|  |
| --- |
|  |
| iHealth  个人健康管理系统  详细设计文档 |
| **———关爱健康，从点滴做起** |

作者：华苏珺

学号：131250047

南京大学软件学院

2015年11月9日

| 修改人员 | 修改目的 | 更新日期 | 版本号 |
| --- | --- | --- | --- |
| 华苏珺 | 完成文档初稿 | 2015-11-03 | V1.0 |
| 华苏珺 | 添加导航设计 | 2015-11-05 | V1.1 |
| 华苏珺 | 添加详细设计 | 2015-11-06 | V1.2 |
| 华苏珺 | 添加交互设计 | 2015-11-09 | V1.3 |
| 华苏珺 | 定稿排版 | 2015-11-10 | V2.0 |

TOC \t "题目, 1,题目 2, 1,题目 3, 1"

1.引言 PAGEREF \_Toc \h 5

1.1编制目的 PAGEREF \_Toc1 \h 5

1.2词汇表 PAGEREF \_Toc2 \h 5

1.3参考资料 PAGEREF \_Toc3 \h 5

2.总体设计 PAGEREF \_Toc4 \h 5

2.1开发环境 PAGEREF \_Toc5 \h 5

2.2模块结构 PAGEREF \_Toc6 \h 5

2.3系统设计模式 PAGEREF \_Toc7 \h 5

2.4软件架构 PAGEREF \_Toc8 \h 7

3.数据库设计 PAGEREF \_Toc9 \h 7

3.1关系模型 PAGEREF \_Toc10 \h 7

4.导航设计 PAGEREF \_Toc11 \h 8

4.1整体导航设计 PAGEREF \_Toc12 \h 8

4.2健康管理导航 PAGEREF \_Toc13 \h 8

4.3活动导航 PAGEREF \_Toc14 \h 8

4.4兴趣组导航 PAGEREF \_Toc15 \h 9

4.5个人信息导航 PAGEREF \_Toc16 \h 9

5.详细设计 PAGEREF \_Toc17 \h 9

5.1登录/注册 PAGEREF \_Toc18 \h 10

5.1.1提供的接口设计 PAGEREF \_Toc19 \h 10

5.1.2需要的接口 PAGEREF \_Toc20 \h 11

5.2设备管理 PAGEREF \_Toc21 \h 11

5.2.1提供的接口设计 PAGEREF \_Toc22 \h 11

5.2.2需要的接口 PAGEREF \_Toc23 \h 11

5.3活动管理 PAGEREF \_Toc24 \h 13

5.3.1提供的接口设计 PAGEREF \_Toc25 \h 13

5.3.2需要的接口 PAGEREF \_Toc26 \h 14

5.4个人管理 PAGEREF \_Toc27 \h 15

5.4.1提供的接口设计 PAGEREF \_Toc28 \h 15

5.4.2需要的接口 PAGEREF \_Toc29 \h 16

5.5用户管理 PAGEREF \_Toc30 \h 16

5.5.1提供的接口设计 PAGEREF \_Toc31 \h 16

5.5.2需要的接口 PAGEREF \_Toc32 \h 16

5.6医生教练建议 PAGEREF \_Toc33 \h 18

5.6.1提供的接口设计 PAGEREF \_Toc34 \h 18

5.6.2需要的接口 PAGEREF \_Toc35 \h 19

5.7健康管理 PAGEREF \_Toc36 \h 20

5.7.1提供的接口设计 PAGEREF \_Toc37 \h 20

5.7.2需要的接口 PAGEREF \_Toc38 \h 21

5.8朋友圈 PAGEREF \_Toc39 \h 22

5.8.1提供的接口设计 PAGEREF \_Toc40 \h 22

5.8.2需要的接口 PAGEREF \_Toc41 \h 24

5.9兴趣组管理 PAGEREF \_Toc42 \h 26

5.9.1提供的接口设计 PAGEREF \_Toc43 \h 26

5.9.2需要的接口 PAGEREF \_Toc44 \h 28

6.交互设计 PAGEREF \_Toc45 \h 29

1.引言

1.1编制目的

本文档是iHealth个人健康管理系统详细设计文档，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）iHealth个人健康管理系统需求文档

