

北京工业大学

2016-2017学年第二学期

信息学部（软件学院）

课程名称：	移动软件开发		
报告性质：	<input type="checkbox"/> 作业报告 <input checked="" type="checkbox"/> 实验报告		
学号：	14080216	姓名：	姚汝达
任课教师：	湛云莉	课程性质：	专业任选课
学分：	2	学时：	32
班级：	802	成绩：	
小组成员：			
教师评语：			

2017年5月30日

## 目录

一、产设设计 .....	3
1.1产品概念设计 .....	3
1.2架构设计 .....	3
1.3移动软件设计 .....	4
二、产品实现 .....	5
2.1环境搭建 .....	5
2.2移动软件实现 .....	6
2.2.1登录 .....	6
2.2.2登陆逻辑验证 .....	6
2.2.3功能选择主页面 .....	7
2.2.4后台循环执行更新用户位置的服务 .....	8
2.2.5我的位置 .....	8
2.2.6朋友方面 .....	9
2.2.7注册功能 .....	12
2.3后台服务器实现 .....	13
2.3.1后台架构 .....	13
2.3.1接口实施等环境部署 .....	14
2.4后台数据库实现 .....	15
2.4.1用户信息数据表 .....	15
2.4.2用户-用户朋友关系数据表 .....	16
三、问题及经验总结 .....	16
3.1ActionBar的使用 .....	16
3.2Service循环启用的方法 .....	17
3.3各种基于ListView的加载方法 .....	17
3.4Json在Log中的显示形式的迷惑性 .....	17
3.5调取接口去执行为何那么慢 .....	17

# 一、产式设计

## 1.1 产品概念设计

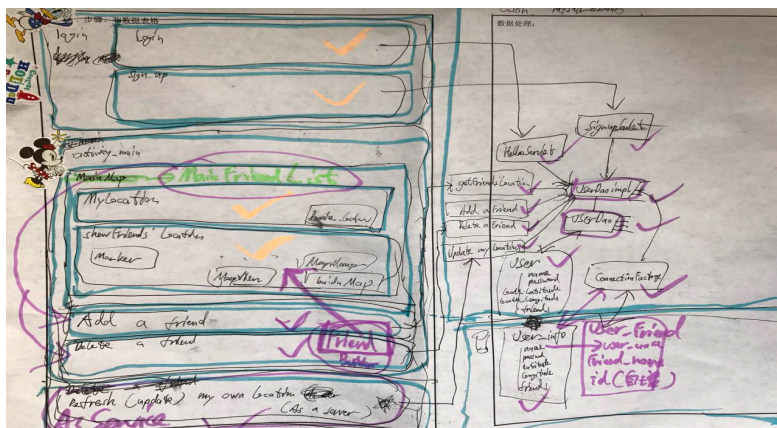
在使用苹果手机的时候，系统的一个应用“查找朋友”能够查看已经添加的朋友的位置，点击某个朋友的选项卡，其实时位置就会显示在地图上……

和一个朋友相互添加好友后，意味着在任何时候（现在大多数人的手机默认都启用定位功能）可以被朋友在未进行任何告知的情况下查看位置。我的亲戚用这个功能查看远在美国儿子的生活情况，我把我最好的几个朋友加入列表与他们共享位置。当前位置是一种很大程度上的隐私，透露意味着存在风险。而在任何情况下对朋友列表中的朋友开启，不仅仅意味着完全的相互信任，使用的时候，无不是关心与牵挂，可以算是对远方亲人、朋友爱的实现的一种途径。

而在安卓平台上，还未有相同机制的位置分享类应用，并且更没有安卓苹果用户相互联通的一个应用，我想和在安卓平台上的母亲，想让两个平台上希望相互实时分享位置的用户使用到类似“查找朋友”的移动软件，因此决定先行开发一个原型版本的安卓平台上的能基本实现如上功能的位置分享类移动软件。我将此应用起名WhatIsGoodFriend，而不是Where...，意在望更多人思考并拥有真正的好朋友，中文名为“鸣啊”，借声音之感，其他并无意义。

