

2015-2016 年度

中山大学本科生实验进度报告

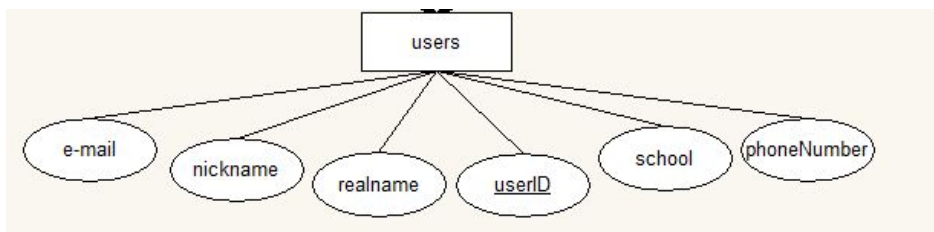
科目：数据库实验

教师：郑贵锋

年级	13 级	专业（方向）	移动信息工程
队长学号	13354485	队长姓名	朱琳
队员 2 学号	13354478	队员 1 姓名	周梦辉
队员 2 学号	13354453	队员 2 姓名	张紫琪

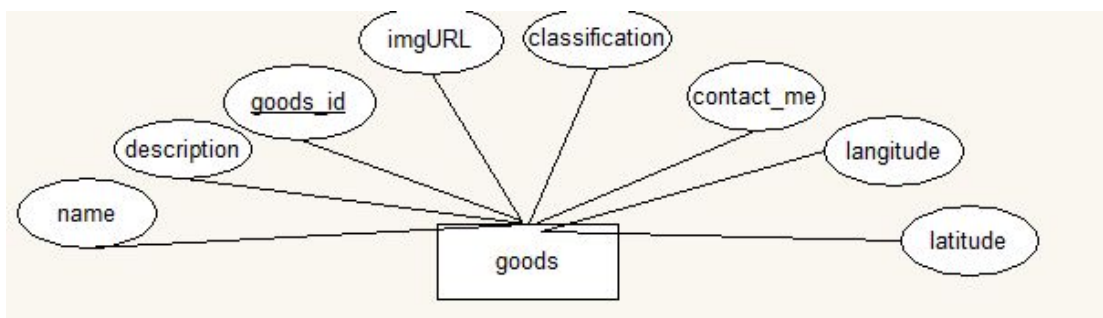
一 完成进度

1.首先我们已经基本完成了在云平台上数据库的搭建。由于数据库维护成本的关系，我们简化了原来设计的数据库的 ER 图，从而搭建了一个更加精简，高效的数据库系统。原来的数据库 ER 图中关于 user 的设计如图：



现在为了充分保护使用者的隐私，我们将 `realname` 去掉了，而将 `phoneNumber` 改为联系人（`contact_me`）这样的形式，放在用户发布的物品信息中。

2.目前实现的就是登陆和注册的功能还有发布商品的功能。数据库建立在云端服务器上，有 `register_and_login` 和 `goods` 两张表，前者用于用户登录注册信息的保存，后者用于保存所有的用户发布的信息。目前的进度是我们在 app 中注册信息请求服务均能成功，但是关于图片的上传和下载还需要进一步实现。`goods` 部分的 ER 图如下，其中 `langitude` 和 `latitude` 记录了发布人的位置（经纬度），我们将根据查找商品的人的位置信息来为他推荐距离比较近的人的商品发布信息。



3.关于 user 部分，我打算使用一个触发器，自动插入 `register` 后的信息到 `user` 当中。不过这部分因为云端链接和计算代码比较繁琐，我们计划在实现云端上传下载 `image` 之后再进行进一步的实现。

