# 我尚运动客户端功能设计

|  |  |
| --- | --- |
| 编号 | MGN001 |
| 作者 | 张悦，高飞，邱俊杰 |
| 版本 | V4 |

# 功能模块设计

[我尚运动客户端功能设计 1](#_Toc367609108)

[功能模块设计 2](#_Toc367609109)

[更新历史 3](#_Toc367609110)

[文档说明 3](#_Toc367609111)

[术语 3](#_Toc367609112)

[新手导引功能： 4](#_Toc367609113)

[用户登录页面： 4](#_Toc367609114)

[记住密码和自动登录 4](#_Toc367609115)

[忘记密码 4](#_Toc367609116)

[运动简报页面： 4](#_Toc367609117)

[数据刷新 5](#_Toc367609118)

[切换时间 5](#_Toc367609119)

[运动历史页面： 5](#_Toc367609120)

[历史列表 5](#_Toc367609121)

[日历呈现 5](#_Toc367609122)

[历史详情 5](#_Toc367609123)

[群组信息页面： 6](#_Toc367609124)

[排名跳转设计实现 6](#_Toc367609125)

[搜索功能的实现 7](#_Toc367609126)

[大头像的显示的功能实现 8](#_Toc367609127)

[活动信息页面： 9](#_Toc367609128)

[设置页面： 9](#_Toc367609129)

[后台杀进程注意问题 10](#_Toc367609130)

[图片加密 10](#_Toc367609131)

## 更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 更新原因 | 更新人 | 更新日期 |
| V01 | 创建文档 | 高飞 | 2013/9/17 |
| V02 | 增加群组相关功能 | 张悦 | 2013/9/18 |
| V03 | 清除内存与图片名加密 | 邱俊杰 | 2013/9/18 |
| V04 | 文档整理，按照不同的功能界面增加功能说明 | 高飞 | 2013/9/22 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 文档说明

描述了我尚运动项目中手机客户端功能模块实现.方便开发人员使用.

## 术语

注意：MJK

M：Mobile 客户端

JK：接口

GN：功能

新手导引功能：

用户登录页面：

运动简报页面：

运动历史页面：

群组信息页面：

活动信息页面：

设置页面：

## 新手导引功能：

滑屏换页

首次进入

## 用户登录页面：

### 记住密码和自动登录

#### 功能描述：

输入用户名和密码登录软件，可以选择“记住密码”和“自动登录”，记住密码勾选后，如果正确登录则始终缺省在密码栏中补上用户的正确密码。用户更改密码和找回密码后也保持密码。

#### 相关功能：

找回密码

更改密码

#### 实现方法：

在原有已设

### 忘记密码

## 运动简报页面：

当天/7天/30天 运动信息 数字显示当天/7天/30天的运动**步数、距离、时常和热量**【有效步数】，图表显示当天运动的有效步数和强度

### 数据刷新

#### 功能描述：

用户点击右上角刷新按钮，获取最新运动数据。

#### 实现方法：

### 切换时间

#### 功能描述：

向前或者向后切换到其他的时间（按1天、7天和30天）。

#### 实现方法：

## 运动历史页面：

### 历史列表

列表形式显示最近一段时间的运动历史情况，包括每次上传时间、运动步数和消耗热量

### 日历呈现

已日历形式呈现历史运动数据

### 历史详情

历史上某天的运动详情与简报类似。显示某一天运动的步数、有效步数和强度

## 群组信息页面：

昨日个人排名列表：

显示昨日个人排名信息，包括日期、个人信息（姓名、步数、排名）、所有人排名列表（头像、姓名、组别、步数）

昨日群组排名列表：

显示昨日群组排名信息，包括日期、群组信息（组名、步数、排名）、所有群组排名列表（头像、姓名、组别、步数）

昨日组内排名列表：

显示昨日本组排名信息，包括日期、个人信息（姓名、步数、排名）、组内人员排名列表（头像、姓名、组别、步数）

昨日其他组排名列表：

显示昨日其他组排名信息，包括日期、个人信息（姓名、步数、排名）、组内人员排名列表（头像、姓名、组别、步数）

### 排名跳转设计实现

#### 功能描述：

用户点击自己的主条目时，根据用户目前排名，跳转到用户目前排名的位置。当用户的排名在目前排名显示的Listview范围之内，则直接跳转到当前排名位置，否则弹出一个dialog，其中显示用户目前排名的前二十以及后二十名的数据。

