WEB设计文档

141250110 侍硕

目录

[1.引言 1](#_Toc466063196)

[1.1编制目的 1](#_Toc466063197)

[1.2参考材料 1](#_Toc466063198)

[2.产品描述 1](#_Toc466063199)

[3.体系结构 1](#_Toc466063200)

[4.界面设计 2](#_Toc466063201)

[5.人机交互设计 2](#_Toc466063202)

[6.详细设计 2](#_Toc466063203)

[6.1登录/注册 2](#_Toc466063204)

[6.1.1提供的接口设计 2](#_Toc466063205)

[6.1.2需要的接口 4](#_Toc466063206)

[6.2动态管理 5](#_Toc466063207)

[6.2.1提供的接口设计 5](#_Toc466063208)

[6.2.2需要的接口 7](#_Toc466063209)

[6.3竞赛管理 7](#_Toc466063210)

[6.3.1提供的接口设计 7](#_Toc466063211)

[6.3.2需要的接口 10](#_Toc466063212)

[6.4用户管理 11](#_Toc466063213)

[6.4.1提供的接口设计 11](#_Toc466063214)

[6.4.2需要的接口 17](#_Toc466063215)

[7.数据格式定义 17](#_Toc466063216)

[7.1用户运动记录 17](#_Toc466063217)

[7.2用户运动总况 18](#_Toc466063218)

[7.3用户运动距离和时长 18](#_Toc466063219)

[7.4用户运动强度 19](#_Toc466063220)

# 1.引言

## 1.1编制目的

本文档详细完成对运动社交网站的概要设计，作为指导开发的目的参考资料

## 1.2参考材料

《Web程序设计 第8版》

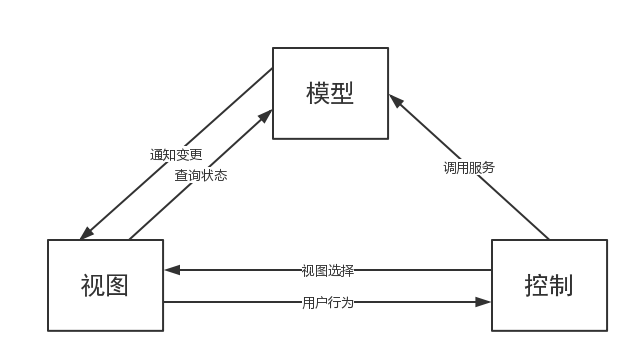
《软件开发的技术基础》

# 2.产品描述

参考本运动社交网站需求文档对产品的概述描述

# 3.体系结构

运动社交网站使用的是Laravel框架的MVC架构，View采用blade模板引擎，Model采用了Eloquent ORM来进行数据库操作，数据库采用SQLite。三层的交互如下：



