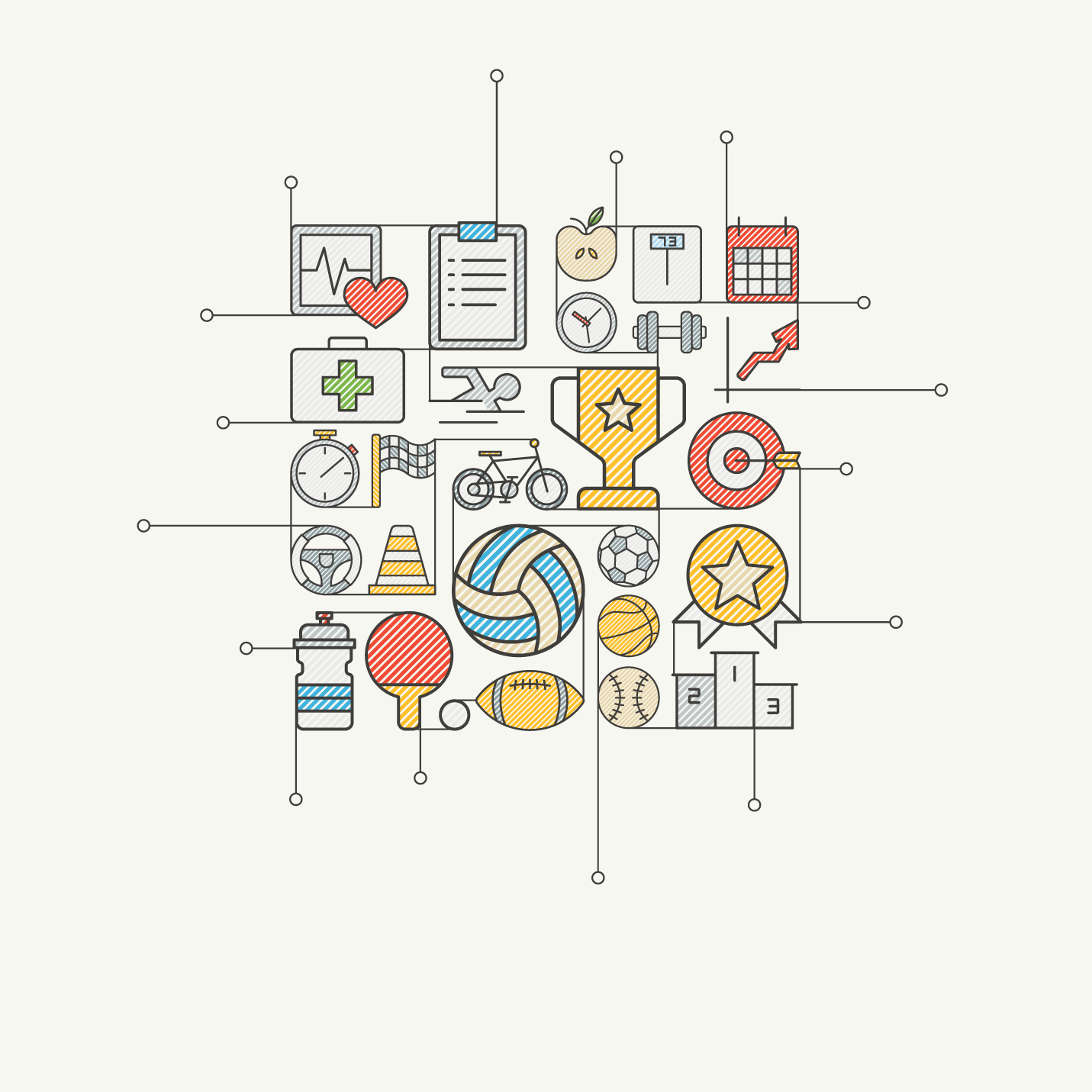
设计文档

“SweatGo”运动社交平台



作者：阙俊杰 141250102

目录

1 文档介绍 1

1.1 文档目的 1

1.2 读者对象 1

1.3 参考文档 1

2产品概要说明 1

2.1 系统用例图 1

2.4 主要页面高保真原型图 3

3体系结构设计 6

3.1 系统概述 6

3.2 体系结构模型设计 7

3.3 数据库结构设计 10

3.4 接口设计 17

主页 /index 17

4界面原型与人机交互设计 17

4.1 前端框架说明 17

4.2 UI规范设计 18

4.3 前端编码规范设计 18

4.4 页面交互逻辑设计 18

5 WEB性能优化设计 18

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 阙俊杰 | 2016.11.02 | 起草设计文档 | V1.0 |
| 阙俊杰 | 2016.11.05 | 完成初步整体架构设计 | V1.1 |

# 1 文档介绍

## 文档目的

SweatGo运动社交平台是一个基于WEB平台的应用，具有运动数据管理，与好友竞赛等功能。通过这个平台，用户能够监管自己的运动与身体基本状况，也能够在结交到同样爱好运动的朋友的同时，激发自己的运动积极性。

