## 中国矿业大学计算机学院

**19 级本科生课程设计报告**

课程名称 程序设计综合实践

报告时间 2021.1.6

学生姓名 吕迎朝

学 号 08193109

专 业 数据科学与大数据技术

任课教师 李鸣

## 成绩考核

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 课程教学目标 | | | 占比 | | 得分 |
| 1 | **目标1：**掌握一门计算机高级语言，并能使用特定的软件开发工具，设计、开发、调试及运行应用程序。 | | | 20% | |  |
| 2 | **目标2：**针对具体的应用问题，进行功能需求分析，确定设计目标，并能绘制算法流程图。 | | | 20% | |  |
| 3 | **目标3：**在进行需求分析的基础上，设计软件运行界面、关键类、编写代码，调试并正确运行满足需求的应用程序。 | | | 60% | |  |
| 总成绩 | | | | | |  |
| 指导教师 | |  | 评阅日期 | |  | |

**实验三 学生通讯录**

1 系统概述

该项目目标是主要内容是设计开发一个学生通讯录软件，该学生通讯录采用前后端分离的设计模式,前端采用Flutter， 后端采用Flask，采用json数据进行交互，具备学生信息添加、编辑、删除、查找等功能,此外还通过与教务系统登录接口的整合，是实现了登录操作，后端程序运行在腾讯云服务器上，通过CynosDB实现了数据的云存储，保证了数据的安全和共享。通过云服务器实现了多设备的互联互通。设计过程包括项目分析、界面设计、代码编写和运行调试。通过该项目的开发，掌握了基本的web开发流程。

2 系统设计

2.1 设计目标

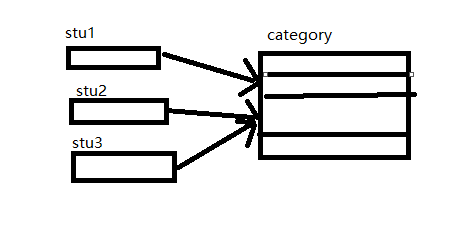
设计出一个学生通讯录，实现基本功能添加，编辑，修改和查找

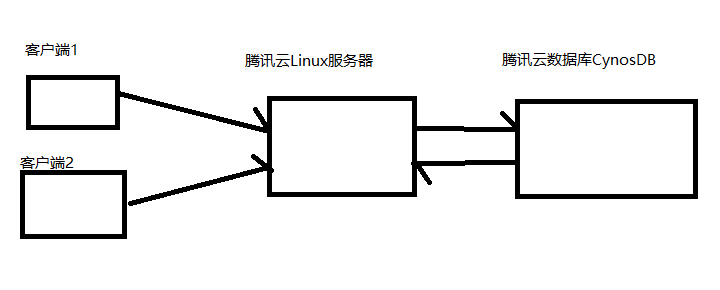
拓展功能：用户注册登录，数据实时云端存储，自定义分组信息，复制手机号并跳转到拨号界面