#### 实现方法：

在原有已设计的ListView的基础之上，通过当前所在的群组类型（全部排名，组内排名，群组排名。一天，七天），通过MHealthProviderMetaData获取当前类型的排名的总条目数，并与当前mRankingDate获取得到对应的用户排名比较，如果用户排名小于当前类型条目数，则表示用户的数据在ListView的显示范围内，则直接使用mListview.setSelection方法来跳转到当前用户排名条目的位置，否则弹出一个dialog框来展示数据。

dialog框其主要使用buildDialog方法来实现的，Dialog在使用的时候需要在创建对象的时候指定所属的Style样式，样式可以参考R.style.dialog\_fullscreen文件进行设定，其主要设定为继承系统dialog的样式并设定全屏选项。由于某些手机系统dialog的设定不为全屏所以无法使所有手机都能全屏显示。

dialog中主要嵌套了一个listview，按照正常方式将从手机本地数据库获得的40条数据显示在listview中即可。

由于支持不在线查看本地缓存排名，所以需要在程序联网时载入数据的过程中一并将用户排名前后二十位存入数据库，这排名分为1天一级7天平均，在sqlite中使用一个标志位字段区别即可。

具体存数据库实现方法可以参照RankActivity中的updateMyRankInfo()方法。

### 人员搜索功能的实现

#### 功能描述：

在titlebar的左边有一个可以点击的放大镜图案，其主要作用是在屏幕上方弹出一个搜索框。并用来实现搜索功能。

#### 实现方法：

弹出的搜索框使用dialog内嵌view的方式来实现，view中包含一个自定义的MyAutoCompleteTextView，以及对应的MyAutoCompleteAdapter，其主要功能是为了弥补AutoCompleteAdapter在设置数据的不足，原AutoCompleteAdapter只能通过选择的String来进行补全操作，而本程序中需要通过拼音缩写，全拼音，汉字，以及电话号码四中方法来进行补全操作，并只显示补全的汉字名称。

配合MyAutoCompleteAdapter的使用，本例中使用了MyAutoCompleteTextView来对原本的AutoCompleteTextView进行控制。MyAutoCompleteTextView主要设定了AutoCompleteTextView的一些属性，并提供了一些返回的接口。具体设定可以参考代码。

MyAutoCompleteAdapter继承了BaseAdapter，由于AutoCompleteTextView默认点击后通过Adapter的getItem获得所需的String，所以必须在getItem返回String值，二在展示补全条目的时候可以通过自定义的view展示名称以及全拼音字母。

点击后回通过setOnItemClickListener，获取点击的item数值，由于测试时发现，setOnItemClickListener并不能传递到上一层，所以本例中另创建了一个接口setOnNameReturnedListener用于返回所选择的的String值，并用于搜索。

在RankActivity中设定setOnNameReturnedListener，然后通过子线程在lisnster中通过setSearchmode方法改变RankListViewFragment第一个页面的ListView 的Adapter的类型为展示搜索数据类型，（搜索类型是根据原Adapter加工而成，具体代码可以参考RankListViewFragment中的MySimpleAdapter），通过网络搜索到的数据口可以展示到排名条目中与自己的排名条目进行对比了。如果网络不佳，没有搜索到数据，则会返回一个toast提示无法找到数据，用户可以返回或者重试。

### 班组搜索功能的实现

#### 功能描述：

在titlebar的左边有一个可以点击的放大镜图案，其主要作用是在屏幕上方弹出一个搜索框。并用来实现搜索功能。本地数据库在第一次登录后获取对应公司的班组数据。

#### 相关功能：

多服务器统一客户端

登录、登出

#### 实现方法：

增加接口：

<http://phr.cmri.cn/data/openClientApi.do?action=pedtargetgroupseq&userid=13811029472&psw=681301&targetgroupid=35>

获取对应组的信息；

{"group7info":{"distant":"3860","calory":"","step":"6778","complianceday":""},"status":"SUCCESS","group1info":{"distant":"4257","calory":"","step":"6974","complianceday":""},"groupname":"无线-无线产品软件功能研究室","groupid":"35","group7seq":"38","group1seq":"39"}

### 大头像的显示的功能实现

#### 功能描述：

用户在排名列表点击自己头像的时候可以跳出大图。

#### 功能实现：

由于popupwindow以及dialog都不可以实现自动以位置的动画，所以本例暂不使用上述两种（虽然更简单），本例中使用的动画依赖于View的setAnimation方法，所以必须使用一个view来执行动画。因此在排名列表的listview上加了一层framelayout，用于在listview之上显示大头像。