本文档给出了整个系统的具体系统架构以及接口定义，直接指导开发的进行。

## 读者对象

该平台的开发人员，测试人员以及产品最终验收人

## 参考文档

《SweatGo运动社交平台需求文档》

# 2产品概要说明

## 2.1 系统用例图

根据需求文档的需求描述，构建系统用例图如下：

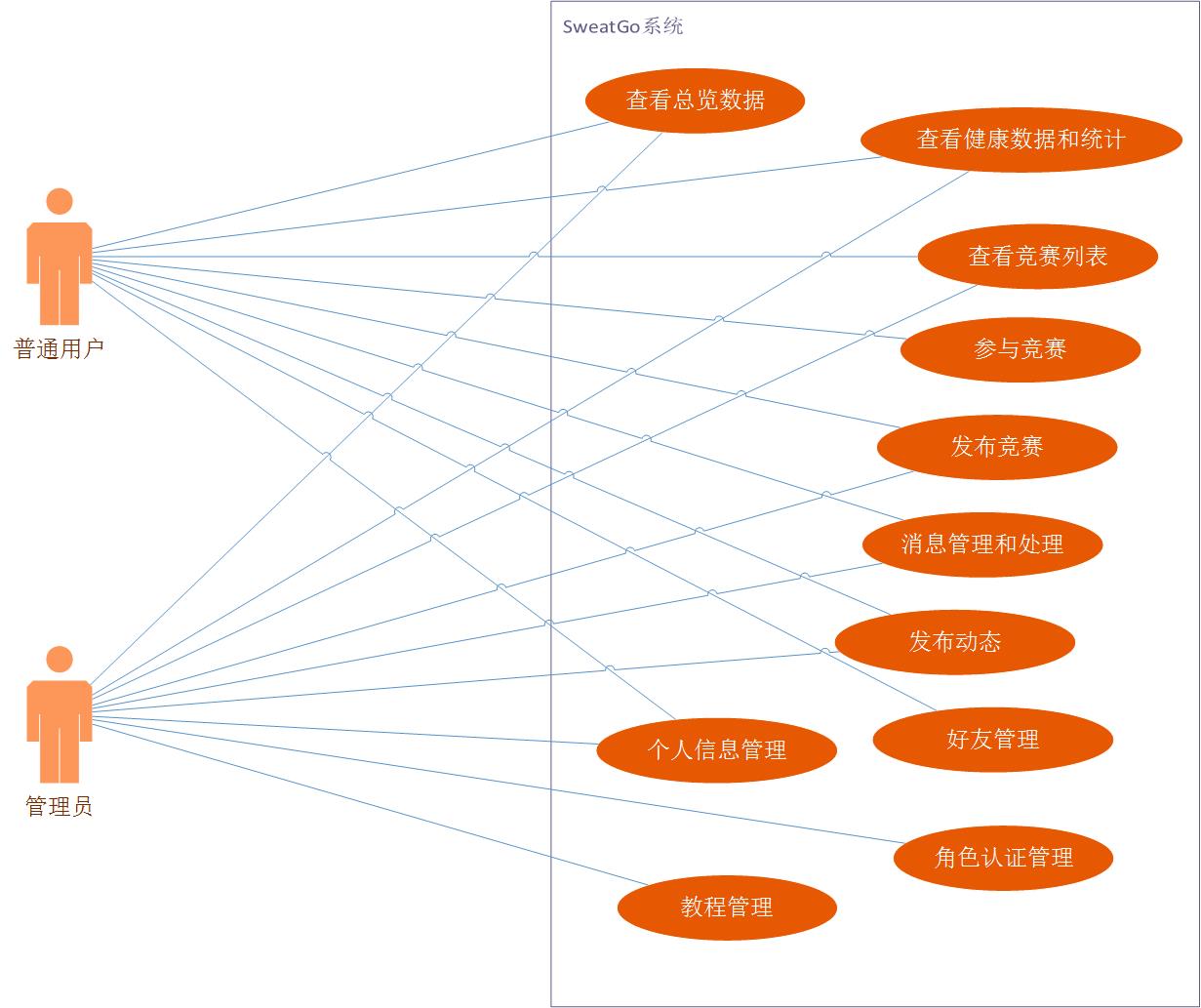


图 1

对应的系统用例描述表如下：

表格 1 新系统用例描述表

|  |  |
| --- | --- |
| 参与者 | 用例 |
| 普通用户 | 查看总览数据  查看健康数据和统计  查看竞赛列表  参与竞赛  查看建议  发布竞赛  消息管理和处理  发布动态  好友管理  个人信息管理 |
| 管理员 | 查看总览数据  查看健康数据和统计  查看竞赛列表  查看建议  发布竞赛  消息管理和处理  发布动态  个人信息管理  **角色认证管理**  **教程管理** |