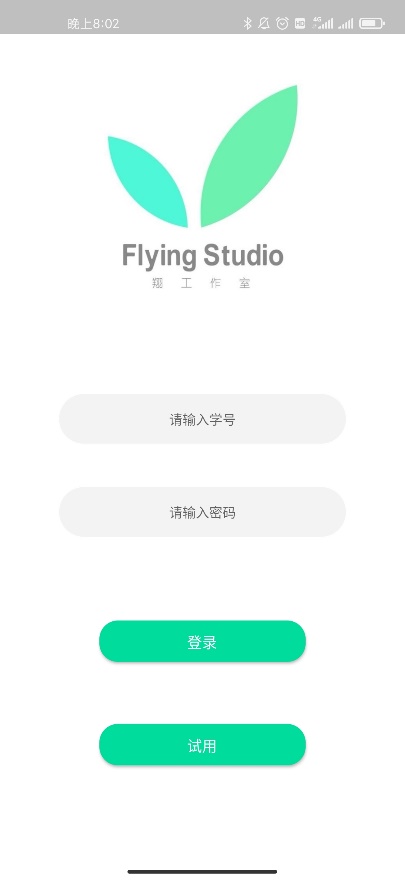
2.2 设计分析与算法流程

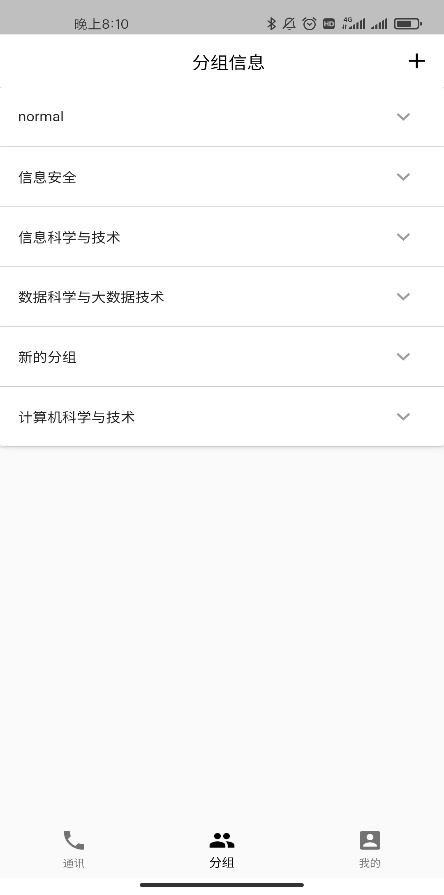
编写apidoc

设计数据库模型



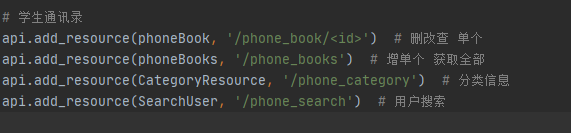
设计模式

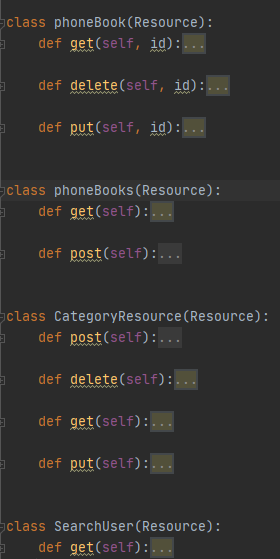
2.3 界面设计





2.4 关键类图





3 系统实现（运行调试）

代码：

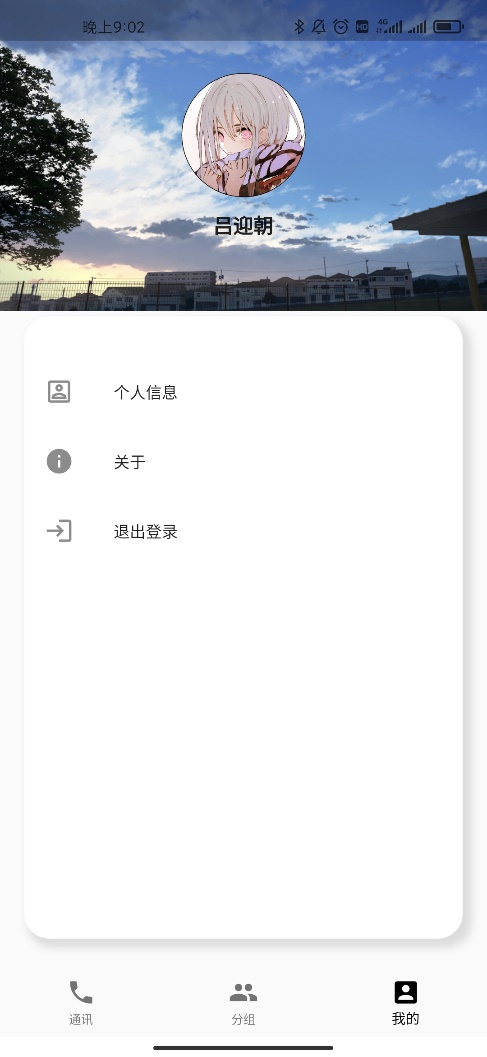
**class** phoneBook(Resource):  
 **def** get(self, id):  
 stu\_info = StuInfo.query.get(id)  
 **if not** stu\_info:  
 abort(404)  
 msg = {  
 **"num"**: 1,  
 **"msg"**: **"获取成功"**,  
 **"data"**: stu\_info  
 }  
 **return** marshal(msg, single\_phone\_fields)  
  