当用户点击listview的小头像的时候，大头像的relateivelayout就设定为visiable，当图像显示之后，用户再次点击，则设定为gone或者invisible（在fragmenet中使用gone容易使得view丢失属性数据，所以尽量使用invisible）。

动画使用teen动画参数设定，具体代码可以参考RankListViewFragment中的createPWforFace方法。

图像显示使用了已存在的imageutil，本例中进行了一些改造：

图像的读取使用了imageview（唯一） 的tag来方式错图，当tag中与子线程中设定的tag不合时，则不向imageview设定图片，避免了因网速慢导致的错图现象。与此同时小头像错图的bug也因此解决。

### 图片文件名加密

#### 功能描述：

服务器保存的图片名称为手机号\_版本号.jpg(如：13552019273\_12.jpg),网络下载的图片会被Android系统抽取缩略图并放到相册中，所以需要进行处理避免被系统显示，也避免泄露手机号。

#### 功能实现：

* 文件名加密

起初使用的是Base64加密，可双向，但是加密出来的文件名很奇怪，都是MTM开头还有就是 ==、= 号结尾，当Bitmap.decodeStream的时候读文件名总是出问题，在eclipse debug watch的时候总是看到结尾有/r/n，然后读取失败，然后改了MD5加密。

加密步骤：

网络拿到图片URL -->储存数据库-->取出URL访问Intent -->MD5加密写入SDCard.

当然先取出图片名加密后与本地图片校对是否已经下好图片。

图片的缓存应用到了强引用和软引用以及弱引用.

强引用用到的是LinkedHashMap

**private** **final** HashMap<String, Bitmap> sHardBitmapCache = **new** LinkedHashMap<String, Bitmap>(*10*, 0.75f, **true**) {

**private** **static** **final** **long** *serialVersionUID* = 1L;

@Override

**protected** **boolean** removeEldestEntry(LinkedHashMap.Entry<String, Bitmap> eldest) {

**if** (size() > *HARD\_CACHE\_CAPACITY*) {

//eldest.getKey() 为集合顺序最前面的地址

*sSoftBitmapCache*.put(eldest.getKey(), **new** SoftReference<Bitmap>(eldest.getValue()));

**return** **true**;

} **else**

**return** **false**;

}

};

这个强引用的最大容量是10，也就是size最大为10，当size大于10的时候在put的数据的时候集合会按照先进先出的原则把前面的数据移除，会调用removeEldestEntry 这个方法，eldest.getKey()移除的的key，eldest.getValue()移除的value，我们把这些移除的数据存到软引用。强引用保证最近的浏览的10张图片不会被回收。犹豫限制了引用的大小，也不会发生OOM。

## 活动信息页面：

显示活动信息:

显示活动信息，分为：正在进行、已经结束和即将开始

查看活动详情:

查看活动的详细信息（待补充）

## 设置页面：

基本信息显示:

显示本人基本信息，包括头像、电话、所属组、运动目标

头像设置 设置本人头像

运动目标设置 设置本人的运动目标，即每天运动步数

版本显示 显示软件当前版本

检测新版本 检测是否有新版本

更改密码 修改客户端登陆密码

意见反馈 反馈使用意见

退出登陆 注销用户并退出软件

## 系统相关能力：

### 后台杀进程注意问题

一些安全软件比如360安全卫士和QQ管家，它们都有清理缓存的功能，其实这个清理缓存就是直接把后台运行优先级较低的进程杀掉，从而优化了系统的内存。

当你的程序进程被杀在点进去的时候会重新创建生命周期，但是一些传递进来的数据会丢失，所以容易出现拿不到数据之类的问题。可以在onSaveInstanceState()方法里保存一些数据，在onCreate的时候再从bundle里取出数据。

项目中群组模块最容易出现后台杀进程空指针异常，fragment+ViewPager 实现左右滑动的时候数据的操作和更新时候需要注意

不可用此方法拿fragment的引用

RankListViewFragment currentFragment1 = (( RankListViewFragment) mFragmentsList.get(i));

因为这样拿到的引用不的一定是当前的View的fragment，所以在操作的时候总是会报空指针异常。

可以通过ViewPager的instantiateItem拿到对应的fragment相对应的引用

RankListViewFragment rlvf0 = (RankListViewFragment) mViewPagerRank.getAdapter().instantiateItem(view, index);