## 2.4 主要页面高保真原型图

2.4.1 首页



图 2 首页图

2.4.2 健康相关页面

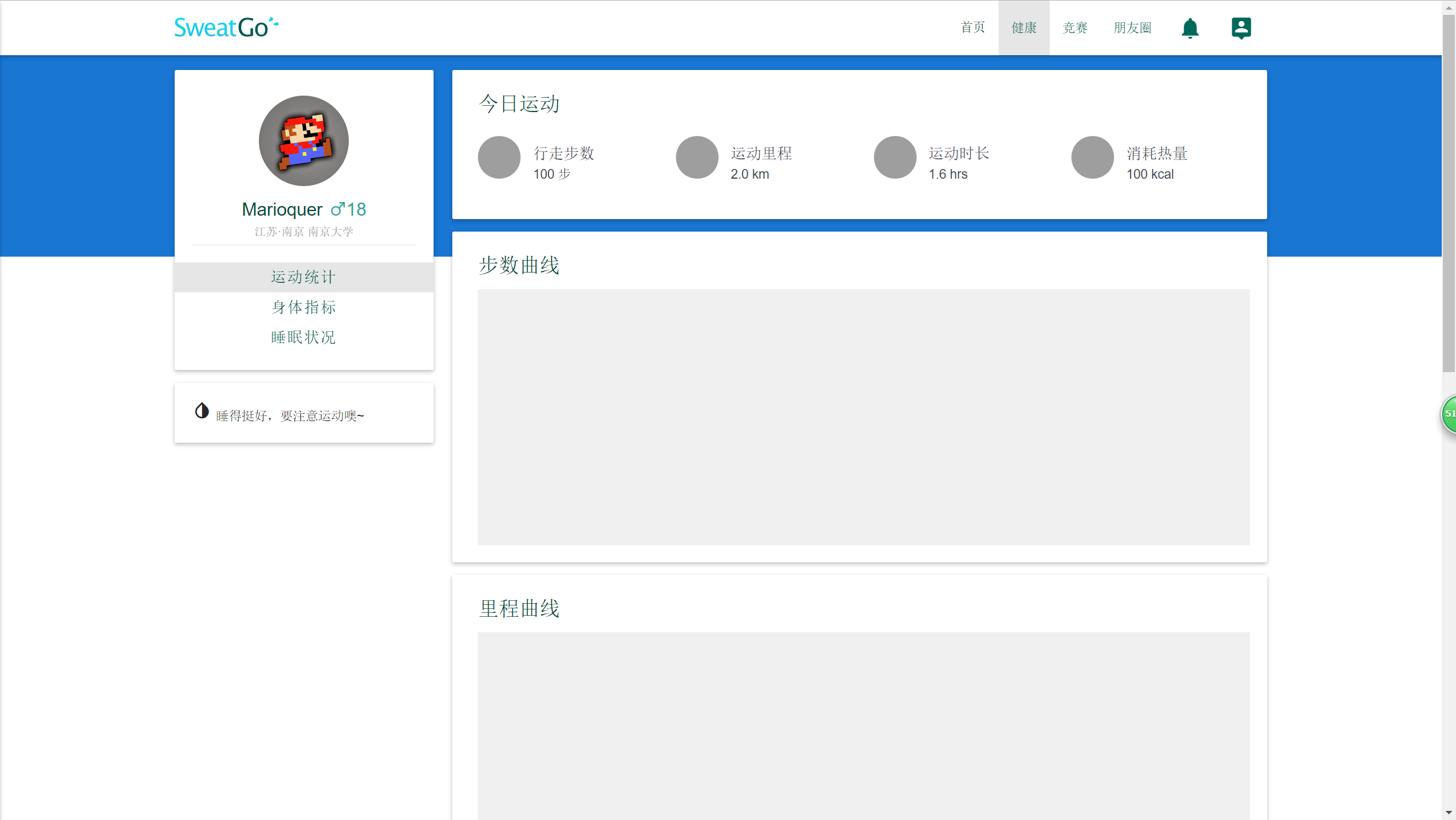


图 3

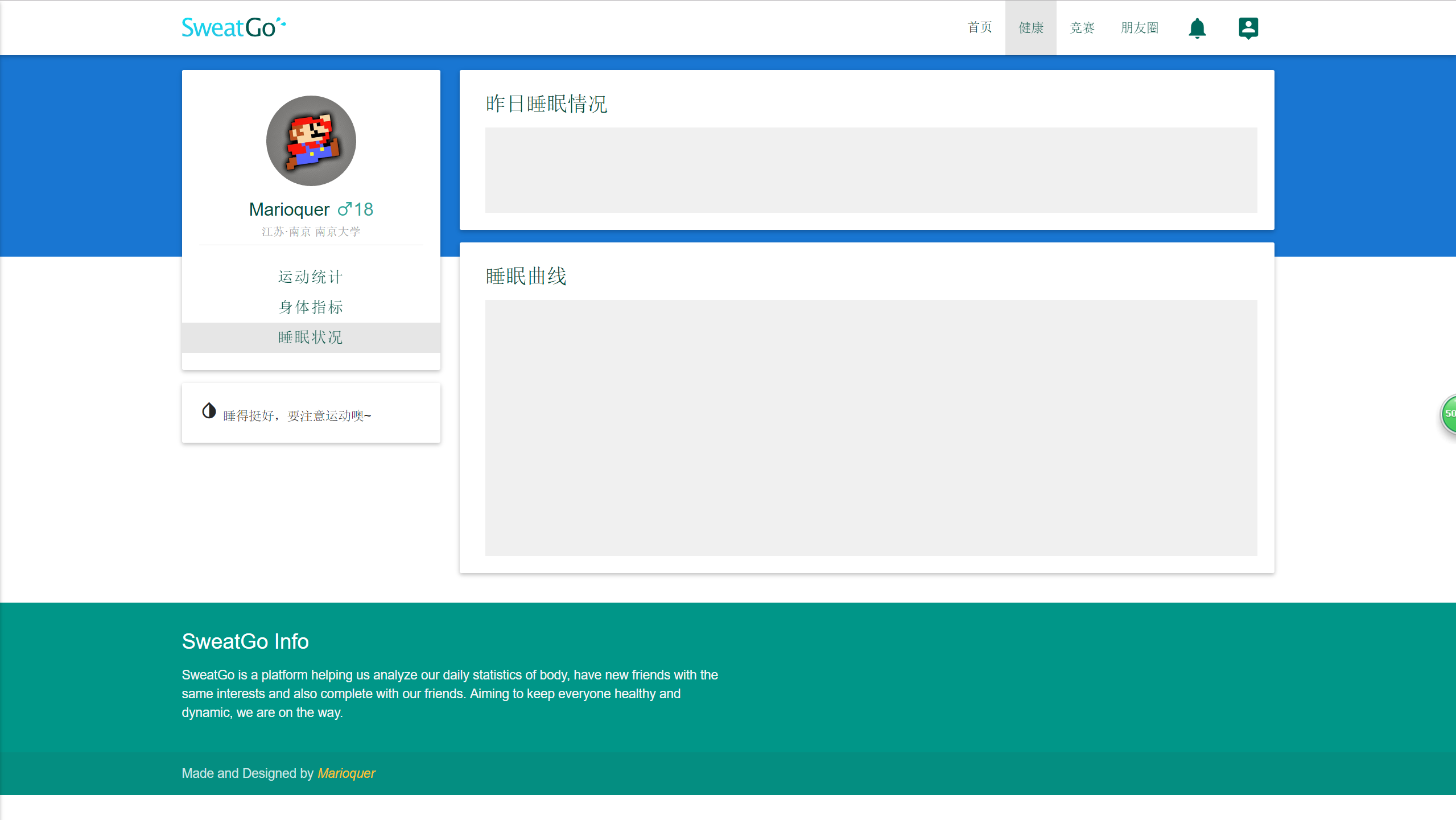


图 4 健康页面图

2.4.3 竞赛相关页面

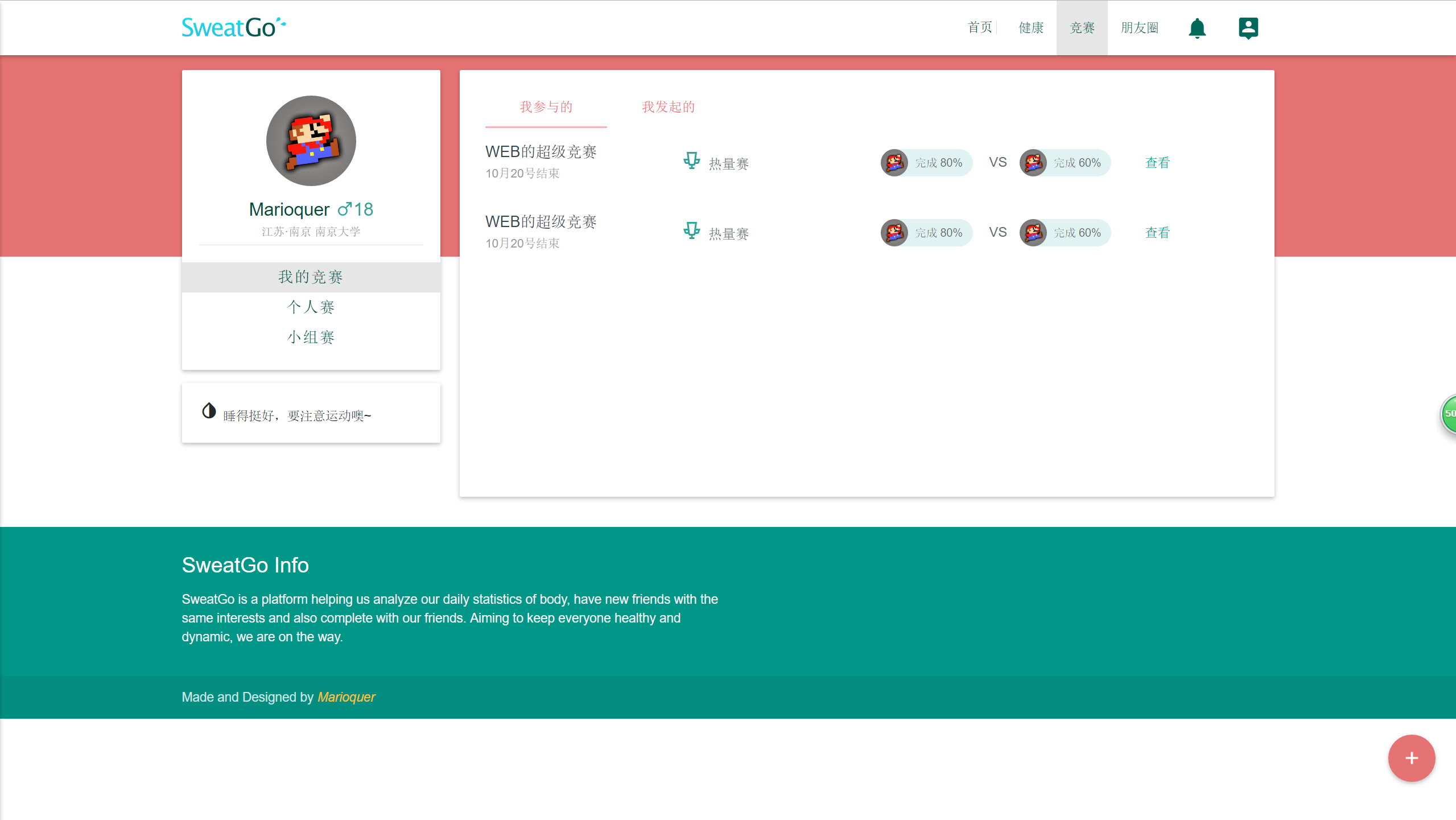


图 5 竞赛页面图

2.4.4 朋友圈相关页面

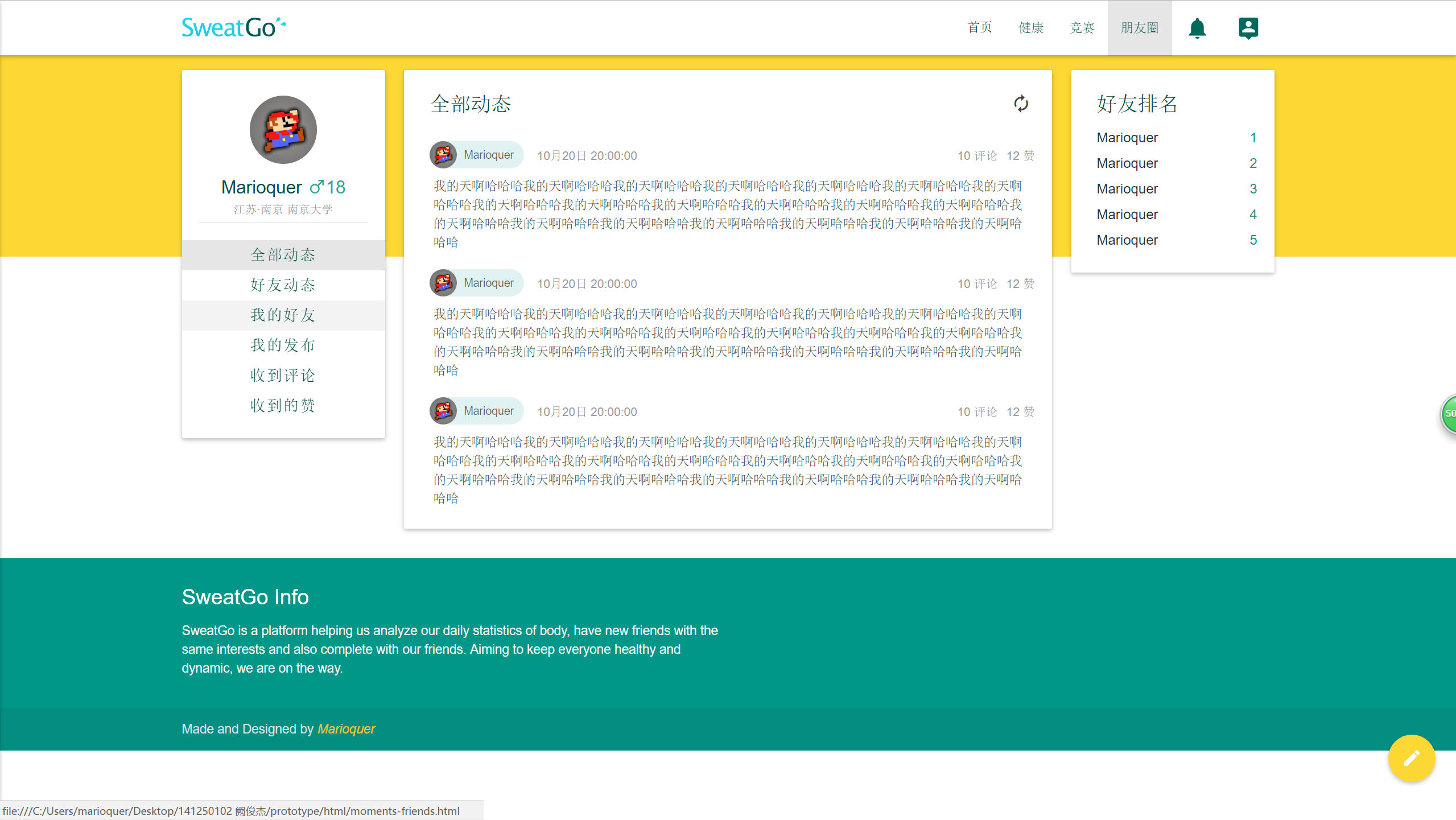


图 6 朋友圈页面图

# 3体系结构设计

## 3.1 系统概述

SweatGo运动平台是一个集社交和教学的健康管理软件，总体架构为前后端分离的MVC架构。前端使用Materialize 前端框架，设计风格符合Material Design设计语言，后端使用Laravel 5.1 LTS 框架配合SQLite3 开发。

**Laravel开发框架：**

表格 2 目录结构

|  |  |
| --- | --- |
| 目录 | 描述 |
| app | 包含了应用的核心代码，如Controller、Model等 |
| bootstrap | 包含了少许文件用于框架的启动和自动载入配置，还有一个cache文件夹用于包含框架生成的启动文件以提高性能 |
| config | 包含了应用所有的配置文件 |
| database | 包含了数据迁移及填充文件，以及SQLite数据库存储文件 |
| public | 包含了Laravel框架核心的引导文件index.php，以及任何可以公开的静态资源，如css，Javascript，images等 |
| resources | 包含了视图文件（views）及原生资源文件（LESS、SASS、CoffeeScript），以及本地化文件 |
| storage | 包含了编译过的Blade模板、基于文件的session、文件缓存，以及其它由框架生成的文件 |
| tests | 包含自动化测试 |
| vender | 包含Composer依赖，包含了Laravel源代码和第三方依赖包 |

其中app文件夹包含了实现的核心代码

表格 3 APP目录结构

|  |  |
| --- | --- |
| 目录/文件 | 描述 |
| /app/Console | 提供进入应用核心的API，包含了所有的Artisan命令 |
| /app/Events | 是放置事件类的地方，事件可以用于通知应用其它部分给定的动作已经发生，并提供灵活的解耦的处理 |