 **def** delete(self, id):  
 **try**:  
 p\_book = StuInfo.query.get(id)  
 db.session.delete(p\_book)  
 db.session.commit()  
 **return** data\_response(1, **"删除成功"**, id, 201)  
 **except** Exception **as** e:  
 **return** data\_response(0, **'删除失败'**, id, 500)  
  
 **def** put(self, id):  
 **try**:  
 args = phone\_book.parse\_args()  
 p\_book = StuInfo.query.get(id)  
 p\_book.stuNumber = args.get(**"stuNumber"**)  
 p\_book.name = args.get(**"name"**)  
 p\_book.sex = args.get(**"sex"**)  
 p\_book.age = args.get(**"age"**)  
 p\_book.birthday = args.get(**"birthday"**)  
 p\_book.address = args.get(**"address"**)  
 p\_book.qq = args.get(**"qq"**)  
 p\_book.mail = args.get(**"mail"**)  
 p\_book.college = args.get(**"college"**)  
 p\_book.belong = args.get(**"belong"**)  
 p\_book.phone = args.get(**"phone"**)  
 p\_book.remarks = args.get(**"remarks"**)  
 db.session.add(p\_book)  
 db.session.commit()  
 **return** data\_response(1, **'修改成功'**, id, 201)  
 **except** Exception **as** e:  
 **return** data\_response(1, e, id, 500)  
  
  
**class** phoneBooks(Resource):  
 **def** get(self):  
 stu\_info = StuInfo().query.all()  
 num = StuInfo.query.count()  
 **if not** StuInfo:  
 abort(404)  
 msg = {  
 **"num"**: num,  
 **"msg"**: **"获取成功"**,  
 **"data"**: stu\_info  
 }  
 **return** marshal(msg, multi\_phone\_fields)  
  
 **def** post(self):  
 args = phone\_book.parse\_args()  
 p\_book = StuInfo()  
 p\_book.stuNumber = args.get(**"stuNumber"**)  
 p\_book.name = args.get(**"name"**)  
 p\_book.sex = args.get(**"sex"**)  
 p\_book.age = args.get(**"age"**)  
 p\_book.birthday = args.get(**"birthday"**)  
 p\_book.address = args.get(**"address"**)  
 p\_book.qq = args.get(**"qq"**)  
 p\_book.mail = args.get(**"mail"**)  
 p\_book.college = args.get(**"college"**)  
 p\_book.belong = args.get(**"belong"**)  
 p\_book.phone = args.get(**"phone"**)  
 p\_book.remarks = args.get(**"remarks"**)  
 **try**:  
 db.session.add(p\_book)  
 db.session.commit()  
 msg = {  
 **"num"**: 1,  
 **"msg"**: **"添加成功"**,  
 **"data"**: p\_book  
 }  
 **return** marshal(msg, single\_phone\_fields), 201  
 **except** Exception **as** e:  
 **return** data\_response(0, e, **'null'**, 500)  
  
  
**class** CategoryResource(Resource):  
 **def** post(self):  
 args = group\_book.parse\_args()  
 group = args.get(**"belong"**)  
 **try**:  
 category = Category()  
 category.group = group  
 db.session.add(category)  
 db.session.commit()  
 **return** data\_response(1, **'分组添加成功'**, group, 201)  
 **except** Exception **as** e:  
 **return** data\_response(0, e, group, 500)  
  
