

**窦 妍**

**南京大学软件学院**

**2016-11-3**

Running Club

详细设计文档

目录

[1.引言 3](#_Toc8412)

[1.1编制目的 3](#_Toc18881)

[1.2词汇表 3](#_Toc28584)

[1.3参考资料 3](#_Toc15067)

[2.总体设计 3](#_Toc1659)

[2.1开发环境 3](#_Toc12325)

[2.2模块结构 3](#_Toc4344)

[2.3系统设计模式 4](#_Toc2745)

[2.4软件架构 4](#_Toc26550)

[3.数据库设计 4](#_Toc21809)

[3.1关系模型 4](#_Toc5592)

[4.详细设计 4](#_Toc13644)

[4.1运动模块 5](#_Toc31909)

[4.1.1提供的接口 5](#_Toc6066)

[4.1.2需要的接口 5](#_Toc3251)

[4.2用户模块 5](#_Toc20195)

[4.2.1提供的接口 5](#_Toc2469)

[4.2.2需要的接口 5](#_Toc26264)

[4.3活动模块 6](#_Toc13447)

[4.3.1提供的接口 6](#_Toc14102)

[4.3.2需要的接口 6](#_Toc23110)

[4.4小组模块 7](#_Toc24653)

[4.4.1提供的接口 7](#_Toc28392)

[4.4.2需要的接口 7](#_Toc21173)

[4.5主题帖模块 8](#_Toc25444)

[4.5.1提供的接口 8](#_Toc31813)

[4.5.2需要的接口 8](#_Toc4935)

[4.6消息模块 8](#_Toc8702)

[4.6.1提供的接口 8](#_Toc20627)

[4.6.2需要的接口 9](#_Toc29399)

[5.交互设计 9](#_Toc1908)

1.引言

1.1编制目的

本文档是Runnning Club运动社区网站详细设计文档，用于指导后续的软件开发，实现与测试及用户的沟通。

本文档说明的内容可能在项目实施过程中发生变更，但是必须由项目执行者仔细分析最终决定，建立持续有效的版本控制。

1.2词汇表

无

1.3参考资料

1）IEEE标准

2）RunningClub运动社区网站需求文档

2.总体设计

2.1开发环境

本系统采用php作为主要开发语言，服务端主要使用php＋sqlite+Apache，客户端使用html+css+js。用Apache作为服务器，采用sqlite作为后台数据管理系统。