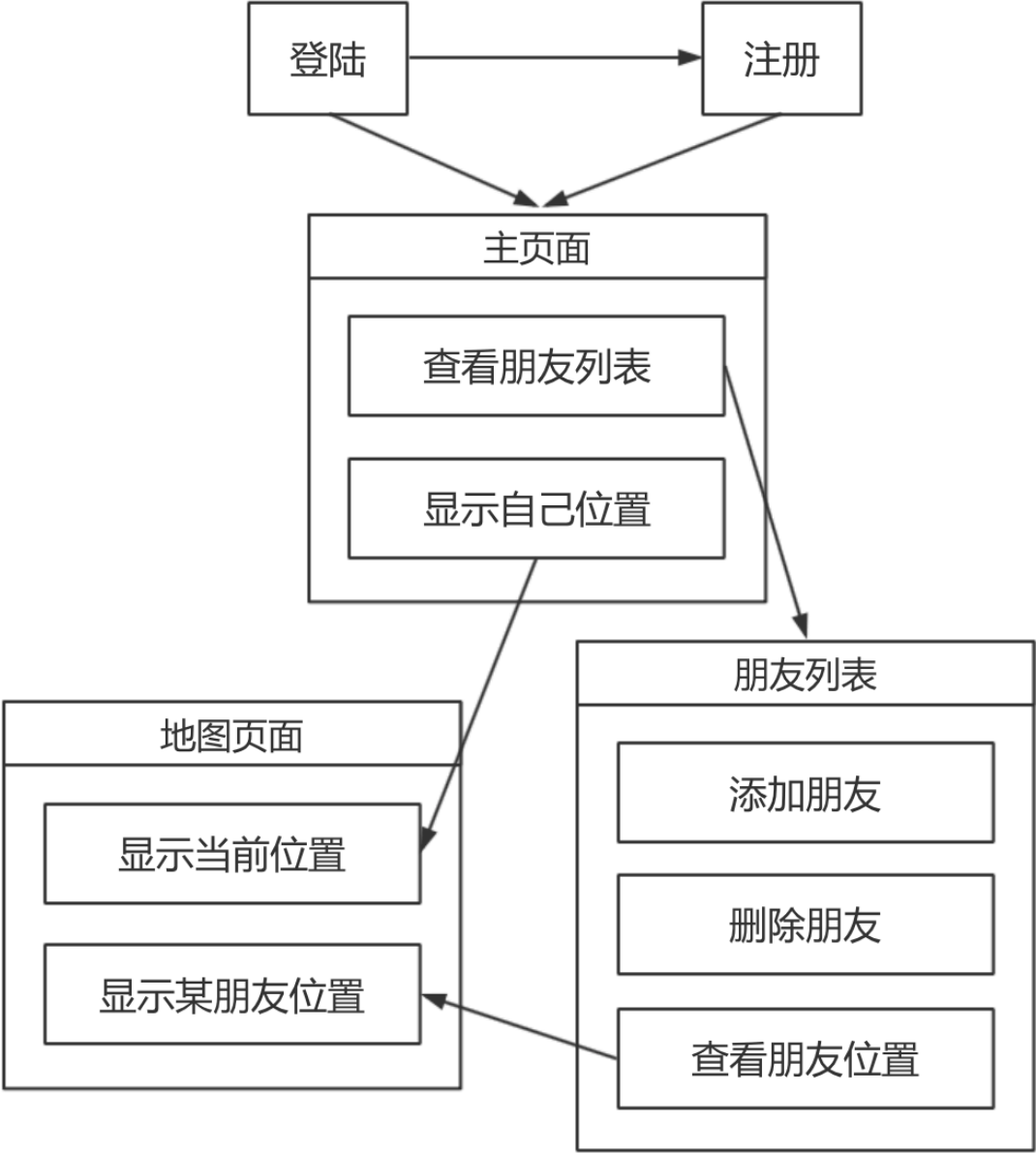
## 1.2 架构设计

因为鸣啊的第一版本只是为了实现基础功能的原型版，故架构设计的要点在于简单与迅速。移动软件端即安卓手机应用，与服务器交互采用json数据格式，服务器采用Tomcat7，响应主要使用Servlet的post方法，数据库使用MySQL。