 **def** delete(self):  
 args = group\_book.parse\_args()  
 group = args.get(**"belong"**)  
 **try**:  
 category = Category.query.filter(Category.group.\_\_eq\_\_(group)).first()  
 belong\_list = StuInfo.query.filter(StuInfo.belong.\_\_eq\_\_(group)).all()  
 **for** a **in** belong\_list:  
 a.belong = **'normal'** db.session.add(a)  
 db.session.commit()  
 db.session.delete(category)  
 db.session.commit()  
 **return** data\_response(1, **'ok'**, **'分类已删除, 变为normal'**, 201)  
  
 **except** Exception **as** e:  
 **return** data\_response(1, e, **'error'**, 500)  
  
 **def** get(self):  
 **try**:  
 category\_list = Category.query.all()  
 category\_num = Category.query.count()  
 d1 = {}  
 d2 = {}  
 d3 = {**"userCategoryCount"**: category\_num}  
 d4 = {}  
 ll = []  
 **for** a **in** category\_list:  
 stu\_l1 = StuInfo.query.filter(StuInfo.belong.\_\_eq\_\_(a.group)).all()  
 num = StuInfo.query.filter(StuInfo.belong.\_\_eq\_\_(a.group)).count()  
 l1 = []  
 d2.update({a.group: num})  
 d5 = {}  
 **for** b **in** stu\_l1:  
 l1.append(  
 {  
 **"id"**: b.id,  
 **"stuNumber"**: b.stuNumber,  
 **"name"**: b.name,  
 **"sex"**: b.sex,  
 **"age"**: b.age,  
 **"birthday"**: b.birthday,  
 **"address"**: b.address,  
 **"qq"**: b.qq,  
 **"mail"**: b.mail,  
 **"college"**: b.college,  
 **"belong"**: b.belong,  
 **"phone"**: b.phone,  
 **"remarks"**: b.remarks  
 }  
 )  
 d5.update({**"groupName"**: a.group})  
 d5.update({**"userInfo"**: l1})  
 ll.append(d5)  
  
 d4.update(d3)  
 d4.update({**"userCount"**: d2})  
 d4.update({**"userList"**: ll})  
 **return** data\_response(1, **"ok"**, d4, 200)  
  
 **except** Exception **as** e:  
 **return** data\_response(1, e, **'error'**, 500)  
  
 **def** put(self):  
 **try**:  
 args = ac\_book.parse\_args()  
 g\_id = args.get(**"id"**)  
 group = args.get(**"belong"**)  
 use = StuInfo.query.get(g\_id)  
 use.belong = group  
 db.session.add(use)  
 db.session.commit()  
 **return** data\_response(1, **"修改成功"**, g\_id+**'现在在'**+group, 201)  
  
 **except** Exception **as** e:  
 **return** data\_response(1, e, **'error'**, 500)  
  
  
**class** SearchUser(Resource):  
 **def** get(self):  
 args = search\_name.parse\_args()  
 name = args.get(**'name'**)  
 **try**:  
 name = StuInfo.query.filter(StuInfo.name.ilike(**'%'**+name+**'%'**)).all()  
 i = 0 *# count会报错。。。* **for** a **in** name:  
 i = i+1  
 *# name\_count = StuInfo.query.filter(StuInfo.name.ilike('%'+name+'%')).count()* msg = {  
 **"num"**: i,  
 **"msg"**: **"查询成功"**,  
 **"data"**: name  
 }  
 **return** marshal(msg, multi\_phone\_fields)  
 **except** Exception **as** e:  
 **return** data\_response(0, e, **"error"**, 500)