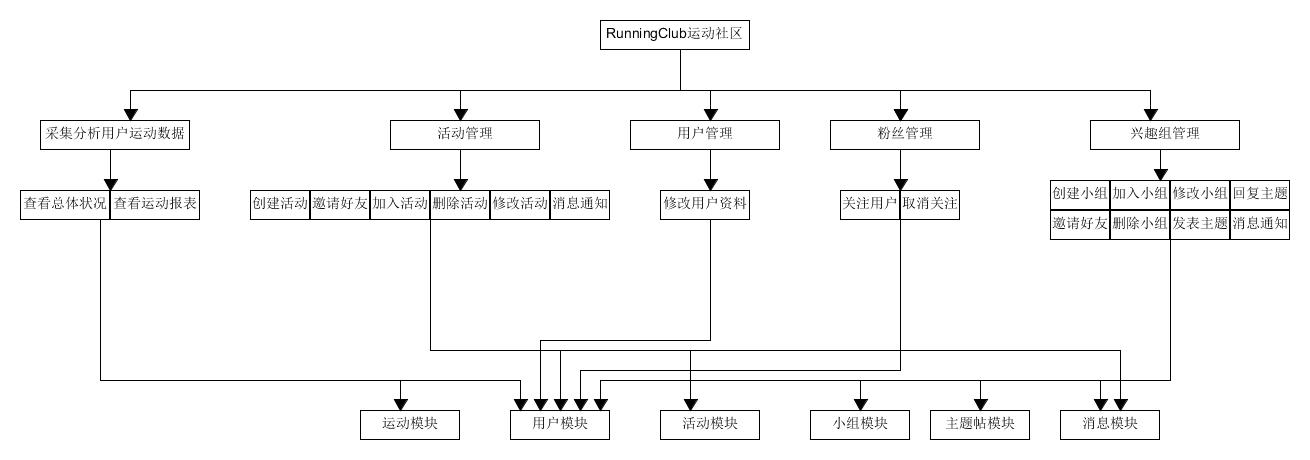
开发环境：OS X 10.10.5

开发工具：phpstorm

测试浏览器：Chrome, Edge, IE

服务器：Apache2

2.2模块结构



2.3系统设计模式

本系统基于MVC设计模式进行开发 。

2.4软件架构

Restful架构

3.数据库设计

3.1关系模型



4.详细设计

4.1运动模块

4.1.1提供的接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **控制器** | **方法** | **参数** | **前置条件** | **后置条件** |
| Get | SportController | /sports/{user}{?start,end,type} | 用户ID，  查询时段（可选），运动类型（可选） | 用户已登录并通过权限验证 | 显示用户运动分析 |

4.1.2需要的接口

|  |  |
| --- | --- |
| **服务名** | **服务** |
| Record::getRecordByUser(username) | 按照时间顺序返回某个用户的所有运动记录 |
| Record::getRecordByUser(username,starttime,endtime) | 返回用户在某段时间内所有运动记录 |

4.2用户模块

4.2.1提供的接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **控制器** | **方法** | **参数** | **前置条件** | **后置条件** |
| Post | UserController | /login?Id,password | 用户ID，密码 | 无 | 登录 |
| Post | UserController | /register?Id,password,email,login,description | 用户资料 | 无 | 创建账号 |
| Get | UserController | /user/{user} | 用户ID | 用户已登录 | 显示用户资料 |
| Get | UserController | /user/{user}/following | 用户ID | 用户已登录 | 显示用户关注的用户列表 |
| Post | UserController | /follow/{user} | 用户ID | 用户已登录 | 关注某人 |
| Post | UserController | /unfollow/{user} | 用户ID | 用户已登录 | 取消关注某人 |

4.2.2需要的接口

|  |  |
| --- | --- |
| **服务名** | **服务** |
| User::getUserById(id) | 返回用户资料 |
| User::save(Id,password,email,login,description) | 存储用户 |
| User::update(Id,password,email,login,description) | 修改用户资料 |
| Message::pushMessage(type,from,to) | 推送关注消息 |
| User::insertFollow(userId,followingId) | 插入一条关注数据 |
| User::deleteFollow(userId,followingId) | 删除一条关注数据 |

4.3活动模块

4.3.1提供的接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **控制器** | **方法** | **参数** | **前置条件** | **后置条件** |
| Post | EventController | /createActivity?Title,content,startAt,endAt,location,userId | 活动信息和创建者id | 用户已登录 | 创建一个活动 |
| Get | EventController | /activity/id | 活动id | 无 | 显示活动信息 |
| Get | EventController | /activitys | 无 | 无 | 按时间显示活动列表 |
| Get | EventController | /activitys?userId | 用户ID | 无 | 按时间显示用户创建和参加的活动列表 |
| Post | EventController | /updateActivity?Title,content,startAt,endAt,location,userId | 活动信息和创建者 | 用户已登录且拥有修改该活动的权限 | 修改活动 |
| Post | EventController | /deleteActivity?Id | 活动ID | 用户已登录且拥有删除该活动的权限 | 删除活动 |
| Post | EventController | /joinActivity?Id | 活动ID | 用户已登录 | 参加活动 |
| Post | EventController | /inviteActivity?fromId,toId | 邀请和被邀请的用户Id | 用户已登录 | 邀请参加活动 |