### 1.3移动软件设计

鸣啊的第一版本为原型版，主要实现用户管理、好友管理、好友位置查看的核心功能。依据如上功能进行设计，界面及响应功能如图所示：

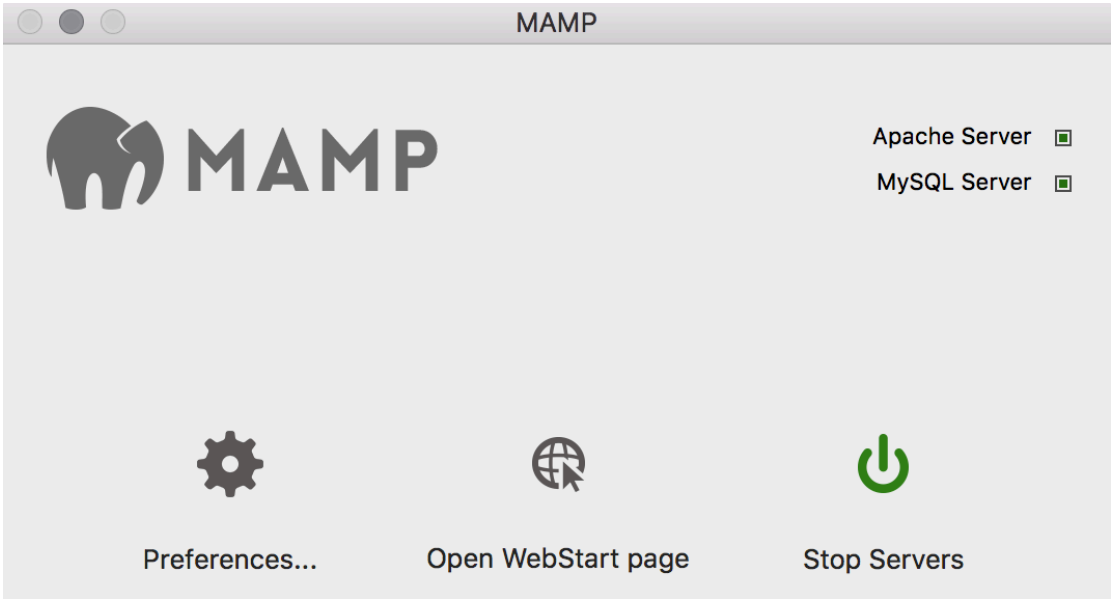


移动软件界面功能设计

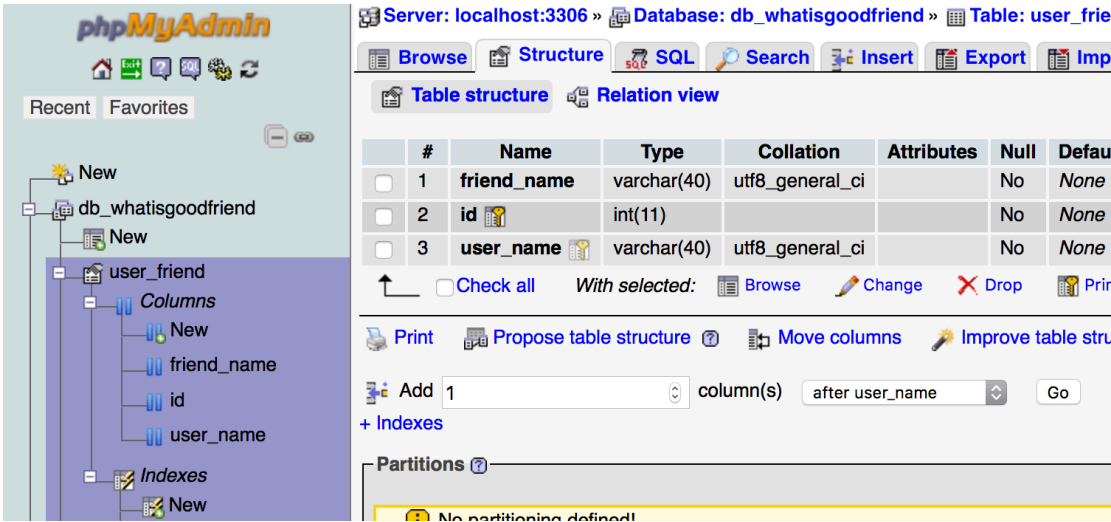
# 二、产品实现

## 2.1环境搭建

因为是个个人开发，所以虚拟机、服务器与数据库在同一个电脑上。电脑为Mac系统，直接使用了MAMP软件运行所需的服务（MySQL、Apache等）、网络配置（端口号等）、数据库管理（PHPMyAdmin）。



