**1开发规划**

1.1开发人员

|  |  |  |
| --- | --- | --- |
| 姓名 | 专业年级 | 分工 |
| 钱怡辰 | 16计算机科学与技术 | 项目整体管理和分工，前后端交互设计，客户端程序开发。 |
| 刘凡莉 | 16计算机科学与技术 | 客户端程序开发、软件测试、财务管理。 |
| 王帅君 | 16计算机科学与技术 | 客户端UI设计、服务器端程序开发、数据库设计。 |
| 李隆岩 | 16计算机科学与技术 | 客户端UI设计、服务器搭建、服务端程序开发。 |

1.2开发环境与工具

客户端使用Android Studio软件开发，使用genymotion模拟器和android手机（android 7.0）进行测试。

服务器端使用Eclipse软件开发。数据库使用SQL Server开发。

服务器使用腾讯云服务器，服务器版本是tomcat9.0。

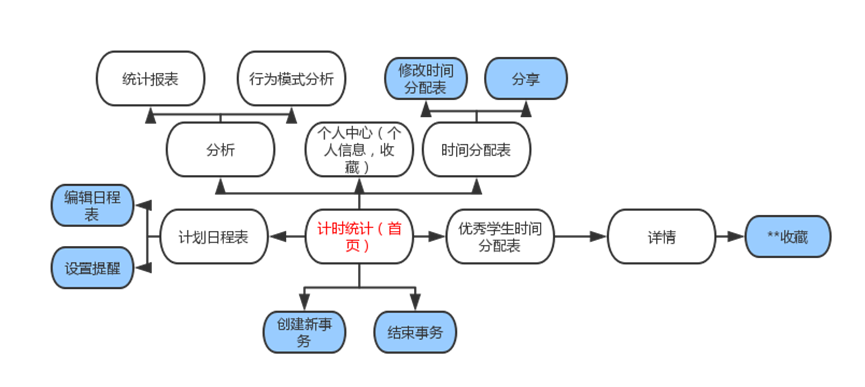
1.3开发规范

编码方式：客户端，服务器端，前后端交互使用UTF-8编码。

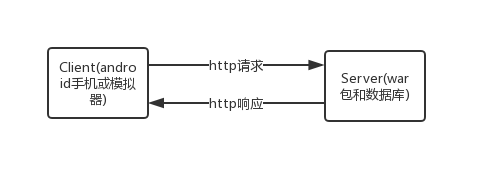
语言版本：使用version 1.8以上版本开发。

**2总体设计**

2.1基本设计结构图



2.2部署结构



2.3模块列表

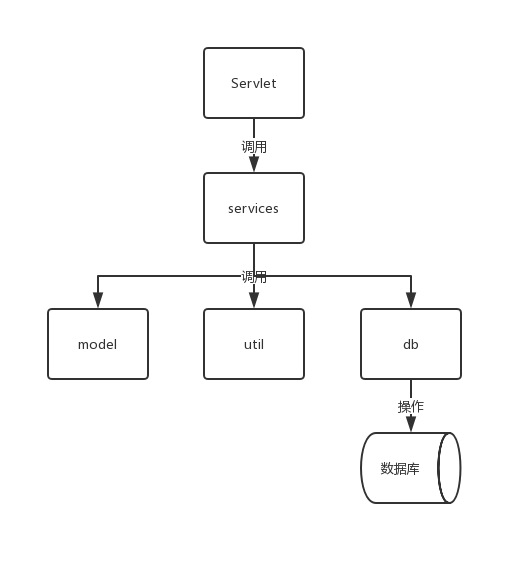
2.3.1客户端部分

|  |  |
| --- | --- |
| 模块 | 功能 |
| UI模块 | 包含所有Activity,完成界面的基本功能设计与用户交互。 |
| Http模块 | 完成各页与服务器端的http交互 |
| Model模块 | 完成与数据库对应的类的定义。 |
| Alarmreceivers模块 | 完成日程提醒功能。 |
| Util模块 | 包含各种工具类，比如持久化存储，时间的格式化处理，token的处理。 |
| Widget模块 | 包含activity要使用的一些组件。 |
| Layout | 包含页面的静态设计。 |
| Drawable | 存储软件中需要的不同分辨率下的图片。 |
| Values | 包含颜色的值，样式类，字符串等。 |