4.在云端的数据库与本地的连接，我们使用 Python 代码来进行数据库服务的请求的：代码部分大致如下：

以下是注册和登录的数据库调用方式，发布商品等实现与此类似，不再赘述。

```
class login:
    def POST(self):
        para = web.input()
        username = para['username']
        password = para['password']
        con = Connect()
        query = 'select password from register_and_login where userID="' + str(username) + '"'
        ok = Exec(con, query)
        if len(ok) == 0:
            return 'no'
        else:
            if ok[0][0] == password:
                return 'ok'
            else:
                return 'no'

class regist:
    def GET(self):
        return 'ok'

    def POST(self):
        para = web.input()
        username = para['username']
        password = para['password']
        nickname = para['nickname']
        u = str(username)
        p = str(password)
        n = nickname
        query = 'insert into register_and_login values("' + u + '", "' + p + '", "' + n + '")'
        #TODO:...
        con = Connect()
        try:
            ok = Exec(con, query)
        except:
            return 'no'

        con.commit()
        return 'ok'
```

二 效果展示

1.首先我们在自己写的 app 上填写一个注册信息。



The image shows a mobile app interface for user registration. It has a teal header with the text '用户注册'. Below the header, there are three input fields: '账号:' with the value '13355555', '密码:' with four dots, and '昵称:' with the value 'dddd'. At the bottom, there is a grey button labeled '注册'.

2.我的账户合法，因此会显示我注册成功，直接跳转到登录界面：



The image shows a mobile app interface for user login. It has a teal header with the text '用户登录'. Below the header, there are two input fields: '用户名' and '密码'. At the bottom, there is a grey button labeled '登录'. Below the '登录' button, there is a link labeled '立即注册'. At the very bottom, there is a dark grey button labeled '注册成功'.

3.这时候我检查我云端数据库，发现已经插入该条注册信息。

			userID	password	nick_name
<input type="checkbox"/>			niexnn	vhj	sfg
<input type="checkbox"/>			niexn	123456	rxn
<input type="checkbox"/>			haha	women	nick
<input type="checkbox"/>			Shelly	123456	rxn
<input type="checkbox"/>			1	2	3
<input type="checkbox"/>			3	2	3
<input type="checkbox"/>			hh	hh	'}
<input type="checkbox"/>			123456789	www	nick
<input type="checkbox"/>			12345678	yyy	yyy
<input type="checkbox"/>			xxx	123	uu
<input type="checkbox"/>			qqq	qqq	AAA
<input type="checkbox"/>			AAAA	1235	123
<input type="checkbox"/>			aa	aa	男女
<input type="checkbox"/>			qqqddf	qq1	CD
<input type="checkbox"/>			aaaaaas	1	QQ
<input type="checkbox"/>			123ggg	321	S时时
<input type="checkbox"/>			13354485	dddd	dddd
<input type="checkbox"/>			13355555	ffff	dddd

4.然后直接登录即可.接下来演示发布信息功能：



5.发布后就可以直接在对应该类中找到：



7.此外，数据库还存放了其他卖家发布的商品的信息：



8.点击进去之后就有详细的物品信息显示:



三 接下来拟解决的问题

- 1.实现图片的存放和下载。之前也咨询过 TA 们，有两种方式，一种是将图片转化成二进制流的形式，另一种是将图片放到服务器，然后使用数据库保存图片的 URL。个人认为后者比较好实现一点。
- 2.进一步完善 user 这个表，实现个人信息的完善。这个比较容易，就是繁琐一些。
- 3.我们目前实现根据位置的远近进行商品的推送这个是在写 app 的代码中实现的，下一步准备将他们放到云端去实现，这样能更好的利用资源。

2.进一步完善 user 这个表，实现个人信息的完善。这个比较容易，就是繁琐一些。

3.我们目前实现根据位置的远近进行商品的推送这个是在写 **app** 的代码中实现的，下一步准备将他们放到云端去实现，这样能更好的利用资源。

四 遇到的问题

- 1.遇到了太多的问题，从一开始不会使用云端数据库到玩转数据库
- 2.中文乱码。困扰了很久。后来发现原来是代码中字符串强制转换的原因。
- 3.返回空表不知道如何判断。后来使用字符串长度判断解决了这个问题。
- 4.还有很多不值一提但是积少成多的细节性的问题。

2.中文乱码。困扰了很久。后来发现原来是代码中字符串强制转换的原因。

3. 返回空表不知道如何判断。后来使用字符串长度判断解决了这个问题。

4.还有很多不值一提但是积少成多的细节性的问题。