| /app/Exceptions | 包含应用的异常处理器，同时还是处理应用抛出的任何异常的好地方 |
| /app/Http | 提供进入应用核心的API，包含了控制器、过滤器和请求等 |
| /app/Http/Controllers | 包含了用于提供基本的逻辑、数据模型交互以及加载应用程序的视图文件的控制器类 |
| /app/Http/Middleware | 包含了HTTP中间件，对HTTP请求进行过滤 |
| /app/Http/Requests | 包含了对核心类Request的扩展 |
| /app/Http/routes.php | 这是应用的路由文件，其中包含路由规则，告诉Laravel如何将传入的请求连接到路由处理的闭包函数、控制器和操作 |
| /app/Jobs | 是放置队列任务的地方，应用中的任务可以被队列化，也可以在当前请求生命周期内同步执行 |
| /app/Listeners | 包含事件的处理器类，处理器接收一个事件并提供对该事件发生后的响应逻辑 |
| /app/Providers | 包含了服务提供者 |
| /app/\*.php | 在app目录下存储模型（Model），模型是代表应用程序的信息（数据）和操作数据的规则的一些类 |

MVC模型中的一次具体流程如下描述：

客户端向服务器发送相关请求，通过路由机制映射到相应的controller以及controller下对应的方法，该方法调用Model里的数据返回数据列表或者是视图给客户端，客户端再将相应的view视图进行渲染。

## 3.2 体系结构模型设计

根据需求定义，根据MVC开发架构将系统模块进一步细化，分为前台和后台管理两个部分，**前台**面对普通用户，以及角色认证过后的相关用户，以及管理员使用，**后台**面对管理员使用，特别要说明的是，管理员的角色相当于内容生产平台的编辑，既包含几乎普通用户的所有功能，另外添加了一些管理的功能

前台模块划分如下：

表格 4 前台系统模块

|  |  |  |
| --- | --- | --- |
| 模块 | 模块名称 | 涉及的用户对象 |
| Health | 健康管理 | 普通用户、管理员 |
| Advice | 建议管理 | 普通用户、管理员 |
| Competitions | 竞赛管理 | 普通用户、管理员 |
| Moments | 朋友圈 | 普通用户、管理员 |
| Messages | 消息管理 | 普通用户、管理员 |
| Personal | 个人信息管理 | 普通用户、管理员 |
| Friends | 好友管理 | 普通用户、管理员 |
| Auth | 用户认证 | 普通用户、管理员 |
| Index | 默认模块 | 普通用户、管理员 |