2.总体设计

2.1开发环境

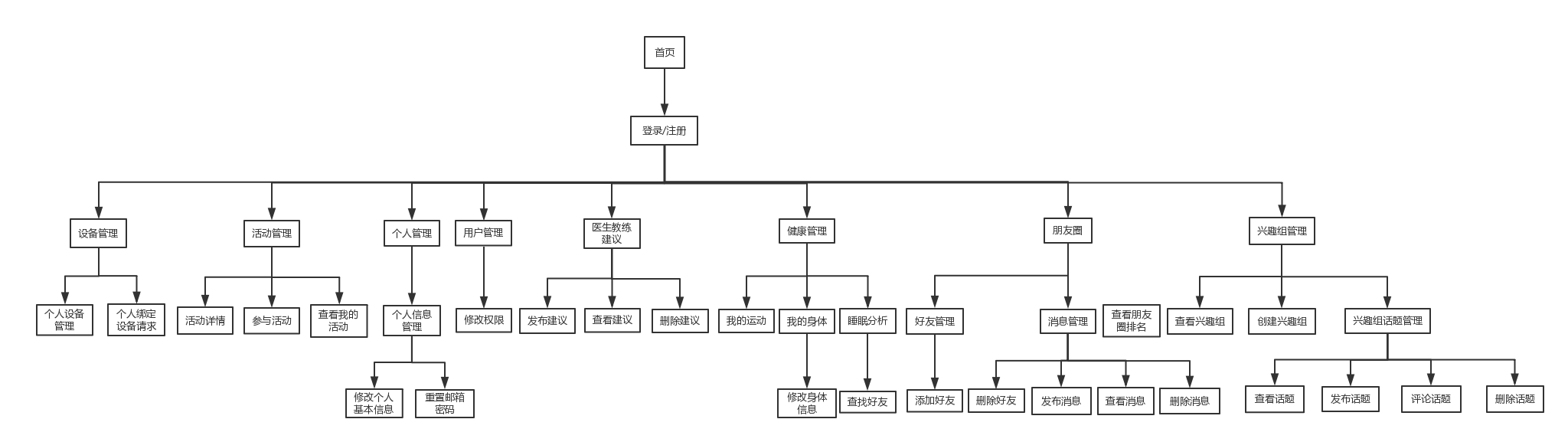
本系统采用php作为主要开发语言，服务端主要使用php＋sqlite+Apache，客户端使用html+css+js。用Apache作为服务器，采用sqlite作为后台数据管理系统。

开发环境：OS X 10.10.5

开发工具：phpstorm

测试浏览器：Safari，Firefox

服务器：Apache2

2.2模块结构

2.3系统设计模式

本系统采用Yii框架进行开发 ，基于MVC设计模式。

MVC设计模式组建图

除MVC外各个部分功能：

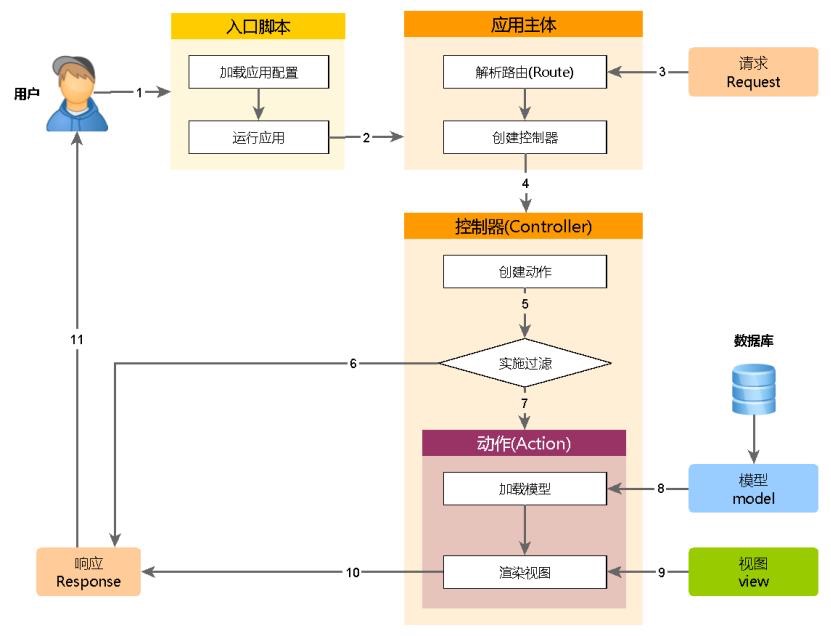
入口脚本：终端用户能直接访问的PHP脚本，负责启动一个请求处理周期。

应用：能全局范围内访问的对象，管理协调组件来完成请求.