4.3.2需要的接口

|  |  |
| --- | --- |
| **服务名** | **服务** |
| Event::getEventById(id) | 返回活动信息 |
| Event::save（Title,content,startAt,endAt,location,userId) | 存储活动 |
| Event::update(Title,content,startAt,endAt,location,userId) | 修改活动资料 |
| Event::deleteEventById(id) | 删除活动 |
| Event::getEvents() | 返回所有活动 |
| Event:getEventsByUser(id) | 返回用户参加的所有活动 |
| User::getFollowingByUser(id) | 返回用户关注的用户列表 |
| Message::pushMessage(type,from,to) | 推送邀请消息 |

4.4小组模块

4.4.1提供的接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **控制器** | **方法** | **参数** | **前置条件** | **后置条件** |
| Post | GroupController | /createGroup?Name,description,userId | 小组信息和创建者id | 用户已登录 | 创建一个小组 |
| Get | GroupController | /Group/id | 小组id | 无 | 显示小组信息 |
| Get | GroupController | /Groups | 无 | 无 | 按时间显示小组列表 |
| Get | GroupController | /Groups?userId | 用户ID | 无 | 按时间显示用户创建和参加的小组列表 |
| Post | GroupController | /updateGroup?Name,description,userId | 小组信息和创建者ID | 用户已登录且拥有修改该小组的权限 | 修改小组 |
| Post | GroupController | /deleteGroup?Id | 小组ID | 用户已登录且拥有删除该小组的权限 | 删除小组 |
| Post | GroupController | /joinGroup?Id | 小组ID | 用户已登录 | 参加小组 |
| Post | GroupController | /inviteGroup?fromId,toId | 邀请和被邀请的用户Id | 用户已登录 | 邀请参加小组 |

4.4.2需要的接口

|  |  |
| --- | --- |
| **服务名** | **服务** |
| Group::getGroupById(id) | 返回小组信息 |
| Group::save（Title,content,startAt,endAt,location,userId) | 存储小组 |
| Group::update(Title,content,startAt,endAt,location,userId) | 修改小组资料 |
| Group::deleteGroupById(id) | 删除小组 |
| Group::getGroups() | 返回所有小组 |
| Group:getGroupsByUser(id) | 返回用户参加的所有小组 |
| User::getFollowingByUser(id) | 返回用户关注的用户列表 |
| Message::pushMessage(type,from,to) | 推送邀请消息 |

4.5主题帖模块

4.5.1提供的接口

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **控制器** | **方法** | **参数** | **前置条件** | **后置条件** |
| Post | TopicController | /createTopic?Title,content,author,groupId | 话题信息 | 用户已登录且拥有发帖权限 | 创建一个话题 |
| Get | TopicController | /Topic/id | 话题id | 无 | 显示话题信息 |
| Get | TopicController | /Topics/{groupId} | 小组ID | 无 | 按时间显示话题列表 |
| Post | TopicController | /createTopic?ParentID,content,author,groupId | 话题信息 | 用户已登录且拥有发帖权限 | 回复一个话题并通知作者 |
| Post | TopicController | /deleteTopic?Id | 话题ID | 用户已登录且拥有删除该话题的权限 | 删除话题 |
| Post | TopicController | /joinTopic?Id | 话题ID | 用户已登录 | 参加话题 |
| Post | TopicController | /inviteTopic?fromId,toId | 邀请和被邀请的用户Id | 用户已登录 | 邀请参加话题 |

4.5.2需要的接口

|  |  |
| --- | --- |
| **服务名** | **服务** |
| Topic::getTopicById(id) | 返回话题信息 |
| Topic::save（Title,content,startAt,endAt,location,userId) | 存储话题 |
| Topic::deleteTopicById(id) | 删除话题 |
| Topic::getTopics(groupID) | 返回所有话题 |
| User::getFollowingByUser(id) | 返回用户关注的用户列表 |
| Message::pushMessage(type,from,to) | 推送回复消息 |