后台模块划分如下：

表格 5 后台系统模块

|  |  |  |
| --- | --- | --- |
| 模块 | 模块名称 | 涉及的用户对象 |
| Figure | 角色管理 | 管理员 |
| Lessons | 教程管理 | 管理员 |
| Auth | 用户认证 | 管理员 |

开发包图结构：

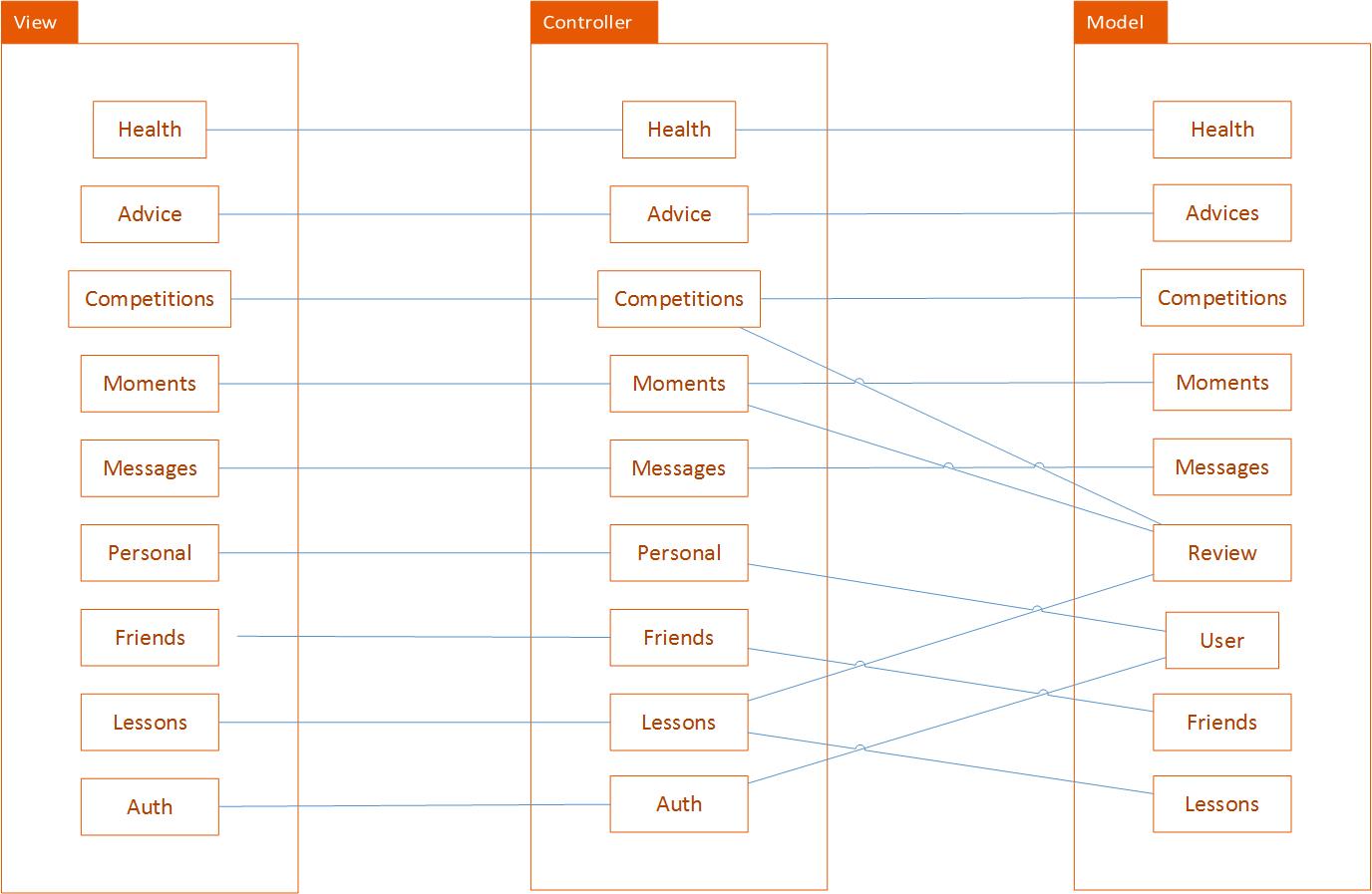


图 7 前台开发包图

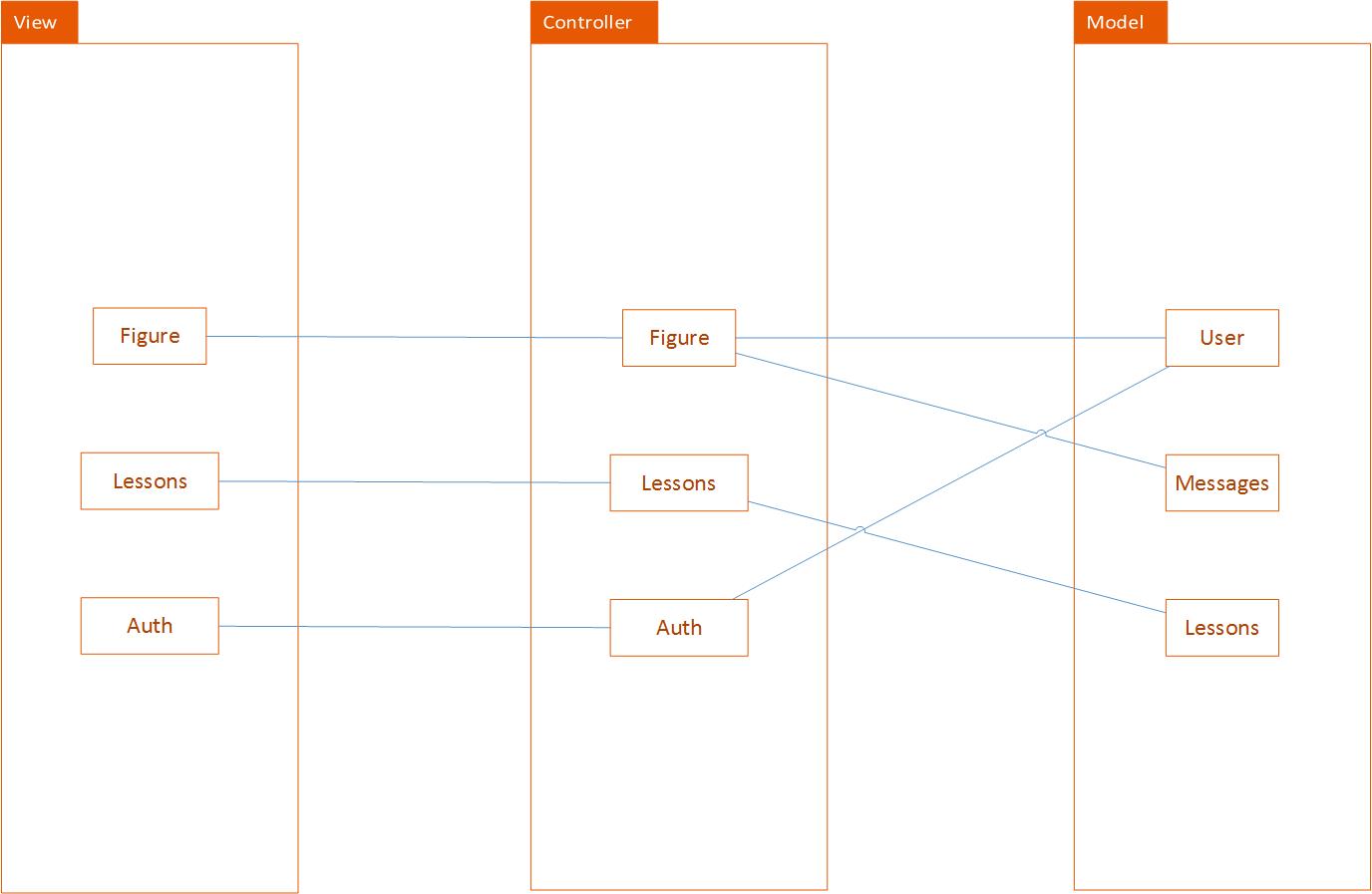


图 8 后台开发包图

## 3.3 数据库结构设计

标注为INC的为自增字段，FK为外键，高亮为黄色的为主键，字体颜色为红色的为外键

3.3.1 认证模块

表格 6 user表（存放用户的所有信息及角色认证结果）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED,INC | 用户ID |
| name | VARCHAR,UNIQUE | 用户名 |
| sex | TINYINT | 性别 |
| province | VARCHAR | 省份 |
| city | VARCHAR | 城市 |
| office | VARCHAR | 学校/单位 |
| role\_id | BIGINT, UNSIGNED,  FK(role\_type - id) | 角色，默认为普通用户，通过认证可成为医生/教练...(类似微博加V) |
| birthday | DATETIME | 生日 |
| avatar\_url | VARCHAR | 头像url |
| introduction | TEXT | 个人简介 |

表格 7 role\_type表（存放角色的类型）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED,INC | 角色ID |
| name | VARCHAR | 角色名称 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 8 role\_apply表（存放角色认证的申请）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED,INC | 角色申请ID |
| applier\_id | BIGINT, UNSIGNED,  FK(user- id) | 申请者ID |
| target\_role | ENUM | 目标申请角色 |
| content | TEXT | 申请内容 |
| image\_url | VARCHAR | 相关证明照片url |
| status | ENUM | 申请状态 |
| createdAt | DATETIME | 申请时间 |
| updatedAt | DATETIME | 状态更新时间 |

3.3.2 健康管理模块

表格 9 Advices表（存放根据健康数据分析提供的简单建议）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED,INC | 建议ID |
| user\_id | BIGINT, UNSIGNED,  FK(user- id) | 接收者ID |
| content | TEXT | 建议内容 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 10 sleep\_data表（存放睡眠数据）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 睡眠记录ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 用户ID |
| startedAt | TIME | 睡眠开始时间 |
| endedAt | TIME | 睡眠结束时间 |
| duration | INT | 睡眠时长 |
| deep\_sleep | INT | 深睡眠时长 |
| wake\_up\_times | INT | 醒来次数 |
| createdAt | DATETIME | 记录创建时间 |
| updatedAt | DATETIME | 记录更新时间 |

表格 11 body\_data表（存放身体数据）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 数据ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 用户ID |
| height | DOUBLE | 身高 |
| weight | DOUBLE | 体重 |
| heart\_rate | INT | 心率 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 12 exercise\_data表（存放运动数据）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 数据ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 用户ID |
| pace | INT | 步数 |
| mileage | DOUBLE | 里程 |
| time | DOUBLE | 时长 |
| heat | DOUBLE | 热量 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

3.3.3 竞赛管理模块

表格 13 competitions\_info表（存放竞赛的基本数据）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 竞赛ID |
| author\_id | BIGINT, UNSIGNED,  FK(user - id) | 作者ID |
| winner\_id | BIGINT, UNSIGNED,  FK(user - id) | 获胜者ID |
| introduction | TEXT | 竞赛简介 |
| end\_time | DATETIME | 竞赛结束时间 |
| class | ENUM | 个人赛/组合赛 |
| type | ENUM | 竞赛类别 |
| target | DOUBLE | 竞赛目标值 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 14 competitions\_process关系表（存放竞赛的参与人及进度）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| competition\_id | BIGINT, UNSIGNED,  FK(competitions\_info - id) | 竞赛ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 参与者ID |
| data | DOUBLE | 目前数据 |
| percent | DOUBLE | 完成百分比 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 15 competitions\_apply关系表（存放竞赛申请）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| applier\_id | BIGINT, UNSIGNED,  FK(user - id) | 申请者ID |