MAMP软件界面



PHPMyAdmin网页界面

## 2.2移动软件实现

### 2.2.1登录

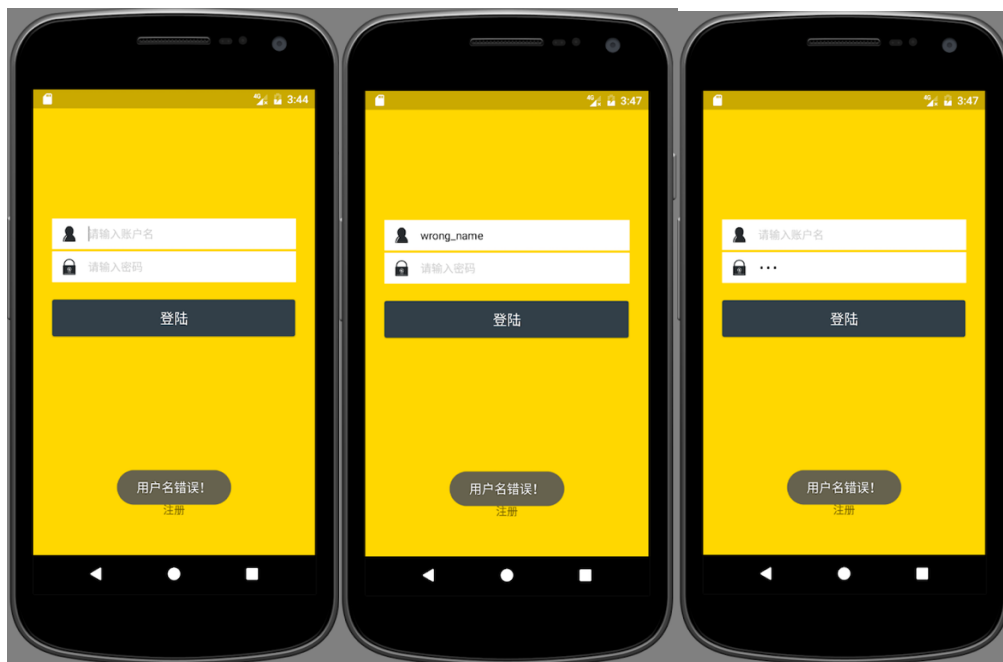


登陆主页面

主要功能：

- 检测输入格式
- 调用登录接口检测用户名是否存在
- 调用登录接口检测密码是否匹配用户名
- 向用户反馈登录情况

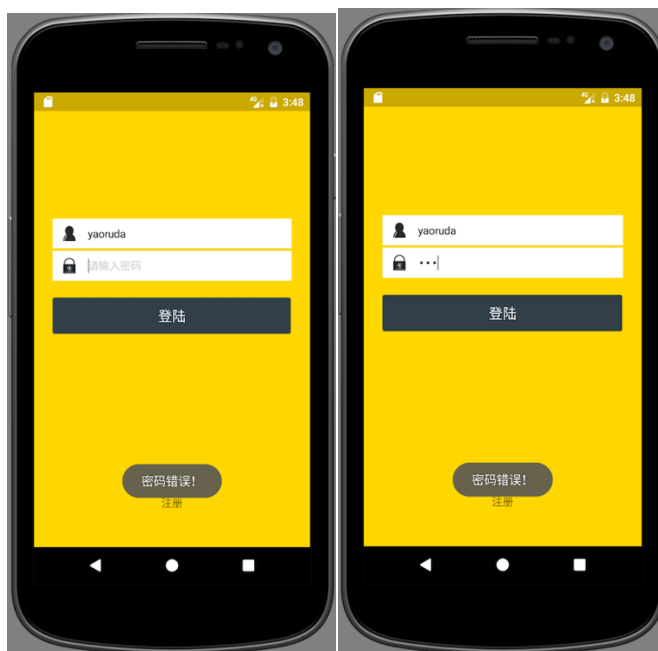
### 2.2.2登陆逻辑验证



用户名未输入

用户名输入错

只有密码



没有密码

密码错误

## 2.2.3功能选择主页面



### 功能选择主页面

主要功能：

- 跳转到各功能界面
- 检测百度地图API状态
- 开启后台更新本账户位置的服务

## 2.2.4后台循环执行更新用户位置的服务

```
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045607374
D/LocationInService: La:39.88
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045612381
D/LocationInService: La:39.88
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045617387
D/LocationInService: La:39.88
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045622392
D/LocationInService: La:39.88
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045627396
D/LocationInService: La:39.88
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045632404
D/LocationInService: La:39.88