4 调试

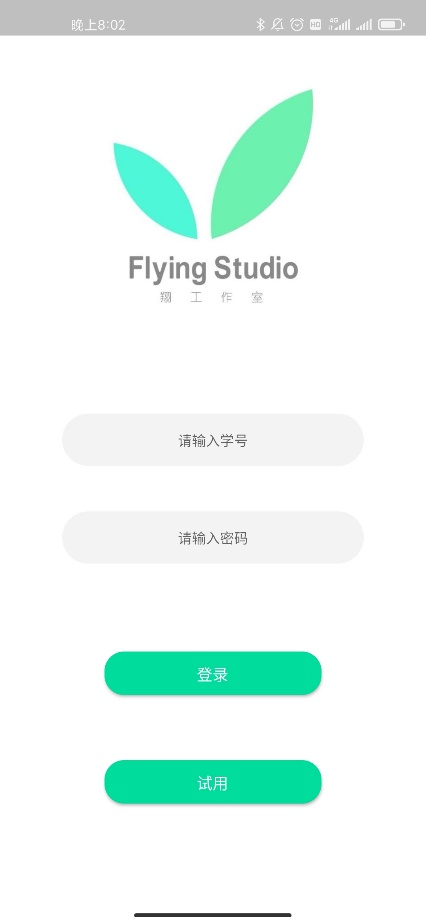
基本功能：

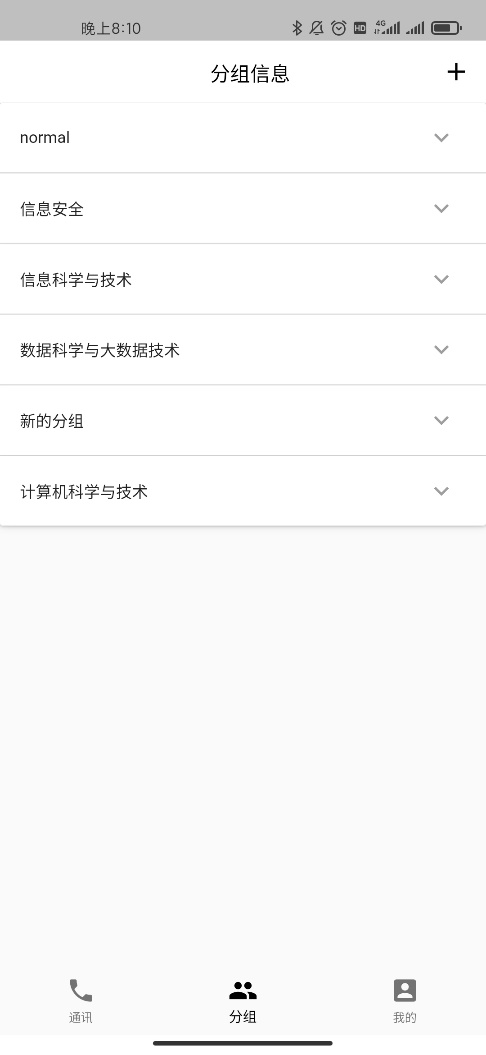
用户信息查看 修改 删除 查找



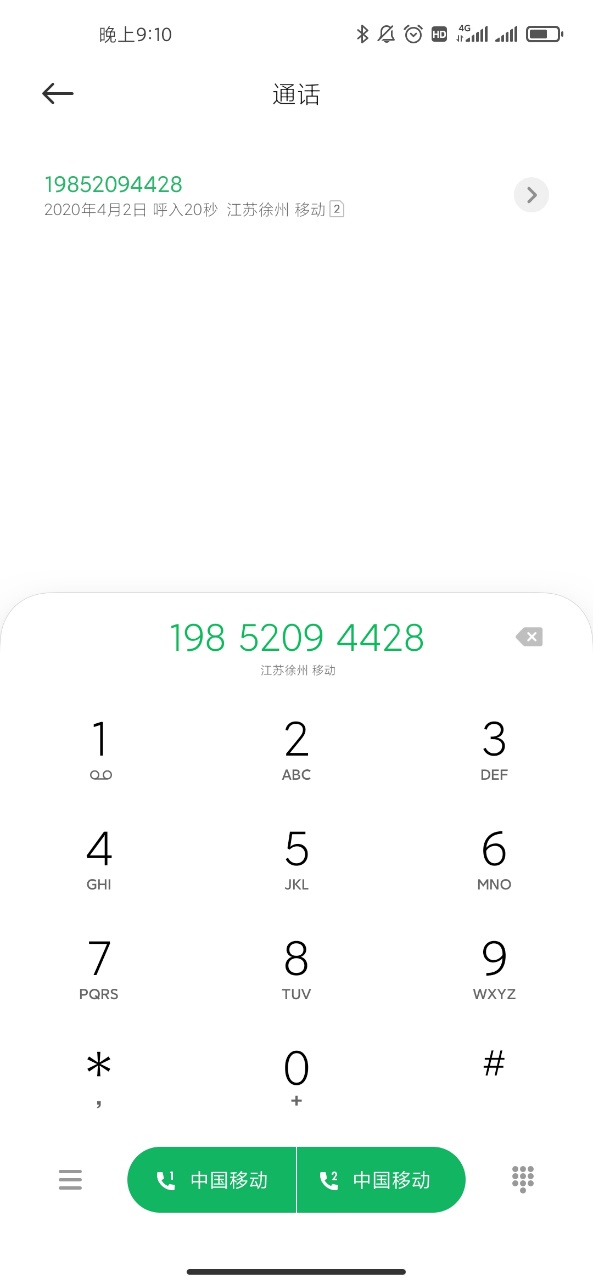
拓展功能

登录

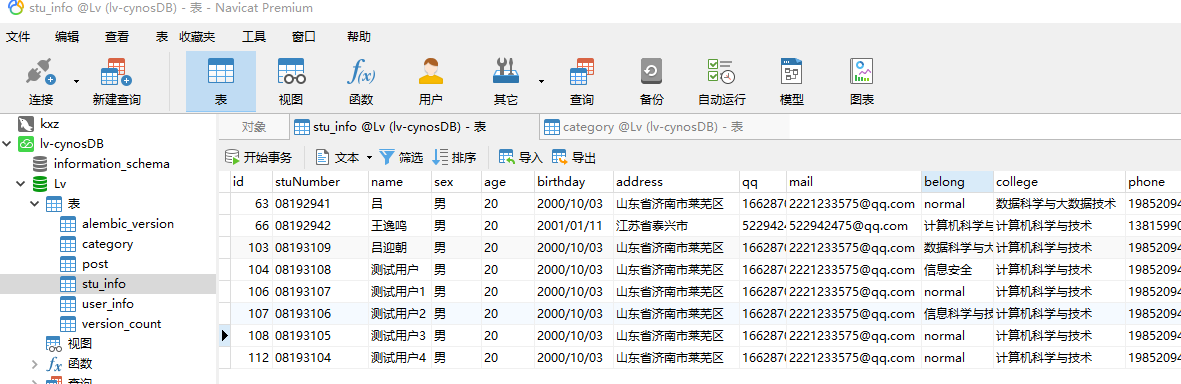