后台杀进程还有要注意的一个问题就是一些静态变量有可能会被销毁，所以要多留意使用静态变量或者在onsaveinstance里做储存

### 多服务器统一客户端问题

#### 功能描述：

用户只需要安装相同的客户端，输入账号密码登录后即可实现不同服务器的自动识别。

初次登陆的时候，首先对列表中的服务器进行一次遍历，并将能够使用的服务器记录下来，选择一个最优服务器并进行登陆。第二次（与以上）登陆的时候，首先判断上次选择的服务器是否可用，如果可用则继续使用。如果不可用，或者服务器发来需要重新遍历的命令，则重新遍历所有列表中的服务器。

#### 相关功能：

忘记密码（密码找回）

人员搜索

组织搜索

登录、登出

自动更新升级

#### 实现方法：

<http://phr.cmri.cn/app/serverlist.json>

保存UTF-8 格式数据如下，包含服务器地址，对应的人员表和组织表。

{"status":"SUCCESS","datavalue":[{"serverName":"phr.cmri.cn\/sport\/","contactfile":"contact\_cmri.json","groupfile":"group\_cmri.json"},{"serverName":"218.206.179.71\/jk\/","contactfile":"contact\_jk.json","groupfile":"group\_jk.json"},{"serverName":"mhealth.cmri.cn\/sport\/","contactfile":"contact\_mhealthsport.json","groupfile":"group\_mhealthsport.json"},{"serverName":"mhealth.cmri.cn\/zdgs\/","contactfile":"contact\_zdgs.json","groupfile":"group\_zdgs.json"}]}

利用loginAuth函数判断是否登录成功。

返回登陆成功，则确认该服务器为用户所在服务器

返回登陆失败，表示该服务器不存在此账号，去下一个服务器继续尝试，如果都失败，则提示用户名错误或密码错误

登录成功后，获取对应服务器上的人员列表和组织列表，以便支持搜索；

=================

注意

=================

自动登录勾选后，升级时，需要保存服务器的记录；

#### 分享：

分享包括友盟的集成，集成了QQ空间、腾讯微博、新浪微博、豆瓣等等。还有一个微信分享，微信包括微信好友分享和朋友圈分享。

友盟分享：在各大分享平台注册应用拿到appkey在友盟集成平台做一个填写，然后在Mainfest里面写必要的一些代码，就直接可以用条用集成分享了。当然还需要把一些文件拷贝进项目中，文件还是比较多的，如果喜欢比较简洁的工程还是不建议引用友盟的集成。

微信分享：微信分享比较麻烦一点，需要注册微信，注册应用，注册应用包括应用的相关网站，应用的哈希码(记得应该是哈希码，生成哈希码需要微信提供的一个手机应用输入自己应用的包名然后生产的一串代码)，当然可以申请两个appkey，一个调试版一个发布版，一个便于调试，一个正式发布用。

微信分享详细方法实现：

//创建微信实例，

IWXAPI mWeiXinAPI = WXAPIFactory.createWXAPI(mActivity, Constants.APP\_ID, true);

//注册微信ID

mWeiXinAPI.registerApp(Constants.APP\_ID);

//是否安装了微信的判断

Boolean b = mWeiXinAPI.isWXAppInstalled()

//api版本是否支持

Boolean b = mWeiXinAPI.isWXAppSupportAPI()

// 这个是微信的图片分享，也就是纯属的图片做分享

WXImageObject imageObject = new WXImageObject();

//需要将图片转换成byte字节

imageObject.imageData = byte{};

当然还可以imagepath指定本地图片，imageUrl网络图片

WXTextObject 这个是微信的文本分享，大概使用方法相同，还有视频、音乐、文件和网页分享

// 对分享信息做一层打包

WXMediaMessage msg = new WXMediaMessage();

msg.mediaObject = imageObject;

分享信息打包好然后就是发送给微信了

// 构造一个Req

SendMessageToWX.Req req = **new** SendMessageToWX.Req();

// transaction字段用于唯一标识一个请求

req.transaction = buildTransaction("text");

req.message = msg;

//指定分享给好友还是朋友圈

req.scene = SendMessageToWX.Req.*WXSceneSession*

SendMessageToWX.Req.*WXSceneTimeline*

//最后

// 调用api接口发送数据到微信

mWeiXinAPI.sendReq(req);