D/LocationInService: Lo:116.48999833333333
D/SetLocation_response: success
E/AlarmService: 循环执行1496045637410
D/LocationInService: La:39.88
```

实现方法：

- 设置一个服务和一个服务管理者
- 将服务管理者收到服务执行成功后的方法设置为再一次开启服务
- 将服务的方法设置为等待一段时间后调取当前位置并上传给服务器

## 2.2.5我的位置



我的位置界面

主要功能

- 将自己当前的位置以蓝点的形式显示在地图上
- 根据时间与距离移动实时更新位置



## 2.2.6朋友方面



朋友列表界面

主要功能：

- 点击跳转到查看各朋友位置的界面
- 点击加号可以添加朋友
- 检测添加朋友中朋友账号输入正确性
- 侧滑某朋友Item提供删除方法
- 对朋友系统的逻辑进行用户友好型提示

具体实现方法：

好友列表：

调取后台接口获取用户信息并显示在ListView上，如果两个用户相互都是好友关系，才会作为好友，否则不会显示。

好友位置：

调取后台接口获取保存在对应朋友数据库中的数据（此用户运行自己app的时候更新位置的服务就会将他的位置上传并保存），之后调取百度地图API进行组织与显示。

删除好友：

调取后台的删除好友接口，传入好友账户名，执行，之后调用ListView的Adapt的刷新方法进行内容的刷新，用户即可发现相应的好友被删除。

添加好友：

调用后台添加好友的接口，传入两个参数，分别是当前用户账户和要添加的用户账户。由后台进行检测与反馈。分为1.已经添加过的好友（这种情况下因为对方没有添加此用户为好友导致此用户无法看到，会进行友好型提示）；2.不存在的好友；3.成功添加。都会进行相应提示。



显示相应朋友位置



没有位置时进行提示



侧滑出现删除按钮



删除成功并提示



输入正确朋友账号

添加成功并提示



添加不存在的朋友

添加已添加朋友

## 2.2.7注册功能



注册主页面

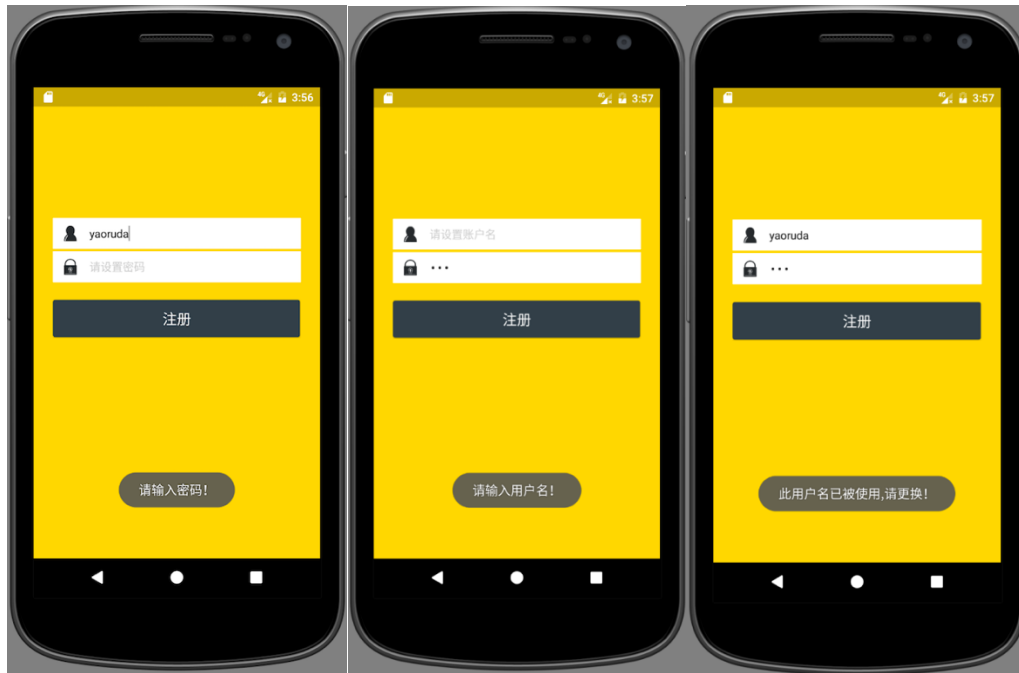
主要功能:

- 检测用户输入的逻辑
- 调用后台接口检测用户输入的正确性
- 反馈注册结果
- 成功即跳转到主页面



输入正确

注册成功



无密码

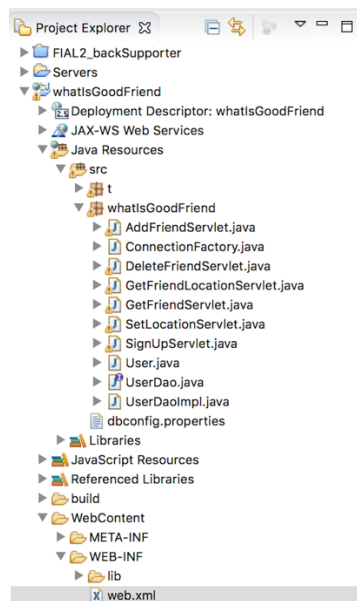
无用户名

用户名已被使用

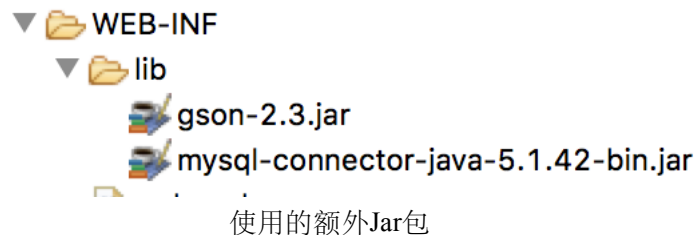
## 2.3后台服务器实现

服务器使用Eclipse开发，使用动态网站工程项目，使用Tomcat7、Gson、连接数据库等使用的中间件及中间件容器。

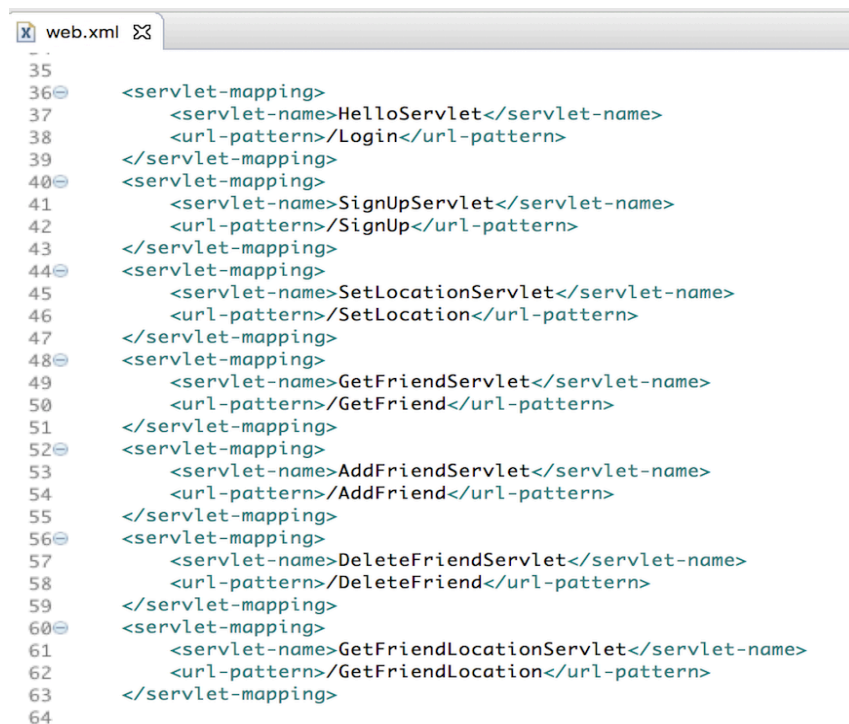
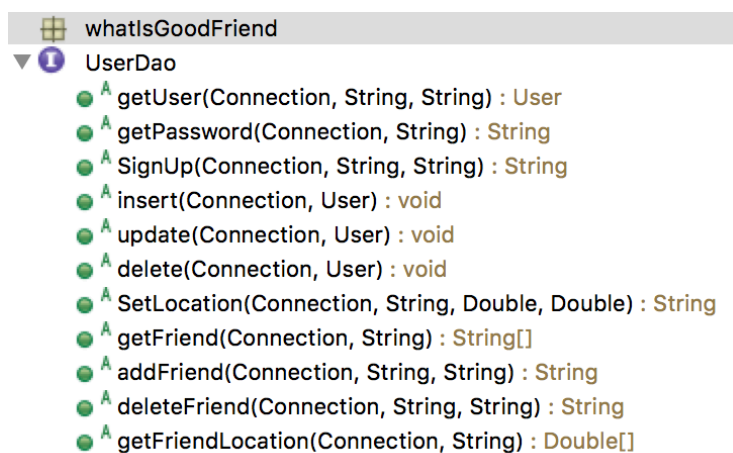
### 2.3.1后台架构



项目结构



### 2.3.1 接口实施等环境部署



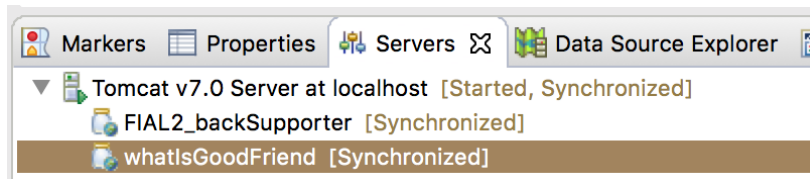
使用Servlet组织接口

```

1 !driver = com.mysql.jdbc.Driver
2 !dburl = jdbc:mysql://localhost:3306/db_whatisgoodfriend
3 !user = root
4 !password = root
5
6 driver = com.mysql.jdbc.Driver
7 dburl = jdbc:mysql://localhost:3306/db_whatisgoodfriend
8 user = root
9 password = root

```

配置链接数据库



Tomcat的使用

```

new Thread((Runnable) () -> {
    try {
        org.apache.http.client.HttpClient httpClient = new org.apache.http.impl.client.DefaultHttpClient();
        HttpGet httpGet = new HttpGet("http://10.0.2.2:8080/whatIsGoodFriend/Login?name="+user_name);
        HttpResponse httpResponse = httpClient.execute(httpGet);
        if (httpResponse.getStatusLine().getStatusCode() == 200){
            HttpEntity entity = httpResponse.getEntity();
            String response = EntityUtils.toString(entity, "utf-8");

            Gson gson = new Gson();
            String password = gson.fromJson(response, String.class);
            if (password != null) {
                Log.d("getjson", password);
                pipedWriter.write(password);
            }else{
                pipedWriter.write("null");
            }
        }
    }catch (Exception e){
        e.printStackTrace();
    }
}).start();

```

用户端中调用接口的方法

## 2.4后台数据库实现

后台数据库的实现:

启用MAMP后自带的PHPMyAdmin的网站管理界面有非常丰富的功能可以创立和管理数据库。

### 2.4.1用户信息数据表

Database: db\_whatisgoodfriend, Table: user\_info, Purpose: Table structure

Column	Type	Null	Default	Comments
id	int(20)	No		
name	char(18)	No		
password	char(40)	No		
latitude	float	Yes	NULL	
longitude	float	Yes	NULL	
friends	text	Yes	NULL	