4.6消息模块

4.6.1提供的接口

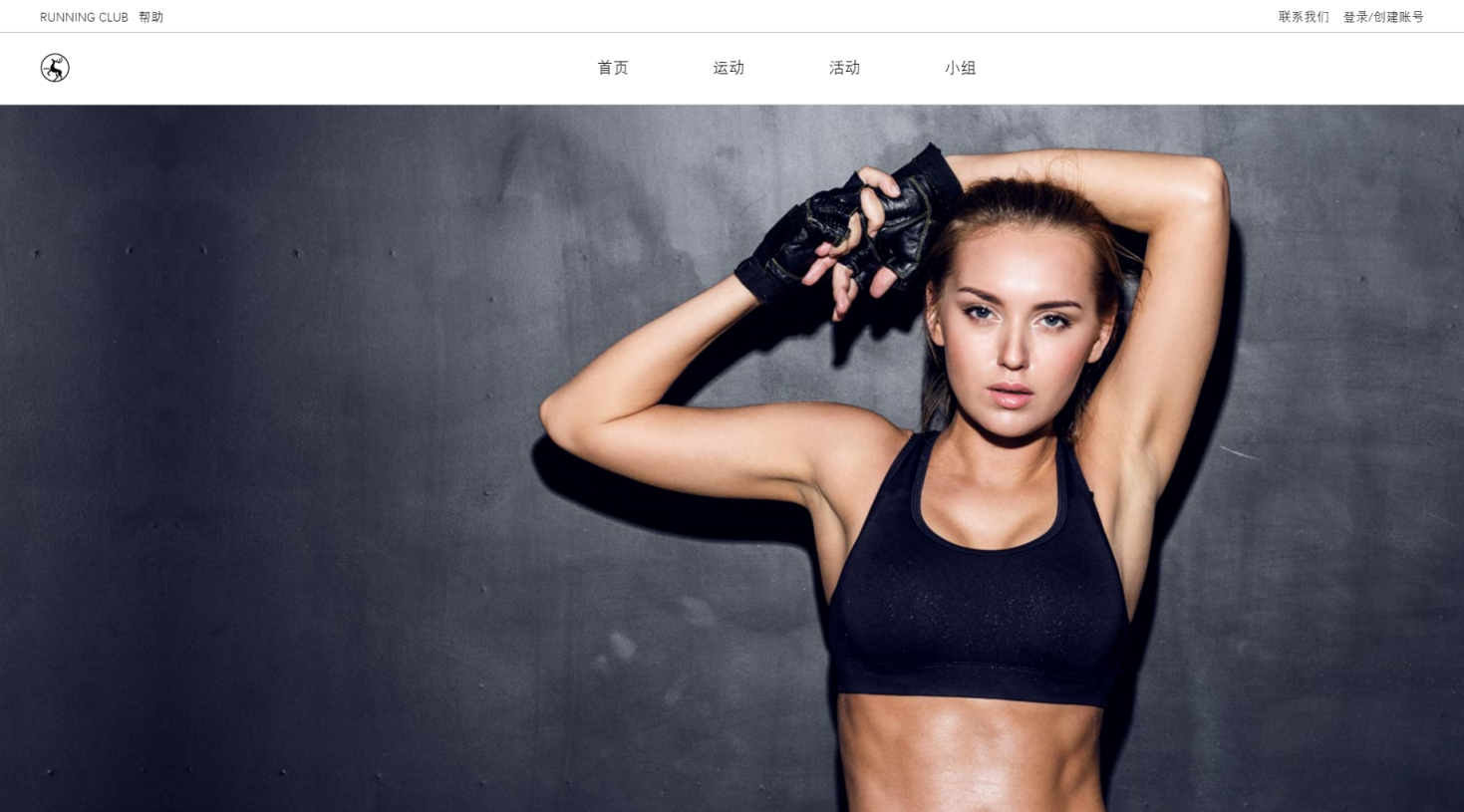
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **类型** | **控制器** | **方法** | **参数** | **前置条件** | **后置条件** |
| Post | MessageController | /pushMessage(type,from,to,content) | 消息类型、来源用户、目标用户、内容 | 用户已登录 | 发送一条通知 |
| Get | MessageController | /messages/{userId} | 用户ID | 用户已登录且拥有查看消息权限 | 显示未读通知 |
| Post | MessageController | /readMessage(id) | 消息id | 用户已登录 | 将通知置为已读状态 |