应用组件：在应用中注册的对象，提供不同的功能来完成请求。

模块：包含完整MVC结构的独立包，一个应用可以由多个模块组建。

过滤器：控制器在处理请求之前或之后需要触发执行的代码。

小部件：可嵌入到视图中的对象，可包含控制器逻辑，可被不同视图重复调用。

Yii请求处理示意图

Yii请求处理示意图

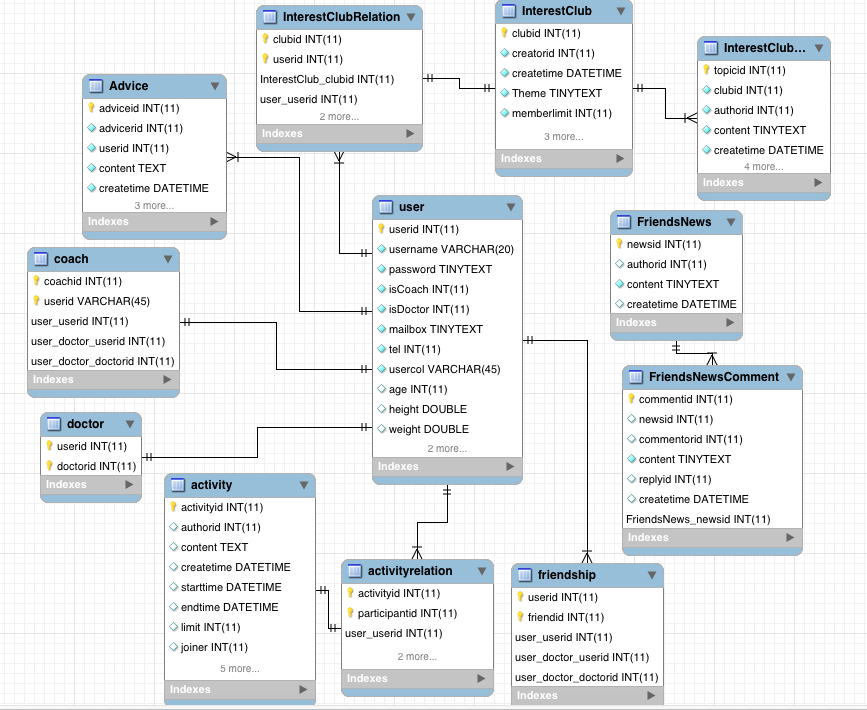
每一次 Yii 应用开始处理HTTP请求时，它都会进行一个近似的流程。

1.用户提交指向入口脚本web/index.php的请求。

2.入口脚本会加载配置数组并创建一个应用实例用于处理该请求。

3.应用会通过request（请求）应用组件解析被请求的路由。

4.应用创建一个controller（控制器）实例具体处理请求。

5.控制器会创建一个action（动作）实例并为该动作执行相关的Filters（访问过滤器）。

6.如果任何一个过滤器验证失败，该动作会被取消。

7.如果全部的过滤器都通过，该动作就会被执行。

8.动作会加载一个数据模型，一般是从数据库中加载。

9.动作会渲染一个View（视图），并为其提供所需的数据模型

10.渲染得到的结果会返回给response（响应）应用组件。

11.响应组件会把渲染结果发回给用户的浏览器。

2.4软件架构

Restful架构

3.数据库设计

3.1关系模型

4.导航设计

4.1整体导航设计

网站的主导航包括：首页、健康管理、活动、兴趣组以及个人信息。

4.2健康管理导航



4.3活动导航



4.4兴趣组导航



4.5个人信息导航

5.详细设计

5.1登录/注册

5.1.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 登录界面 | url | /Login |
| 方法类型 | Get |
| 控制器 | AuthController |
| 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 |
| 登录请求 | url | /Login |
| 方法类型 | Post |
| 控制器 | AuthController |
| 方法 | postLogin(Request$request) |
| 参数 | 用户名，密码（密文） |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回登录信息，失败返回失败提示信息 |
| 注册界面 | url | /SignUp |
| 方法类型 | Get |
| 控制器 | AuthController |
| 方法 | getRegister() |
| 参数 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册界面 |
| 注册请求 | url | /SignUp |
| 方法类型 | Post |
| 控制器 | AuthController |
| 方法 | postRegister(Request$request) |
| 参数 | 邮箱，用户名，密码（密文） |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册结果 |

5.1.2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 |
| User::save | 保存用户信息并更新数据库 |

5.2设备管理

5.2.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 个人设备管理 | url | /Device |
| 方法类型 | Get |
| 控制器 | UserController |
| 方法 | getDevice() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回设备管理界面 |
| 个人绑定设备请求 | url | /device |
| 方法类型 | Post |
| 控制器 | UserController |
| 方法 | postDeviceBind(Request$request) |
| 参数 | 设备编号 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回绑定结果 |

