem(str(data[x][y])))
                y = y + 1
            x = x + 1
        else:
            QMessageBox.warning(self, "提醒", "请输入账号!", QMessageBox.Ok)
    else:
        QMessageBox.warning(self, "提醒", "账号不存在!", QMessageBox.Ok)
    cur.close()
    conn.close()

def qsh_input(self):
    self.line_dormi = self.lineEdit_qsh.text()

def floor_combox(self):
    self.line_floor = self.comboBox_floor.currentText()

def bed_combox(self):
    self.line_bed = self.comboBox_bed.currentText()

def sno_input(self):
    self.line_sno = self.lineEdit_sno.text()

def name_input(self):
    self.line_name = self.lineEdit_name.text()

def save_stu(self):
    conn = pymysql.connect(
        host='localhost',
        port=3306,
        user='root',
        password="000000",
        db="dormitory-manager")
    cur = conn.cursor()
    sql = f"SELECT SNO FROM student_info;"
    cur.execute(sql)
    data_list = cur.fetchall()
    data_list_sno = list(chain.from_iterable(data_list))
    if self.line_sno in data_list_sno:
        if self.line_sno != 0:
            sql = f"SELECT DISTINCT DORMI_NUM FROM dormitory;"
            cur.execute(sql)
            data_list = cur.fetchall()
```

```

data_list_dormi = list(chain.from_iterable(data_list))

sql = f"SELECT SNAME FROM student_info WHERE SNO = '{self.line_sno}'"
cur.execute(sql)
data_name = cur.fetchall()

if self.line_floor == '' or self.line_dormi == 0 or self.line_bed == '0':
    QMessageBox.warning(self, "提醒", "学生信息不全!", QMessageBox.Ok)
elif self.line_name != data_name[0][0]:
    QMessageBox.warning(self, "提醒", "账号或姓名不正确!", QMessageBox.Ok)
elif self.line_dormi not in data_list_dormi:
    QMessageBox.warning(self, "提醒", "寝室号不正确!", QMessageBox.Ok)
else:
    sql = f"SELECT FLOOR_NUM, DORMI_NUM, BED_NUM FROM dormitory WHERE REMARKS
= '无';"

    cur.execute(sql)
    data_list = cur.fetchall()
    a = [self.line_floor, self.line_dormi, self.line_bed]
    a = tuple(a)
    if a in data_list:
        sql = f"UPDATE dormitory_info SET FLOOR_NUM = '{self.line_floor}', " \
            f"DORMI_NUM = '{self.line_dormi}', " \
            f"BED_NUM = '{self.line_bed}' WHERE SNO = '{self.line_sno}';"
        cur.execute(sql)
        conn.commit()
        self.find_stu()
        QMessageBox.information(self, "标题", "信息修改成功!", QMessageBox.Yes)
    else:
        QMessageBox.warning(self, "提醒", "床位已有人使用!", QMessageBox.Ok)
else:
    QMessageBox.warning(self, "提醒", "输入账号!", QMessageBox.Ok)
else:
    QMessageBox.warning(self, "提醒", "账号不存在!", QMessageBox.Ok)

def back_add(self):
    ui_add.close()

```