4.6.2需要的接口

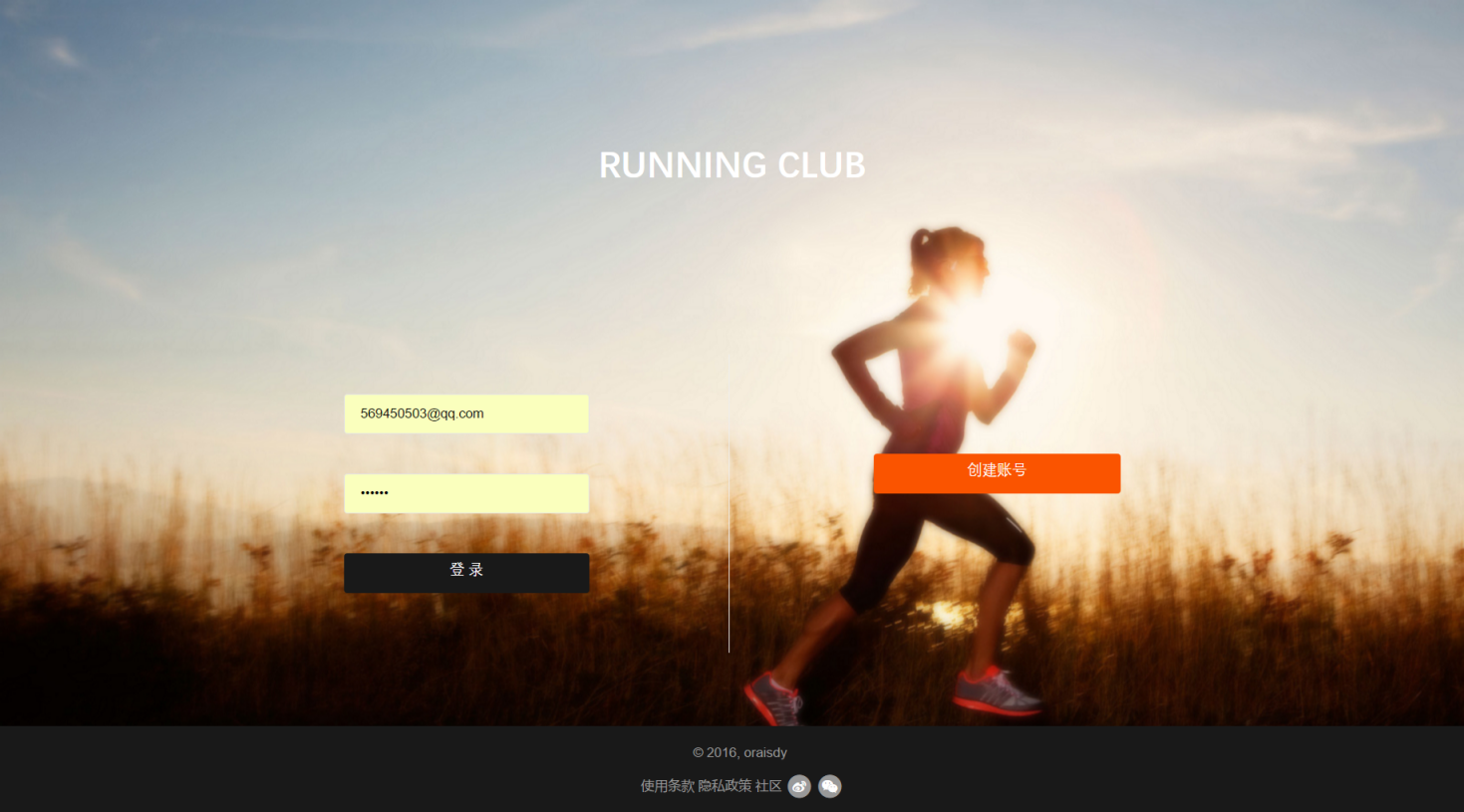
|  |  |
| --- | --- |
| **服务名** | **服务** |
| Message::pushMessage(type,from,to) | 推送消息 |
| Message::updateMessage(type,from,to,content,state) | 将消息状态更改为已读 |
| Message::getMessagesByUser(id) | 返回用户所有未读消息 |