5.2.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| User::find | 查找用户 |
| User::save | 保存用户信息并更新数据库 |

5.3活动管理

5.3.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 活动界面 | url | /Activity |
| 方法类型 | Get |
| 控制器 | ActivityController |
| 方法 | getActivityList() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回活动界面 |
| 活动详情 | url | /Activity |
| 方法类型 | Get |
| 控制器 | ActivityController |
| 方法 | getActivity() |
| 参数 | 活动id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回活动详情 |
| 参与活动 | url | /Activity |
| 方法类型 | Post |
| 控制器 | GameController |
| 方法 | participateIn(Request$request) |
| 参数 | 活动id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回活动界面 |
| 查看我的活动 | url | /Activity |
| 方法类型 | Get |
| 控制器 | ActivityController |
| 方法 | getMyActivity() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回我的活动 |

5.3.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| Activity::getActivityList | 获得活动列表 |
| Activity::getActivity | 获得活动详情 |
| Activity::participateInActivity | 参与活动 |
| User::getMyActivity | 查看我的活动 |

5.4个人管理

5.4.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 个人信息界面 | url | /SelfInfo |
| 方法类型 | Get |
| 控制器 | UserController |
| 方法 | getUserInfo() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回个人信息界面 |
| 修改个人信息请求 | url | /SelfInfo |
| 方法类型 | Post |
| 控制器 | UserController |
| 方法 | postReset(Request$request) |
| 参数 | 昵称，个性域名，性别，生日，所在地，兴趣，运动宣言 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回修改结果 |
| 重置邮箱密码页面 | url | /SelfInfo |
| 方法类型 | Get |
| 控制器 | PasswordController |
| 方法 | getReset() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回重置密码页面 |
| 重置邮箱密码请求 | url | /SelfInfo |
| 方法类型 | Post |
| 控制器 | PasswordController |
| 方法 | postReset |
| 参数 | 邮箱或密码 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回重置结果 |

5.4.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| User::find | 查找用户 |
| User::save | 保存用户信息并更新数据库 |

5.5用户管理

5.5.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 用户权限管理界面 | url | /UserAdmin |
| 方法类型 | Get |
| 控制器 | UserAdminController |
| 方法 | UserAdminIndex() |
| 参数 | 无 |
| 前置条件 | 管理员已登录 |
| 后置条件 | 返回用户权限管理界面 |
| 权限修改请求 | url | /UserAdmin |
| 方法类型 | Post |
| 控制器 | UserAdminController |
| 方法 | modifyUserAdmin(Request$request) |
| 参数 | 用户信息，变更权限 |
| 前置条件 | 管理员已登录 |
| 后置条件 | 保存用户权限 |

5.5.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| User::find | 查找用户 |
| Admin::save | 保存权限修改信息 |

5.6医生教练建议

5.6.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 建议界面 | url | /PersonalCentre |
| 方法类型 | Get |
| 控制器 | DocnteController |
| 方法 | getSuggestion() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回建议结果 |
| 发布建议 | url | /PersonalCentre |
| 方法类型 | Post |
| 控制器 | DocnteController |
| 方法 | postSuggestion(Request$request) |
| 参数 | 医生id，建议内容 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回发布结果 |
| 查看建议 | url | /PersonalCentre |
| 方法类型 | Get |
| 控制器 | DocnteController |
| 方法 | getSuggestion() |
| 参数 | 建议id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回查看建议界面 |
| 删除建议 | url | /PersonalCentre |
| 方法类型 | Delete |
| 控制器 | DocnteController |
| 方法 | deleteSuggestion() |
| 参数 | 建议id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回删除建议界面 |

5.6.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| Suggestion::create | 创建建议并加入数据库 |
| Suggestion::find | 查找建议 |
| Suggestion::save | 删除建议并更新数据库 |

5.7健康管理

5.7.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 运动界面 | url | /PersonalCentre |
| 方法类型 | Get |
| 控制器 | SportsController |
| 方法 | sportsIndex() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回运动主界面 |
| 身体管理界面 | url | /PersonalCentre |
| 方法类型 | Get |
| 控制器 | BodyController |
| 方法 | bodyIndex() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回身体管理界面 |
| 修改身体信息请求 | url | /PersonalCentre |
| 方法类型 | Post |
| 控制器 | BodyController |
| 方法 | modifyBodyInfo(Request$request) |
| 参数 | 身高，体重 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回身体管理界面 |
| 睡眠分析界面 | url | /PersonalCentre |
| 方法类型 | Get |
| 控制器 | SleepController |
| 方法 | sleepIndex() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回睡眠分析界面 |