| competition\_id | BIGINT, UNSIGNED,  FK(competitions\_info - id) | 竞赛ID |
| status | ENUM | 申请状态 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 16 competition\_comments表（存放所有的评论记录）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 竞赛评论ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 评论者ID |
| to\_user\_id | BIGINT, UNSIGNED,  FK(user - id) | 被评论者ID |
| competition\_id | BIGINT, UNSIGNED,  FK(competitions\_info - id) | 竞赛ID |
| content | TEXT | 评论内容 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 17 competiton\_likes表（存放所有的点赞记录）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 竞赛点赞ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 点赞者ID |
| competition\_id | BIGINT, UNSIGNED,  FK(competitions\_info - id) | 竞赛ID |
| like | TINYINT | 是否喜欢：1喜欢 0不喜欢 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

3.3.4 好友管理模块

表格 18 friends关系表（存放朋友关系）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| me\_id | BIGINT, UNSIGNED,  FK(user - id) | 我的ID |
| you\_id | BIGINT, UNSIGNED,  FK(user - id) | 朋友ID |
| status | ENUM | 朋友状态 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 19 friend\_apply关系表（存放朋友申请）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| sender\_id | BIGINT, UNSIGNED,  FK(user - id) | 申请者ID |
| receiver\_id | BIGINT, UNSIGNED,  FK(user - id) | 被申请者ID |
| hello | TEXT | 招呼 |
| status | ENUM | 申请状态 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

3.3.5 朋友圈模块

表格 20 moments表（存放朋友圈动态）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 朋友圈动态ID |
| author\_id | BIGINT, UNSIGNED,  FK(user - id) | 作者ID |
| image\_url | VARCHAR | 照片URL |
| content | TEXT | 内容 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 21 moment\_comments表（存放所有的评论记录）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 朋友圈评论ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 评论者ID |
| to\_user\_id | BIGINT, UNSIGNED,  FK(user - id) | 被评论者 |
| moment\_id | BIGINT, UNSIGNED,  FK(moments - id) | 朋友圈动态ID |
| content | TEXT | 评论内容 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 22 moment\_likes表（存放所有的点赞记录）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 朋友圈点赞ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 点赞者ID |
| moment\_id | BIGINT, UNSIGNED,  FK(moments - id) | 朋友圈动态ID |
| like | TINYINT | 是否喜欢 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

3.3.6消息模块

表格 23 info\_messages表（存放通知型消息）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 消息ID |
| sender\_id | BIGINT, UNSIGNED,  FK(user - id) | 发送者ID |
| receiver\_id | BIGINT, UNSIGNED,  FK(user - id) | 接收者ID |
| content | TEXT | 内容 |
| isRead | TINYINT | 是否阅读 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

好友申请以及竞赛申请相关的表也同样属于该模块

3.3.7教程模块

表格 24 lessons表（存放所有的教程）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 教程ID |
| author\_id | BIGINT, UNSIGNED,  FK(user - id) | 创建者ID |
| content | TEXT | 内容 |
| video | CHAR | 视频url |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 25 lesson\_comments表（存放所有的评论记录）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 教程评论ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 评论者ID |
| to\_user\_id | BIGINT, UNSIGNED,  FK(user - id) | 被评论者 |
| lesson\_id | BIGINT, UNSIGNED,  FK(lessons - id) | 教程ID |
| content | TEXT | 内容 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