2.3.2服务器端部分

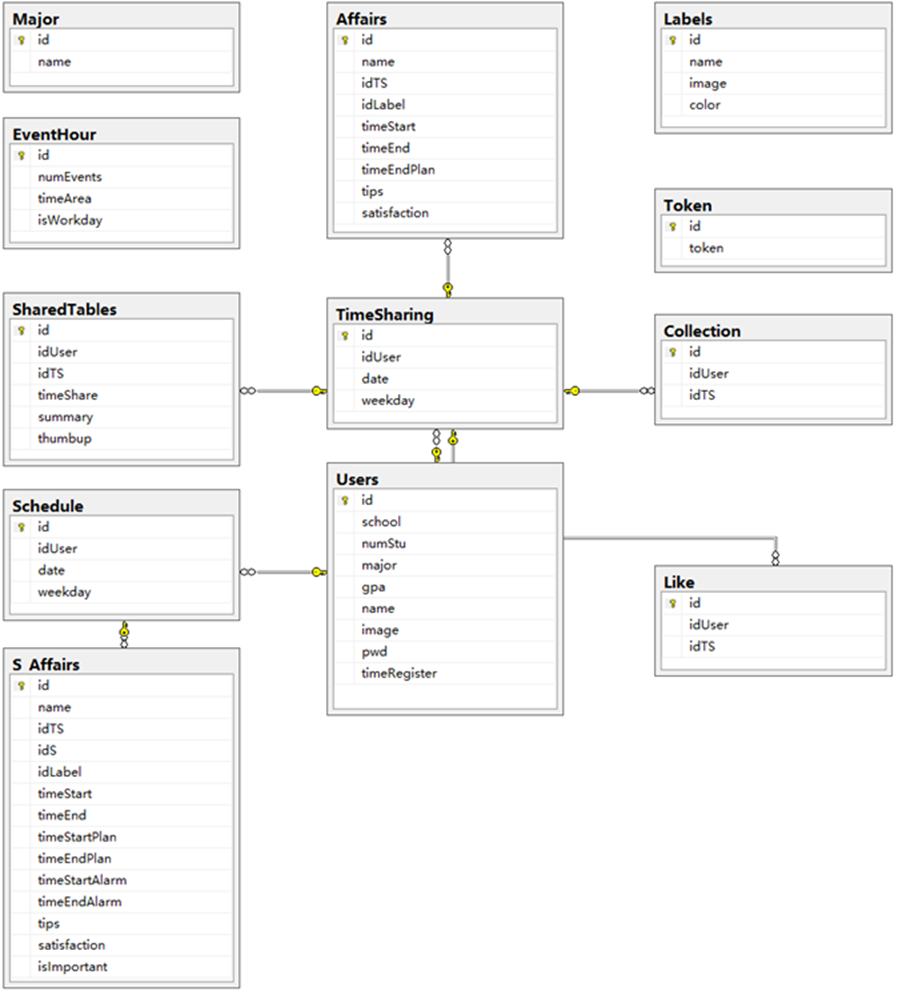
|  |  |
| --- | --- |
| 模块 | 功能 |
| Db模块 | 连接、关闭数据库等基本操作。对每张表进行增删查改处理。 |
| Model模块 | 完成与数据库对应的类的定义。 |
| Services模块 | 一些具体的比如登录，修改时间分配表的具体实现。 |
| Servlet模块 | 完成与前端的交互处理。 |
| Util模块 | 一些工具类的实现，比如处理请求工具类，处理时间格式工具类。 |