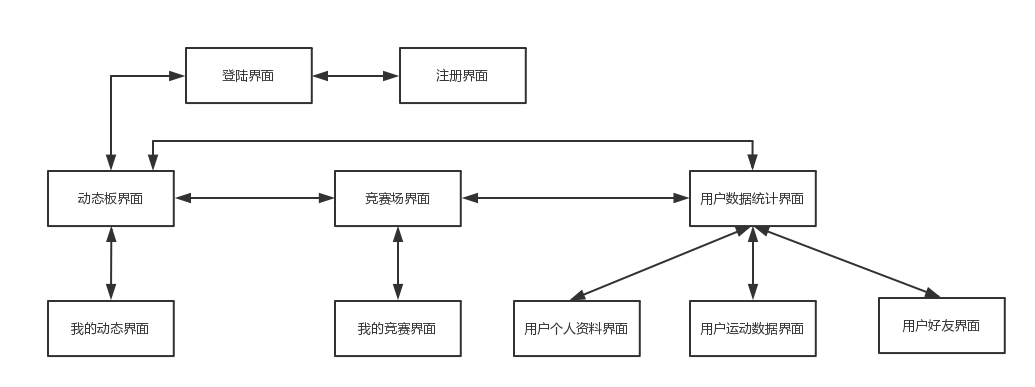
# 4.界面设计

请参考高保真模型，这里不再赘述

# 5.人机交互设计

一共有10个界面，包括登陆界面、注册界面、动态板、我的动态、竞赛场、我的竞赛、数据统计、个人资料、运动数据、我的朋友

界面转换如下：



# 6.详细设计

## 6.1登录/注册

### 6.1.1提供的接口设计

#### 6.1.1.1登录界面

|  |  |
| --- | --- |
| 登陆界面 | |
| url | /Login |
| 方法类型 | Get |
| 控制器 | LoginController |
| 方法 | getLogin() |
| 参数 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回登录界面 |
| 备注 |  |

#### 6.1.1.2登录请求

|  |  |
| --- | --- |
| 登陆界面 | |
| url | /Login |
| 方法类型 | Post |
| 控制器 | LoginController |
| 方法 | postLogin() |
| 参数 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 验证失败，返回登陆界面并显示失败提示；  验证成功返回动态板界面 |
| 备注 |  |

#### 6.1.1.3注册界面

|  |  |
| --- | --- |
| 登陆界面 | |
| url | /Register |
| 方法类型 | Get |
| 控制器 | RegisterController |
| 方法 | getRegister() |
| 参数 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册界面 |
| 备注 |  |

#### 6.1.1.4注册请求

|  |  |
| --- | --- |
| 登陆界面 | |
| url | /Register |
| 方法类型 | Post |
| 控制器 | RegisterController |
| 方法 | postRegister() |
| 参数 | 无 |
| 前置条件 | 用户未注册 |
| 后置条件 | 注册失败，返回注册界面并显示失败提示；  注册成功返回注册界面 |
| 备注 |  |

### 6.1.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| User::create | 创建用户并加入数据库 |
| User::find | 查找用户 |

## 6.2动态管理

### 6.2.1提供的接口设计

#### 6.2.1.1 动态板界面

|  |  |
| --- | --- |
| 动态版界面 | |
| url | /user/{userId}/related-moments |
| 方法类型 | Get |
| 控制器 | MomentController |
| 方法 | getRelatedMoments() |
| 参数 | 无 |
| 前置条件 | 用户已登陆 |
| 后置条件 | 返回动态板界面 |
| 备注 | 包含该用户及其好友的全部动态，按时间降序排列 |

#### 6.2.1.2用户动态界面

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{userId}/moments |
| 方法类型 | Get |
| 控制器 | MomentController |
| 方法 | getMyMoments ($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登陆 |
| 后置条件 | 返回该用户的动态界面 |
| 备注 | 包含该用户的全部动态，按时间降序排列 |

#### 6.2.1.3发布动态

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{userId}/related-moments |
| 方法类型 | POST |
| 控制器 | MomentController |
| 方法 | createMoment($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登陆 |
| 后置条件 | 保存新动态并返回该用户的动态界面 |
| 备注 | 包含该用户的全部动态，按时间降序排列 |

#### 6.2.1.4删除动态

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | user/{userId}/moments/{momentId} |
| 方法类型 | Delete |
| 控制器 | MomentController |
| 方法 | deleteMoments($momentId) |
| 参数 | $momentId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 删除动态并返回我的动态界面 |
| 备注 |  |

### 6.2.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Moment::find | 查看某个动态 |
| Moment::findAll | 查看某个用户的动态 |
| Moment::create | 新建动态 |
| Moment::delete | 删除动态 |

## 6.3竞赛管理

### 6.3.1提供的接口设计

#### 6.3.1.1竞赛场界面

|  |  |
| --- | --- |
| 动态版界面 | |
| url | /competitions |
| 方法类型 | Get |
| 控制器 | CompetitionController |
| 方法 | getAllCompetitions() |
| 参数 | 无 |
| 前置条件 | 用户已登陆 |
| 后置条件 | 返回竞赛场界面 |
| 备注 | 包含全部竞赛按时间降序排列 |