用户信息表结构设计

Database: db\_whatisgoodfriend, Table: user\_info, Purpose: Dumping data

id	name	password	latitude	longitude	friends
2	yaoruda	123456	39.88	116.49	
7	aaa	qqq			
8	zzc	123456	39.83	116.53	
9	zzc2	123456	39.99	116.33	
10	lishuang	111	39.88	116.49	

截至目前的用户信息表数据

## 2.4.2用户-用户朋友关系数据表

Database: db\_whatisgoodfriend, Table: user\_friend, Purpose: Table structure

Column	Type	Null	Default	Comments
friend_name	varchar(40)	No		
id	int(11)	No		
user_name	varchar(40)	No		

用户朋友关系表结构

Database: db\_whatisgoodfriend, Table: user\_friend, Purpose: Dumping data

friend_name	id	user_name
yaoruda	3	zzc
yaoruda	5	zzc2
zzc2	24	yaoruda
yaoruda	26	aaa
aaa	27	yaoruda
zzc	28	yaoruda

截至目前的用户朋友关系表数据

# 三、问题及经验总结

## 3.1ActionBar的使用

有界面需要返回按钮、剧中的指定样式的标题、右侧还要有对称的一个按钮。最开始用的是Style中的DarkActionBar的Theme，在网上查找资料后发现这个Theme有很多定制化的方法，比如设置返回按钮（甚至阴影）的图标、文字格式，右侧隐藏的拓展按钮的方法，但是用到最后发现不太够，比如文字的样式就有局限性。故只好使用自定义的ActionBar，查找的方法有不少，最终使用了customerActionBar的v7方法。需要注意的是，使用这个方法的时候所使用的Theme必需是有ActionBar的。



### 3.2Service循环启用的方法

鸣啊需要在用户使用自己的时候，不断的上传当前用户的坐标到当前用户的数据库中，所以需要的不止是定时服务，还是循环定时服务。经过查询，没有直接的循环定时启动的服务，所以只能利用服务和消息来实现循环。使用服务管理，在服务结束的时候执行开启同样的这个服务，并在每个服务执行之前设置一定的等待时间就可实现循环定时后台服务了。

### 3.3各种基于ListView的加载方法

ListView是我在鸣啊中使用的可以说是出了Baidu地图控件之外最复杂的了，它最主要的方法是GetView，即每次需要新的Item的时候才会加载，极大的提升了Items展示的流畅性，也同时节约了系统开支。我在使用的时候，在网上发现使用的最多的不是最普通的ListView以及其Adapter，而是各种人写的比如swipeAdapter，而且还要与特定的布局控件结合，比如com.daimajia.swipe.SwipeLayout。就拿他们举例来说，我在主页面使用的是普通的，我需要自己些GetView、生成、赋值的方法。但使用swipeAdapter后，它的GetView是写好动态加载逻辑的，我只需要定义好那些控件要被生成，填写数据的方法是怎样的这些就可以了。但我觉得使用这些还是不好，不仅需要引用额外的jar包，而且定制化程度也大大降低了，万一我根本的加载模式改变，在这里是无法更改的。

### 3.4Json在Log中的显示形式的迷惑性

在一次使用接口获取json格式的数据的时候，因为传回的信息太过简单，我忘记将string类型的json里的数据拿出来就直接用了。测试的时候问题出现了，我就Log来检测，发现一直没有问题（打印出来的就是已经解析出来的string而不带json的格式信息），直到我自己意识到忘了解析了。所以Log来打印一些常见格式的时候还是要慎重考虑，这些打印出的格式是否已经被解析过了，因为Log会自己解析。

### 3.5调取接口去执行为何那么慢

最后一条不是个经验，是我的未解决的问题，服务器和数据库还有虚拟机都在一个电脑上，我单独操作数据库、用浏览器方法直接执行后台Servlet的方法也非常的迅速。但是用虚拟机跑的时候就会很慢，比如登陆就要等个2秒，很难接受，也不知道怎样解决。