5.7.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| User::getSportsRecord | 获得运动信息 |
| User::getBodyRecord | 获得身体信息 |
| User::modifyBodyRecord | 修改身体信息 |
| User::getSleepRecord | 获得睡眠信息 |

5.8朋友圈

5.8.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 朋友圈界面 | url | /SelfInfo/friends |
| 方法类型 | Get |
| 控制器 | FriendsController |
| 方法 | getFriends() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回好友列表 |
| 查找好友 | url | /SelfInfo/friends |
| 方法类型 | Post |
| 控制器 | FriendsController |
| 方法 | findFriend() |
| 参数 | 好友id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回查找结果 |
| 添加好友 | url | /SelfInfo/friends |
| 方法类型 | Post |
| 控制器 | FriendsController |
| 方法 | makeFriend() |
| 参数 | 用户id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回添加结果 |
| 删除好友 | url | /SelfInfo/friends |
| 方法类型 | Delete |
| 控制器 | FriendsController |
| 方法 | deleteFriend(Request$request) |
| 参数 | 好友id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回查找好友界面 |
| 查看朋友圈排名 | url | /SelfInfo/friends |
| 方法类型 | Get |
| 控制器 | FriendsController |
| 方法 | getRank() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回排名信息 |
| 发布朋友圈消息 | url | /SelfInfo/friends |
| 方法类型 | Post |
| 控制器 | FriendsController |
| 方法 | postNews(Request$request) |
| 参数 | 消息内容 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回发布情况 |
| 查看朋友圈消息 | url | /SelfInfo/friends |
| 方法类型 | Get |
| 控制器 | FriendsController |
| 方法 | getNews() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回朋友圈消息 |
| 删除朋友圈消息 | url | /SelfInfo/friends |
| 方法类型 | Delete |
| 控制器 | FriendsController |
| 方法 | deleteNews(Request$request) |
| 参数 | 消息id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回删除情况 |

5.8.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| User::getFriends | 获得好友列表 |
| User::findFriend | 查找好友 |
| User::makeFriend | 添加好友 |
| User::deleteFriend | 删除好友 |
| User::getRank | 查看朋友圈排名 |
| User::getNews | 查找朋友圈消息 |
| User::deleteNews | 删除朋友圈消息 |

5.9兴趣组管理

5.9.1提供的接口设计

|  |  |  |
| --- | --- | --- |
| 兴趣组界面 | url | /InterestClub |
| 方法类型 | Get |
| 控制器 | InterestClubController |
| 方法 | InterestClubIndex() |
| 参数 | 无 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回兴趣组界面 |
| 查看兴趣组 | url | /InterestClub |
| 方法类型 | Get |
| 控制器 | InterestClubController |
| 方法 | getInterestClub() |
| 参数 | 兴趣组id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回兴趣组详情 |
| 创建兴趣组 | url | /InterestClub |
| 方法类型 | Post |
| 控制器 | InterestClubController |
| 方法 | createInterestClub(Request$request) |
| 参数 | 兴趣组名称，简介 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回创建情况 |
| 查看兴趣组话题 | url | /InterestClub |
| 方法类型 | Get |
| 控制器 | InterestClubController |
| 方法 | getInterestClubTopic() |
| 参数 | 话题id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回话题列表 |
| 发布兴趣组话题 | url | /InterestClub |
| 方法类型 | Post |
| 控制器 | InterestClubController |
| 方法 | createInterestClubTopic(Request$request) |
| 参数 | 话题内容 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回发布情况 |
| 评论兴趣组话题 | url | /InterestClub |
| 方法类型 | Post |
| 控制器 | InterestClubController |
| 方法 | reviewInterestClubTopic(Request$request) |
| 参数 | 评论内容 |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回评论情况 |
| 删除兴趣组话题 | url | /InterestClub |
| 方法类型 | Post |
| 控制器 | InterestClubController |
| 方法 | deleteInterestClubTopic(Request$request) |
| 参数 | 兴趣组id |
| 前置条件 | 用户已登录 |
| 后置条件 | 返回删除情况 |

5.9.2需要的接口

| 服务名 | 服务 |
| --- | --- |
| InterestClub::getInterestClubList | 获得兴趣组列表 |
| InterestClub::find | 查看兴趣组 |
| InterestClub::create | 创建兴趣组 |
| InterestClub::createTopic | 发布兴趣组话题 |
| InterestClub::findTopic | 查看兴趣组话题 |
| InterestClub::reviewTopic | 评论兴趣组话题 |
| InterestClub::deleteTopic | 删除兴趣组话题 |

6.交互设计