服务器端模块示意图：



**3数据库结构**

数据库设计E-R图如下



**4细化模块设计**

|  |  |  |
| --- | --- | --- |
| 模块 | 主要函数 | 功能 |
| AffairService | **public** **boolean** changeAffairById(**boolean** isAffair,**int** id,String startTime,String endTime)  **public** **int** changeAffair(**int** change,**int** isAffair,Affair affair,String date,**int** idU)  **public** **int** DeleteAffair(**int** id,**int** isAffair)  **public** S\_Affair guessAffair(String time,String date,String name) | 根据id修改具体事务的时间、修改（创建）事务、删除事务、查找当天的同名日程 |
| AnalysisService | **public** **static** String getDelayedTime(**int** userId,**boolean** weekday)  **public** **static** **int** getUnfinishedPercent(**int** userId,**boolean** weekday)  **public** **static** **int** getDays(**int** userId) **throws** ParseException  **public** **static** JSONObject getChart(**int** userId,**boolean** weekday,String type) | 统计日程的平均推迟时间、统计日程没有完成的百分比、统计使用APP的总天数、计算得到各类行为模式分析的报表数据。 |
| LikeService | String thumb(**int** userId,**int** idTs) | 用户为欣赏的时间分配表点赞 |
| CollectionService | **public** String collection(**int** idTs,**int** userId)  **public** JSONObject getCollection(**int** userId) | 用户收藏欣赏的时间分配表。  获取某用户收藏的所有时间分配表。 |
| ScheduleService | **public** JSONObject getScheduleTable( String userId,String date) | 获取用户某天的日程表 |
| SharedtableService | **public** JSONObject getMajorList(String majorKeyword) | 根据热度，分享时间等给分享出来的时间分配表排序，将排序后的列表返回。 |
|  | **public** JSONObject getDailySheet(**int** userId,String date)  **public** JSONObject getWeeklyChange(**int** userId,String date,**int** labelid)  **public** JSONObject getWeeklySheet(**int** userId,String date) | 计算统计得到日报表、周报表，周报表趋势（每周各个类型事件的变化趋势）。 |
| TimeSharingService | **public** JSONObject getSharingTable(String userId,String date) | 获取用户某天的时间分配表 |
| UserService | **public** String register(User user,String verify)  **int** judgeVerify(String verify)  **public** **byte**[] getVerify()  **public** String login(String username,String password,String verify,String token) | 注册、登录、生成验证码、判断验证码是否正确。 |

**5接口API**

5.1前后端API

1、UserServlet

1.1 registerServlet

请求：

{

"operation":"register",

"verify":"4635",

"id":233,

"numStu":"13467"//user的其他基本信息

}

返回

{

"status":"success"||"usernamefail"||"verifyfail"

#### 1.2 login

{

"operation":"login",

"name":"lajfla",

"pwd":"fdlvnv",

"verify":"4353",

"token":"gprejgp"

}

{

"status":"success"||”usernamefail"||"passwordfail","verifyfail",

"token":"rgegg"

//user的基本信息

}

#### 1.3 changeInfor

{

"operation":"infor"

"token",

"id":233,

"nameStu":"5240",

}

{

"status":"success"||"fail"||"unlogin"

}

#### 1.4 getUser

{

"operation":"getUser",

"token":,

"name":"爱吃肉的兔子"

}

{

user转化成的json字符串加上

"status":"success"||"fail"||"unlogin"

}

#### 1.5 logout

{

"operation":"logout",

"token":,

"id":13555

}

{

"status":"success"||"fail"||"unlogin"

}

2、TimeSharingServlet

?token userId year month day

{

"id":

"idUser":

"date":

"weekday":

"affair":[{},{}]

"saffair":[{},{}]

"status":"success"||"fail"||"unlogin"

}

3、affairServlet

#### 3.1 创建修改

{

"token":

"sign1"://1 means modify,0 means create

"sign2"://1 means affair,0 means saffair

"date":

"username"://修改版本去掉username这一项

//各种affair消息

}

{

“id”:1333,//affairid

```

"isAffair":true,

"status":"success"||"fail"||"unlogin"

```

}

#### 3.2 删除

{

"token":

"sign1":-1,

"id":

"isAffair":

}

{

"status":"success"||"fail"||"unlogin"

}

#### 3.3 猜日程

{

"token":

"name":

"date":

"time":

}

{

"status":"success"||"fail"||"unlogin"

"hasSAffair":1表示有同名日程，0表示没有

//各种s\_affair信息

}

#### 3.4 修改时间

{

"token":

"sign2"://1 means affair,0 means saffair

"id"://表示要修改的affair或者saffair的id

```

"timeEnd"://什么时候结束该事务

```

}

{

"status":"success"||"fail"||"unlogin"