#### 6.3.1.2用户竞赛界面

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{userId}/ competitions |
| 方法类型 | Get |
| 控制器 | CompetitionController |
| 方法 | getCompetitionsByUserId($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登陆 |
| 后置条件 | 返回该用户的竞赛界面 |
| 备注 | 包含该用户的全部竞赛，按照进行中、已结束划分 |

#### 6.3.1.3发布竞赛

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{userId}/ competitions |
| 方法类型 | POST |
| 控制器 | CompetitionController |
| 方法 | createCompetition($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登陆 |
| 后置条件 | 保存新竞赛并返回竞赛场界面 |
| 备注 | 包含全部竞赛按时间降序排列 |

#### 6.3.1.4删除竞赛

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /competitions/{competitionId} |
| 方法类型 | Delete |
| 控制器 | CompetitionController |
| 方法 | deleteCompetition($competitionId) |
| 参数 | competitionId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 删除该竞赛并返回我的竞赛界面 |
| 备注 |  |

#### 6.3.1.5加入竞赛

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /competitions/{competitionId}/users/{userId} |
| 方法类型 | Post |
| 控制器 | CompetitionController |
| 方法 | joinCompetition($competitionId,$userId) |
| 参数 | $userId:Integer  competitionId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 修改该竞赛的成员列表并返回竞赛场界面 |
| 备注 |  |

#### 6.3.1.6退出竞赛

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /competitions/{competitionId}/users/{userId} |
| 方法类型 | Delete |
| 控制器 | CompetitionController |
| 方法 | quitCompetition($competitionId,$userId) |
| 参数 | $userId:Integer  competitionId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 修改该竞赛的成员列表并返回竞赛场界面 |
| 备注 |  |

### 6.3.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| Competition::find | 查看某个竞赛 |
| Competition::findAll | 查看全部竞赛 |
| Competition::create | 新建竞赛 |
| Competition::delete | 删除竞赛 |
| Competition::getMembers | 查看竞赛的全部成员 |
| Competition::getRedTeamMembers | 查看团队竞赛中红队成员 |
| Competition::getBlueTeamMembers | 查看团队竞赛中蓝队成员 |
| Competition::getResultIndividual | 查看个人竞赛结果：用户排名以及各自的奖金 |
| Competition::getResultTeam | 查看团队竞赛结果：获胜队伍排名以及各自的奖金 |

## 6.4用户管理

### 6.4.1提供的接口设计

#### 6.4.1.1用户个人信息界面

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/userInfo |
| 方法类型 | Get |
| 控制器 | UserInfoController |
| 方法 | getUserInfo($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户个人信息界面 |
| 备注 |  |

#### 6.4.1.2修改用户个人信息

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/userInfo |
| 方法类型 | Post |
| 控制器 | UserInfoController |
| 方法 | updateUserInfo($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 更新用户个人信息并返回个人信息界面 |
| 备注 |  |

#### 6.4.1.3用户运动记录界面（指定日期）

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/deviceRecords/{date} |
| 方法类型 | Get |
| 控制器 | DeviceRecordController |
| 方法 | getRecordByDate($userId,$date) |
| 参数 | $userId:Integer  $date:Date |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户设备数据记录界面 |
| 备注 | 详情见数据定义1 |

#### 6.4.1.4用户数据统计界面

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/statistics |
| 方法类型 | Get |
| 控制器 | UserStatsController |
| 方法 | getUserStats($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户数据统计界面 |
| 备注 |  |

#### 6.4.1.6用户运动总况

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/statistics/summary |
| 方法类型 | Get |
| 控制器 | UserStatsController |
| 方法 | getUserStatsSum($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户数据统计中的概述数据 |
| 备注 | 详情见数据定义2 |