表格 26 lesson\_likes表（存放所有的点赞记录）

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, INC | 教程点赞ID |
| user\_id | BIGINT, UNSIGNED,  FK(user - id) | 点赞者ID |
| lesson\_id | BIGINT, UNSIGNED,  FK(lessons - id) | 教程ID |
| like | TINYINT | 是否喜欢 |
| createdAt | DATETIME | 创建时间 |
| updatedAt | DATETIME | 更新时间 |

## 3.4 接口设计

### 健康-运动统计 /health/exercise

|  |  |
| --- | --- |
| 方法名称 | 获得今天运动数据 |
| Url | /getToday |
| Method | GET |
| 参数列表 |  |
| 返回值说明 | {  “exercise”:{4000, 2.0, 1.6, 150.5} //步数、里程、时长、热量  } |

|  |  |
| --- | --- |
| 方法名称 | 获得所有运动数据 |
| Url | /getAll |
| Method | POST |
| 参数列表 |  |
| 返回值说明 | {  “pace”:{4000,4000,.....} //步数、里程、时长、热量  “mileage”:{2.0,2.0,......}  “time”:{2.0,2.0,......}  “heat”:{106,107,......}  } |

|  |  |
| --- | --- |
| 方法名称 | 添加今天的运动数据 |
| Url | /exercise/addToday |
| Method | POST |
| 参数列表 | {  “exercise”: {4000, 2.0, 1.6, 150.5}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否插入成功  } |

|  |  |
| --- | --- |
| 方法名称 | 添加多条运动数据 |
| Url | /addMany |
| Method | POST |
| 参数列表 | {  “exercises”: {{4000, 2.0, 1.6, 150.5}，  {4000, 2.0, 1.6, 150.5}，  ......} //批量导入数据  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否插入成功  } |

### 健康-身体指标/health/body

|  |  |
| --- | --- |
| 方法名称 | 获取最近的身体数据 |
| Url | /getRecent |
| Method | GET |
| 参数列表 |  |
| 返回值说明 | {  “body”: {1.8, 50.0, 104} //身高，体重，心率  } |

|  |  |
| --- | --- |
| 方法名称 | 获取全部的身体数据 |
| Url | /getAll |
| Method | POST |
| 参数列表 |  |
| 返回值说明 | {  “height”: {1.8, 1.8, 1.8,.......}, //身高，体重，心率  “weight”: {150.0,90.0,80.0,......},  “heartRate”:{103,103,103,......}  } |

|  |  |
| --- | --- |
| 方法名称 | 插入一条身体数据 |
| Url | /addToday |
| Method | POST |
| 参数列表 | {  “body”: {1.8, 50.0, 104}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否插入成功  } |

|  |  |
| --- | --- |
| 方法名称 | 插入多条身体数据 |
| Url | /addMany |
| Method | POST |
| 参数列表 | {  “bodies”:{ {1.8, 50.0, 104},  {1.8, 50.0, 104},  {1.8, 50.0, 104},.......}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否插入成功  } |

### 健康-睡眠指标/health/sleep

|  |  |
| --- | --- |
| 方法名称 | 获得今天睡眠数据 |
| Url | /getToday |
| Method | GET |
| 参数列表 |  |
| 返回值说明 | {  “sleep”:{“2016-07-08 16:00:00”,  “2016-07-08 16:00:00”,  7.0,4.0,4}  //睡觉时间、醒来时间、睡眠时长、深睡眠时长、醒来次数  } |

|  |  |
| --- | --- |
| 方法名称 | 获得全部睡眠数据 |
| Url | /getAll |
| Method | POST |
| 参数列表 |  |
| 返回值说明 | {  “sleeps”:{{“2016-07-08 16:00:00”,  “2016-07-08 16:00:00”,  7.0,4.0,4},......}  //睡觉时间、醒来时间、睡眠时长、深睡眠时长、醒来次数  } |

|  |  |
| --- | --- |
| 方法名称 | 插入一条睡眠数据 |
| Url | /addToday |
| Method | POST |
| 参数列表 | {  “sleep”:{“2016-07-08 16:00:00”,  “2016-07-08 16:00:00”,  7.0,4.0,4}  //睡觉时间、醒来时间、睡眠时长、深睡眠时长、醒来次数  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否插入成功  } |

|  |  |
| --- | --- |
| 方法名称 | 插入多条睡眠数据 |
| Url | /addMany |
| Method | POST |
| 参数列表 | {  “sleeps”:{{“2016-07-08 16:00:00”,  “2016-07-08 16:00:00”,  7.0,4.0,4},......}  //睡觉时间、醒来时间、睡眠时长、深睡眠时长、醒来次数  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否插入成功  } |

### 竞赛-我的竞赛/competitions/mine

|  |  |
| --- | --- |
| 方法名称 | 获取我参与的个人赛 |
| Url | /getJoinedIndividual |
| Method | POST |
| 参数列表 | {  “id”: {14} //用户id  } |
| 返回值说明 | {  “individual”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “individual”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ， 发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |

|  |  |
| --- | --- |
| 方法名称 | 获取我参与的小组赛 |
| Url | /getJoinedGroup |
| Method | POST |
| 参数列表 | {  “id”: {14} //用户id  } |
| 返回值说明 | {  “group”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “group”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ，发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |

|  |  |
| --- | --- |
| 方法名称 | 获取我发起的个人赛 |
| Url | /getMyIndividual |
| Method | POST |
| 参数列表 | {  “id”: {14} //用户id  } |
| 返回值说明 | {  “individual”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “group”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ，发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |

|  |  |
| --- | --- |
| 方法名称 | 获取我发起的小组赛 |
| Url | /getMyGroup |
| Method | POST |
| 参数列表 | {  “id”: {14} //用户id  } |
| 返回值说明 | {  “group”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “group”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ，发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |

### 竞赛-个人赛/competitions/individual

|  |  |
| --- | --- |
| 方法名称 | 获取所有个人赛 |
| Url | /getAll |
| Method | POST |
| 参数列表 |  |
| 返回值说明 | {  “individual”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “group”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ，发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |

### 竞赛-小组赛/competitions/group

|  |  |
| --- | --- |
| 方法名称 | 获取所有小组赛 |
| Url | /getAll |
| Method | POST |
| 参数列表 |  |
| 返回值说明 | {  “group”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “group”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ，发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |

### 竞赛-加入比赛

|  |  |
| --- | --- |
| 方法名称 | 获取所有小组赛 |
| Url | /join |
| Method | POST |
| 参数列表 | {  “id”: {14} //竞赛id  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发出申请成功  } |

### 竞赛-新建比赛

|  |  |
| --- | --- |
| 方法名称 | 获取所有小组赛 |
| Url | /publish |
| Method | POST |
| 参数列表 | {  “competition”: {1，12,31,”啊啊啊”,”2016-11-08 23:00:00”, “group”, “heat” , 200.0, ”2016-11-08 23:00:00”，list}  //比赛id ，发布者id，获胜者id，简介，结束时间，个人赛/小组赛，目标，创建时间, 参与者数据的list  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发出申请成功  } |