1. 交互设计

首页



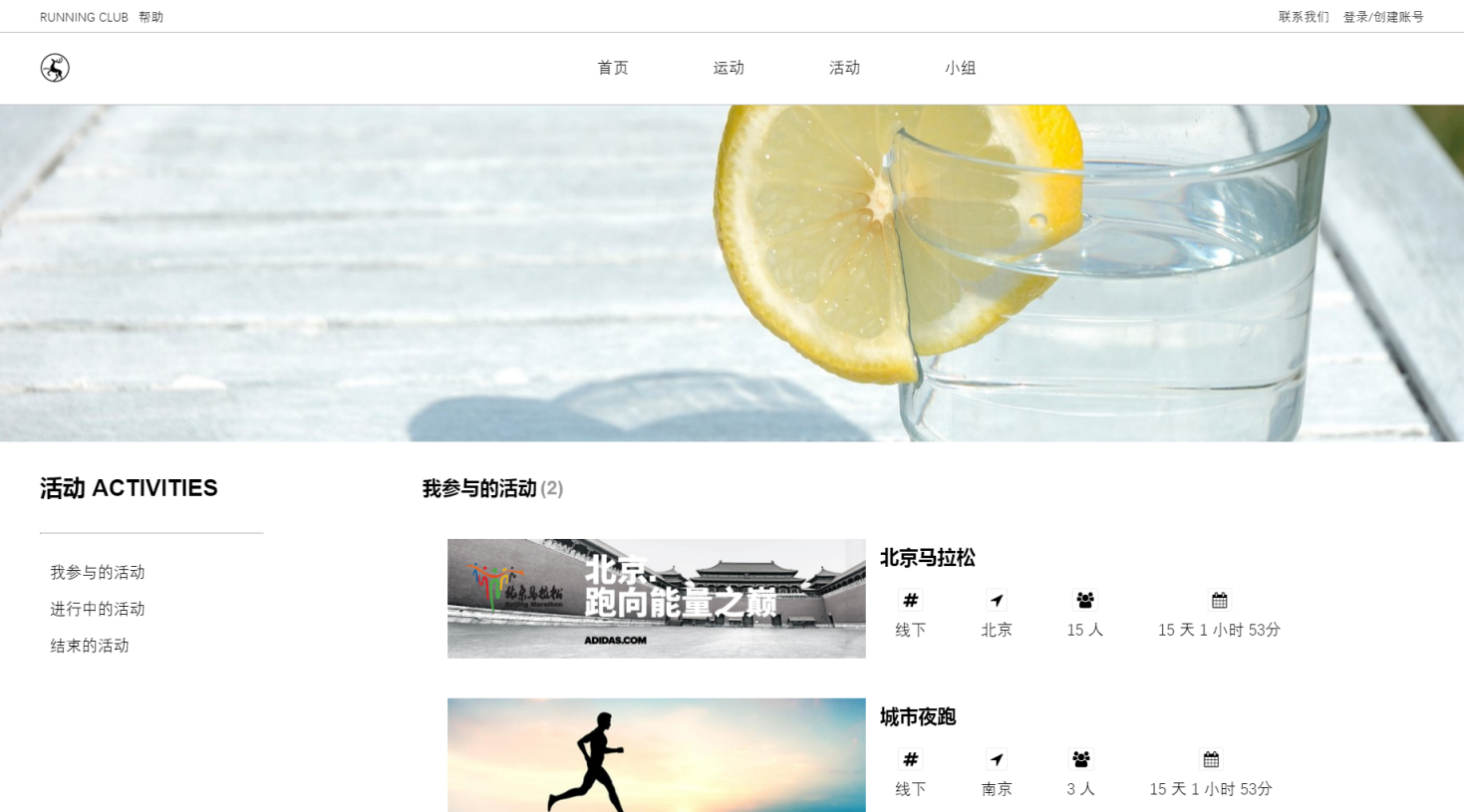
登录