#### 6.4.1.7用户运动距离和时长

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/statistics/distance\_time/{duration} |
| 方法类型 | Get |
| 控制器 | UserStatsController |
| 方法 | getUserStatsDistAndTime($userId,$duration) |
| 参数 | $userId:Integer  $duration:String |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户数据统计中的概述数据 |
| 备注 | 详情见数据定义3 |

#### 6.4.1.8用户运动时间和燃脂

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/statistics/time\_calorie/{duration} |
| 方法类型 | Get |
| 控制器 | UserStatsController |
| 方法 | getUserStatsTimeAndKal($userId,$duration) |
| 参数 | $userId:Integer  $duration:String |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户数据统计中的概述数据 |
| 备注 | 详情见数据定义4 |

#### 6.4.1.9用户朋友界面

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/friends |
| 方法类型 | Get |
| 控制器 | UserFriendsController |
| 方法 | getUserFriends($userId) |
| 参数 | $userId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回用户的好友界面 |
| 备注 |  |

#### 6.4.1.10关注用户

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/watch/{watchId} |
| 方法类型 | Post |
| 控制器 | UserFriendsController |
| 方法 | follow($userId,$watchId) |
| 参数 | $userId:Integer  $watchId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 更新用户关注列表 |
| 备注 | Watch表示用户关注的人 |

#### 6.4.1.11取消关注用户

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /user/{ userId }/watch/{watchId} |
| 方法类型 | Delete |
| 控制器 | UserFriendsController |
| 方法 | unfollow($userId,$watchId) |
| 参数 | $userId:Integer  $watchId:Integer |
| 前置条件 | 用户已登录 |
| 后置条件 | 更新用户关注列表 |
| 备注 | Watch表示用户关注的人 |

#### 6.4.1.12搜索用户

|  |  |
| --- | --- |
| 动态版界面 |  |
| url | /search |
| 方法类型 | Get |
| 控制器 | SearchController |
| 方法 | search($keyword) |
| 参数 | $keyword:String |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回搜索结果(用户列表) |
| 备注 | 当无搜索结果时，提示用户没有匹配的用户名 |

### 6.4.2需要的接口

|  |  |
| --- | --- |
| 服务名 | 服务 |
| User::find | 查看用户个人信息 |
| User::update | 更新用户个人信息 |
| User::getFollowers | 查看用户粉丝 |
| User::getWatches | 查看用户关注 |
| DeviceReocrd::find | 查看特定日期的设备记录 |
| UserStats::findSummary | 查看用户统计数据 |
| UserStats::findDistAndTime | 查看用户运动距离和时间数据 |
| UserStats::findTimeAndCalorie | 查看用户运动时段和燃烧的卡路里数据 |
| Search::find | 查看用户搜索结果 |

# 7.数据格式定义

## 7.1用户运动记录

{

"data":{

"userId":"Integer",

"date":"YYYY-MM-DD",

"distance":"Double"

“start\_at”:”HH:MM”

"duration":"HH-MM-SS",

"avg\_pace":"MM-SS",

"calorie":"Double",

"avg\_speed":"Double",

"stride\_frequency":"Integer",

"avg\_stride":"Integer",

"total\_stride":"Integer",

"heart\_rate":"Integer"

}

}

## 7.2用户运动总况

{

"data":{

"total\_distance":"Integer",

"total\_duration":"Integer",

"total\_calorie":"Integer"

}

}

## 7.3用户运动距离和时长

{

"data":[

{

"day":"Integer",

"distance":"Double",

"hour":"Double"

},

{

"day":"Integer",

"distance":"Double",

"hour":"Double"

}

]

}

## 7.4用户运动强度

{

"data":[

{

"day":"Integer",

"calorie":"Double",

"hour":"Double"

},

{

"day":"Integer",

"calorie ":"Double",

"hour":"Double"

}

]

}