}

#### 3.5 修改事件时间

{

```

“token”:,

"id":,

"isAffair":,//boolean类型

"StartTime":,

"EndTime":

```

}

{

```

"status":"success"||"fail"||"unlogin"

```

}

4、**AnalysisServlet**

**4.1 获取用户使用总日期**

method=GetDays

{  
   "token":,  
}  
//后端返回形式：jsonobject  
{  
   "days":1234  
}

**4.2 获取平均推迟时间**

method=GetDelayedTime

{  
   "token":,  
   "weekday":"true"||"false"//true表示工作日  
}  
//后端返回形式：jsonobject  
{  
   "delayedtime":"3时12分"     
}

**4.3 获取未完成事件所占比例**

method=GetUnfinishedPercent

{  
   "token":,  
   "weekday":"true"||"false"//true表示工作日  
}  
//后端返回形式：jsonobject  
{  
   "unfininshedPercent":"45%"   
}  
​

**4.4 获取统计图**

method=GetChart

{

"token":,

"weekday":"true"||"false"//true表示工作日,

"type":"SimpleAnalysis"||"detailedAnalysis"||"densityAnalysis"

}

//后端返回形式：jsonobject

//如果前端请求是"type":"SimpleAnalysis"

{

"chartInfor":[{

"labelId":1,

"labelName":"学习"，//标签的名字

"percents":45, //超时或节约比例，45%该项为45，最多保留到小数点后2位

"type":"timeSaving"||"timeOut"//如果相比日程安排节约时间该项为timeSaving，如果超 时为timeOut

},

{

"labelId":2,

"labelName":"休息" ，//标签的名字

"percents":43.72,//45%该项为45，最多保留到小数点后2位，

"type":"timeSaving"||"timeOut"//如果相比日程安排节约时间该项为timeSaving，如果超时为timeOut

}...]

}

//如果前端请求是"type":"detailedAnalysis"

{

​

"chartInfor":[{

"labelId":1,

​

"labelName":"学习"，//标签的名字

​

"percents":[0.45,0.13,0.42], //浮点型数组形式 分别表示少于，符合，超过预计时间的比例

​

},

​

{

"labelId":2,

​

"labelName":"休息" ，//标签的名字

​

"percents":[0.45,0.13,0.42], //浮点型数组形式 分别表示少于，符合，超过预计时间的比例

​

}...]

​

}

//如果前端请求是"type":"densityAnalysis"

{

​

"density":6.88， //表示迄今为止，平均每小时完成的事件数,保留到小数点后2位

​

“chartInfor”:[{ //数组总共7项，对应7个时间段的行为密集程度

​

"period":"凌晨"，//时间段的名称

​

"percents":5.50, //表示该时间段每小时平均做的事件数，保留到小数点后2位

​

},

​

{

​

"period":"早晨"，//时间段的名称

​

"percents":5.50, //表示该时间段每小时平均做的事件数，保留到小数点后2位

​

}..]

​

}

5.2数据库API

**SqlServerManager**

* - url(static String)
* - username(static String)
* - password(static String)
* - con(static Connection)
* - stmt(static Statement)
* - rs(static ResultSet)
* + Connect():连接数据库
* + Create():创建数据表
* + Close():关闭数据库

**AffairManager**

* + add():增加Affair,成功时返回true
* + delete():删除Affair,成功时返回true
* + change():更改Affair,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIdLabel():使用Label的Id进行查找,返回动态数组,没有时返回null
* + findWithSatisfaction():使用满意度进行查找,返回动态数组,没有时返回null
* + findWithName():使用用户名进行查找,返回一个动态数组,没有时返回null

**CollectionManager**

* + add():增加Collection,成功时返回true
* + delete():删除Collection,成功时返回true
* + change():更改Collection,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIdUser():使用User的Id进行查找,返回动态数组,没有时返回null
* + findWithIdTS():使用时间分配表的Id进行查找,返回动态数组,没有时返回null

**EventHourManager**

* + add():增加EventHour,成功时返回true
* + delete():删除EventHour,成功时返回true
* + change():更改EventHour,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIsWorkday:通过判断是否为工作日进行查找,返回动态数组,没有时返回null
* + findWithTimeArea():通过时间段编号进行查找,返回动态数组,没有时返回null