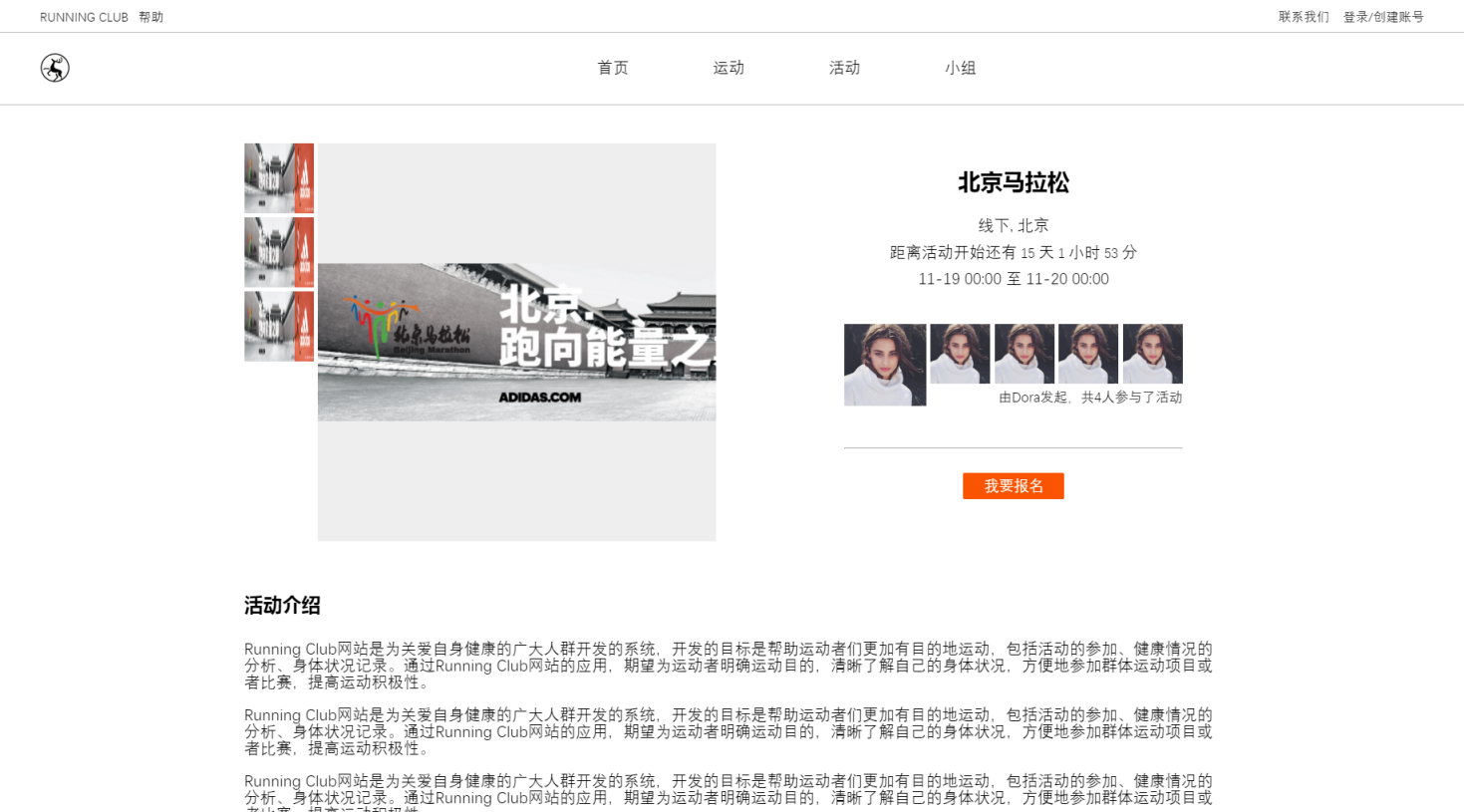
用户



活动