### 朋友圈-动态/moments

|  |  |
| --- | --- |
| 方法名称 | 获取所有动态 |
| Url | /getAll |
| Method | POST |
| 参数列表 |  |
| 返回值说明 | {  “moments”: {{user, time, content, url, comments, likes，role\_id},......} //所有动态列表  } |

|  |  |
| --- | --- |
| 方法名称 | 获取更新动态 |
| Url | /getNew |
| Method | POST |
| 参数列表 | {  “latestUpdate”:{2016-09-09 16:00:00} //最后更新时间  } |
| 返回值说明 | {  “moments”: {{user, time, content, url, comments, likes, role\_id },......} //最新动态列表  } |

|  |  |
| --- | --- |
| 方法名称 | 获取好友动态 |
| Url | /getFriendsPost |
| Method | POST |
| 参数列表 | {  “id”:{20} //用户id  } |
| 返回值说明 | {  “moments”: {{user, time, content, url, comments, likes, role\_id },......} //好友动态列表  } |

|  |  |
| --- | --- |
| 方法名称 | 获取我的动态 |
| Url | /getMine |
| Method | POST |
| 参数列表 | {  “id”:{20} //用户id  } |
| 返回值说明 | {  “moments”: {{user, time, content, url, comments, likes, role\_id },......} //好友动态列表  } |

|  |  |
| --- | --- |
| 方法名称 | 发布我的动态 |
| Url | /postMine |
| Method | POST |
| 参数列表 | {  “moments”: {user, time, content, url, comments, likes,}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布成功  } |

|  |  |
| --- | --- |
| 方法名称 | 评论动态 |
| Url | /comment |
| Method | POST |
| 参数列表 | {  “comment”: {id, to\_id, moment\_id, content, time}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布成功  } |

|  |  |
| --- | --- |
| 方法名称 | 点赞/取消点赞动态 |
| Url | /like |
| Method | POST |
| 参数列表 | {  “like”: {id, moment\_id, likeOrNot}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布成功  } |

|  |  |
| --- | --- |
| 方法名称 | 收到评论 |
| Url | /receiveComments |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  } |
| 返回值说明 | {  “comments”: {{id, to\_id, moment\_id, content, time},...}  } |

|  |  |
| --- | --- |
| 方法名称 | 收到的赞 |
| Url | /receiveLikes |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  } |
| 返回值说明 | {  “likes”: {id, moment\_id, like}  } |

### 好友/friends

|  |  |
| --- | --- |
| 方法名称 | 我的好友 |
| Url | /getMine |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  } |
| 返回值说明 | {  “friends”: {{id, name, sex, age,province,city,office, avator\_url },...}  //好友列表  } |

|  |  |
| --- | --- |
| 方法名称 | 添加好友申请 |
| Url | /add |
| Method | POST |
| 参数列表 | {  “me\_id”: {id} //用户id  “you\_id”:{id} //目标id  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布申请成功  } |

|  |  |
| --- | --- |
| 方法名称 | 同意 |
| Url | /agree |
| Method | POST |
| 参数列表 | {  “me\_id”: {id} //用户id  “you\_id”:{id} //目标id  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布申请成功  } |

|  |  |
| --- | --- |
| 方法名称 | 忽略 |
| Url | /ignore |
| Method | POST |
| 参数列表 | {  “me\_id”: {id} //用户id  “you\_id”:{id} //目标id  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布申请成功  } |

|  |  |
| --- | --- |
| 方法名称 | 删除好友 |
| Url | /delete |
| Method | POST |
| 参数列表 | {  “me\_id”: {id} //用户id  “you\_id”:{id} //目标id  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否删除成功  } |

|  |  |
| --- | --- |
| 方法名称 | 获取好友运动排名 |
| Url | /getRank |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  } |
| 返回值说明 | {  “rank”:{id,name,avator\_url,rank} //排名  } |

### 用户/users

|  |  |
| --- | --- |
| 方法名称 | 查看用户信息 |
| Url | /getInfo |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  } |
| 返回值说明 | {  “user”: {id, name, sex, age,province,city,office, avator\_url, role\_id ,pace, mileage, time, heat } //标识是否删除成功  } |

|  |  |
| --- | --- |
| 方法名称 | 登陆 |
| Url | /login |
| Method | POST |
| 参数列表 | {  “userName”: {userName} //用户名  “passWord”:{passWord} //目标密码  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否登陆成功  } |

|  |  |
| --- | --- |
| 方法名称 | 登出 |
| Url | /logout |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否登陆成功  } |

|  |  |
| --- | --- |
| 方法名称 | 注册 |
| Url | /logup |
| Method | POST |
| 参数列表 | {  “userName”: {userName} //用户名  “passWord”:{passWord}, //目标密码  “sex”:{sex},  “birthday”:{birthday},  “province”:{province},  “city”:{city},  “office”:{office}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否登陆成功  } |

|  |  |
| --- | --- |
| 方法名称 | 申请角色 |
| Url | /applyRole |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  “role\_id”:{role\_id},  “description”:{...},  “img\_url”: {...}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布申请成功  } |

|  |  |
| --- | --- |
| 方法名称 | 通过角色 |
| Url | /agreeRole |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  “role\_id”:{role\_id}  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布申请成功  } |

|  |  |
| --- | --- |
| 方法名称 | 拒绝角色 |
| Url | /agreeRole |
| Method | POST |
| 参数列表 | {  “id”: {id} //用户id  “role\_id”:{role\_id}  “content”:{..} //拒绝理由  } |
| 返回值说明 | {  “SUCCEED”/”FAILED” //标识是否发布申请成功  } |

# 4界面原型与人机交互设计

## 4.1 前端框架说明

Materialize是一个基于Material Design设计语言的前端框架，提供了：

- CSS: 一整套的Material Design颜色板，阴影的层数，以及响应式布局的定义，以及Google图标的IconFont

- Components: 横幕、按钮、面包屑导航、翻页......

- JS：侧边栏，FloatButton的实现，WavesEffect的动效，以及诸多符合设计语言的微动效

本地化：

Materialize是一个很棒的前端框架，界面前卫而且流畅，但使用过程中依旧需要进行重设计以及本地化操作，具体适应项目如下：

- 所有的Logo，页面布局，以及页面布局等等UX、UI设计均自己完成

- Google的Icon存于本地，解决墙的问题

- 中文字体，为了适应设计语言，windows平台首选微软雅黑，而Mac平台首选苹方字体，而英文则使用框架自带的Roboto字体

- 栅栏增定义：对于原有的Grid定义，不能满足一些特定的需求，因此加入了中间值的Grid定义，使其更加好地适应项目

- 自定义CSS: 使用框架的好处就是能够学习到CSS定义上的一些技巧，于是自己在定义CSS的时候，也采用了单个频繁使用属性转化成类的做法，将一些常用css属性，提取成类名，使用起来较为方便。

# 5 WEB性能优化设计

“SweatGo”是一款关注个人健康的Web社交应用，使用群体是社会中任何对此有需求的个人用户群体，为此需提供良好的用户体验，因此需要进行Web性能优化，于是设计了一套Web性能优化的方案，具体如下。

1. **减少HTTP请求**

这主要针对前端，将多个CSS文件和JS文件分别合并成一个，从而减少HTTP请求。

1. **压缩CSS、JS文件**

将CSS、JS文件进行压缩，减小文件体积大小。

1. **使用压缩组件**

传输之前，服务端先使用GZIP压缩再传输给客户端，客户端接收之后由浏览器解压。

1. **SQL语句优化**

对于数据库的增删改查，要对SQL语句进行优化，以达到最高效率。

1. **使用Redis或者Memcached缓存**

使用Redis或者Memcached缓存以加快页面和数据访问的速度。