**LabelManager**

* + add():增加Label,成功时返回true
* + delete():删除Label,成功时返回true
* + change():更改Label,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithName():使用用户名进行查找,返回一个动态数组,没有时返回null
* + findWithColor():通过标签的颜色（即种类）进行查找,返回动态数组,没有时返回null

**LikeManager**

* + add():增加Like,成功时返回true
* + delete():删除Like,成功时返回true
* + change():更改Like,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIdUser():使用User的Id进行查找,返回动态数组,没有时返回null
* + findWithIdTS():使用时间分配表的Id进行查找,返回动态数组,没有时返回null

**S\_AffairManager**

* + add():增加S\_Affair,成功时返回true
* + delete():删除S\_Affair,成功时返回true
* + change():更改S\_Affair,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIdLabel():使用Label的Id进行查找,返回动态数组,没有时返回null
* + findWithIdTS():使用时间分配表的Id进行查找,返回动态数组,没有时返回null
* + findWithIdS():使用日程表的Id进行查找,返回动态数组,没有时返回null
* + findWithSatisfaction():使用满意度进行查找,返回动态数组,没有时返回null
* + findWithName():使用用户名进行查找,返回一个动态数组,没有时返回null
* + findWithIsImportant:通过判断是否为重要事件进行查找,返回动态数组,没有时返回null

**ScheduleManager**

* + add():增加Schedule,成功时返回true
* + delete():删除Schedule,成功时返回true
* + change():更改Schedule,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIdUser():使用User的Id进行查找,返回动态数组,没有时返回null
* + findWithDate():使用日期进行查找,返回动态数组,没有时返回null
* + findWithWeekday():使用星期几的编号进行查找,返回当前周的动态数组,没有时返回null(另一种按日期查找)

**SharedTableManager**

* + add():增加Schedule,成功时返回true
* + delete():删除Schedule,成功时返回true
* + change():更改Schedule,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithTimeShared():通过分享的时间查找,返回一个人变量，没有时返回null
* + findWithTimeShared():通过点赞数查找,返回动态数组,没有时返回null

**TimeSharingManager**

* + add():增加TimeSharing,成功时返回true
* + delete():删除TimeSharing,成功时返回true
* + change():更改TimeSharing,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithIdUser():使用User的Id进行查找,返回动态数组,没有时返回null
* + findWithDate():使用日期进行查找,返回动态数组,没有时返回null
* + findWithWeekday():使用星期几的编号进行查找,返回当前周的动态数组,没有时返回null(另一种按日期查找)

**TokenManager**

* + add():增加Token,成功时返回true
* + delete():删除Token,成功时返回true
* + change():更改Token,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithToken():使用Token进行查找,返回一个变量,没有时返回null

**UserManager**

* + add():增加User,成功时返回true
* + delete():删除User,成功时返回true
* + change():更改User,成功时返回true
* + findWithId():使用Id进行查找,返回一个变量,没有时返回null
* + findWithNumStu():使用学号进行查找（在思考是否需要传入学校）,返回一个变量,没有时返回null
* + findWithSchool:通过学校进行查找,返回动态数组,没有时返回null
* + findWithMajor:通过专业进行查找(在思考是否需要传入学校),返回动态数组,没有时返回null
* + findWithName():使用用户名进行查找,返回一个动态数组,没有时返回null
* + findWithTimeRegister():通过注册时间查找,返回动态数组,没有时返回null

**6第三方组件**

|  |  |
| --- | --- |
| 客户端 | com.android.support:appcompat-v7:26.1.0 |
| com.android.support:percent:26.0.0-alpha1 |
| com.android.support.constraint:constraint-layout:1.0.2 |
| com.alibaba:fastjson:1.2.47 |
| junit:junit:4.12 |
| com.android.support:recyclerview-v7:26.1.0 |
| com.github.PhilJay:MPAndroidChart:v3.0.3 |
| 服务器端 | Fastjson-1.2.47.jar |
| Jaudiotagger-2.0.3.jar |
| Jl1.0.1.jar |
| Sqljdbc42.jar |