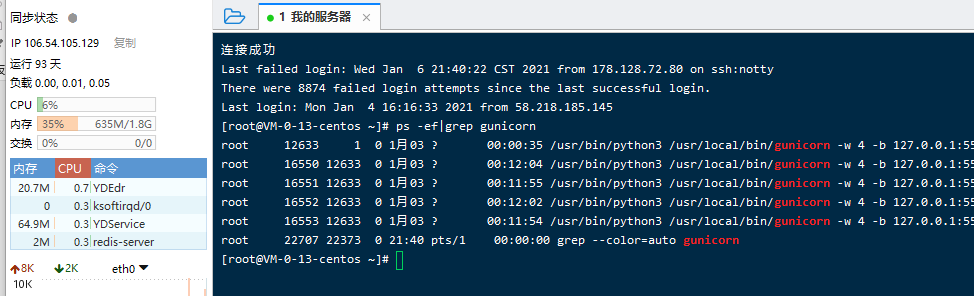
自定义分组

拨号



数据云端存储





域名

https://www.lvyingzhao.cn

5 总结

通过本次实验，我掌握了基本的Web开发流程，实现了代码的本地调试和云端部署，学习了Linux的基本操作，基本的SQL语句操作，Restful风格的使用。在本次实践中，我使用了全新框架，其中遇到了不少Bug，在排查Bug的过程中，我明白了异常处理和日志管理的重要性