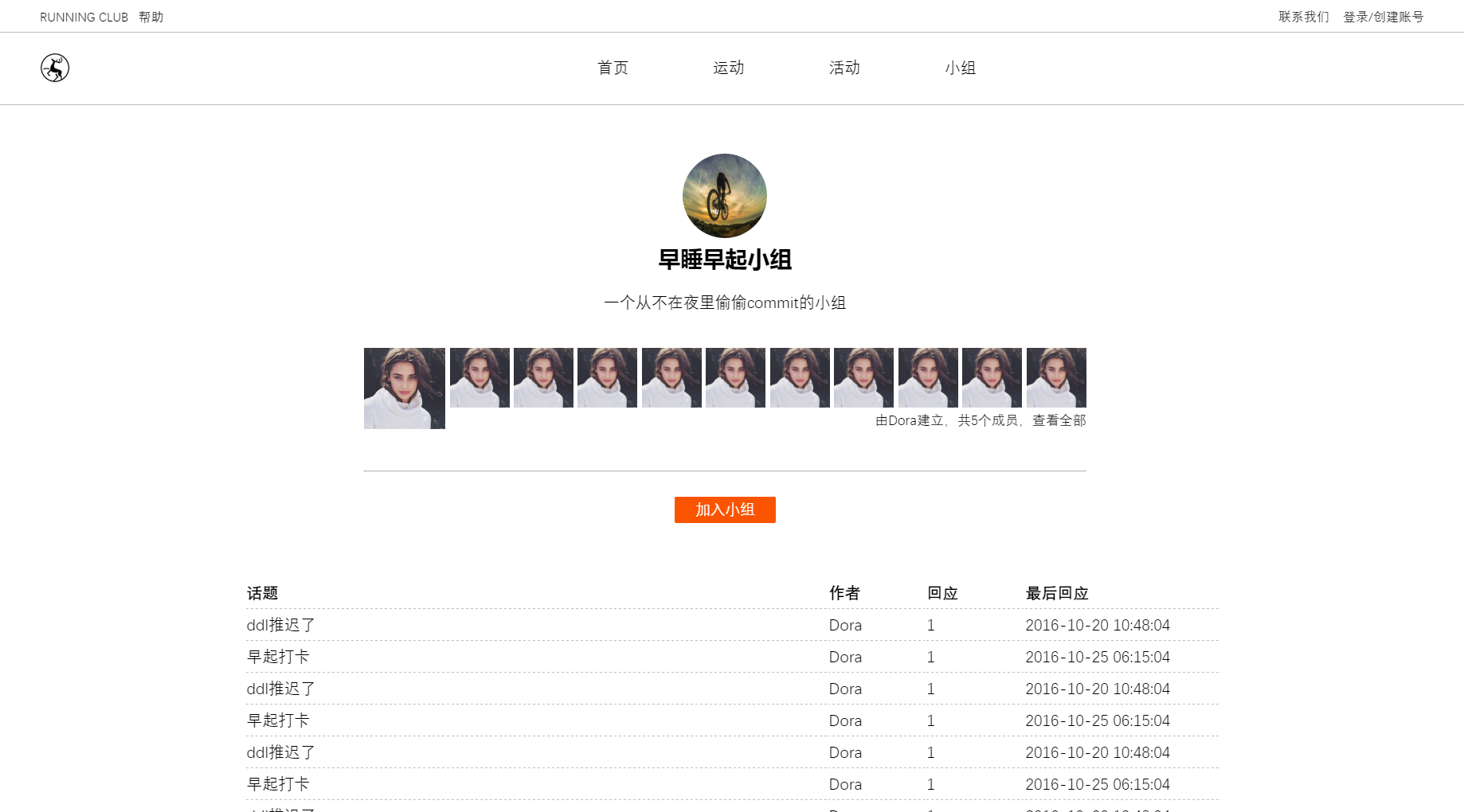
活动详细



小组



小组详细



主题帖

