详细设计描述文档

WORTHY运动社交应用

学校： 南京大学

院系： 软件学院

编写人员：141250047何永俊

目录

1 引言 3

1.1编制目的 3

1.2词汇表 3

1.3导读 3

1.4参考资料 3

2 产品概述 4

3 总体设计…………………………………………………………………………………………………….4

3.1开发环境…………………………………………………………………………………………….4

3.2模块结构…………………………………………………………………………………………….4

3.3系统设计模式……………………………………………………………………………………..4

3.4软件架构…………………………………………………………………………………………….5

4 数据库设计

5 详细设计

6 导航设计

# 1 引言

## 1.1编制目的

本文档承接web需求规格说明文档，继续完成web运动社交应用的详细设计工作，达到指导后续的软件构造的目的。

本文档的预期读者为开发人员、构造人员、测试人员、最终用户以及项目管理人员。

## 1.2词汇表

|  |  |  |
| --- | --- | --- |
| 词汇名称 | 词汇含义 | 备注 |
| WORTHY | 运动设计应用 | 为本次工程的开发产品 |

## 1.3导读

该报告分为5个部分，第一部分是引言，主要是介绍文档的基本信息；第二部分是产品概述，是快递物流系统的概括性描述；第三部分是对体系结构设计的概述；第四、第五和第六部分分别是结构视角、依赖视角，从多个视角对软件设计进行详细的描述。

## 1.4参考资料

《软件工程与计算（卷二）——软件开发的技术基础》 机械工程出版社

《WEB需求规格说明文档》

# 2 产品概述

Worthy运动社交应用是为有锻炼需求的人群开发的web应用，开发的目的是通过记录用户身体指标和运动数据，提供统计分析，让用户对自己的健康及运动状况有更好的了解。

# 3总体设计

## 3.1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。

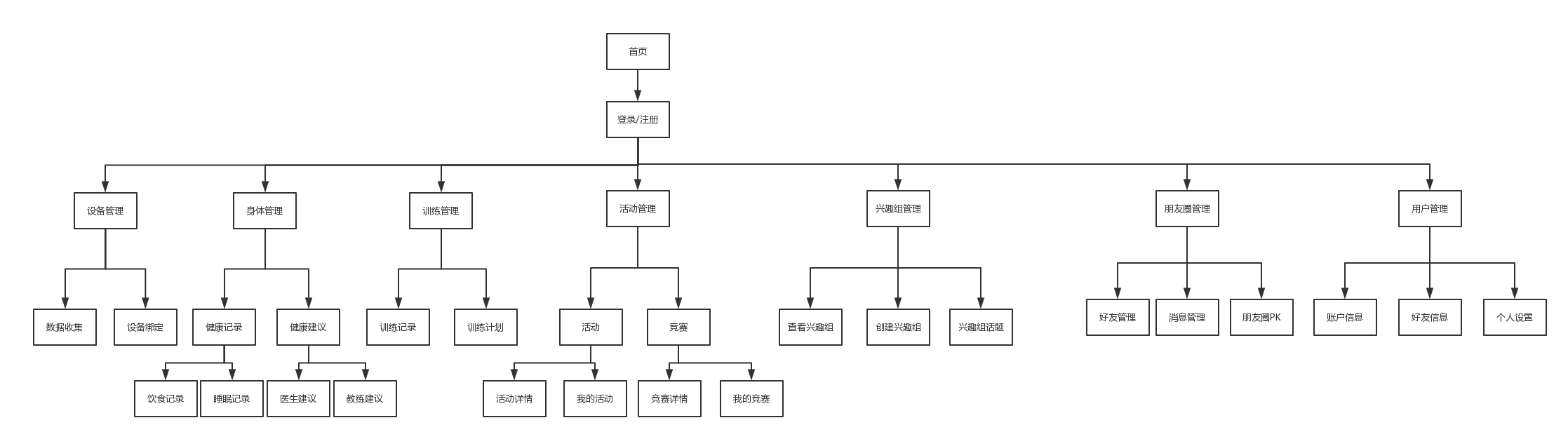
开发环境：WINDOWS 10

开发工具：eclipse jee

测试浏览器：Maxthon浏览器

服务器：Apache2

## 3.2模块结构



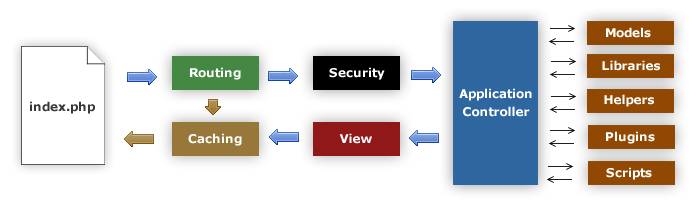
## 3.3系统设计模式

本系统采用CodeIgniter框架进行开发 ，基于 MVC（模型-视图-控制器）设计模式。MVC 是一种 用于将应用程序的逻辑层和表现层分离出来的软件方法。

模型:代表你的数据结构。通常来说，模型类将包含帮助你对数据库进行增删改查的方法。

视图:是要展现给用户的信息。一个视图通常就是一个网页，但是在 CodeIgniter 中，一个视图也可以是一部分页面（例如页头、页尾），它也可以是一个 RSS 页面，或其他任何类型的页面。

控制器:是模型、视图以及其他任何处理 HTTP 请求所必须的资源之间的中介，并生成网页。

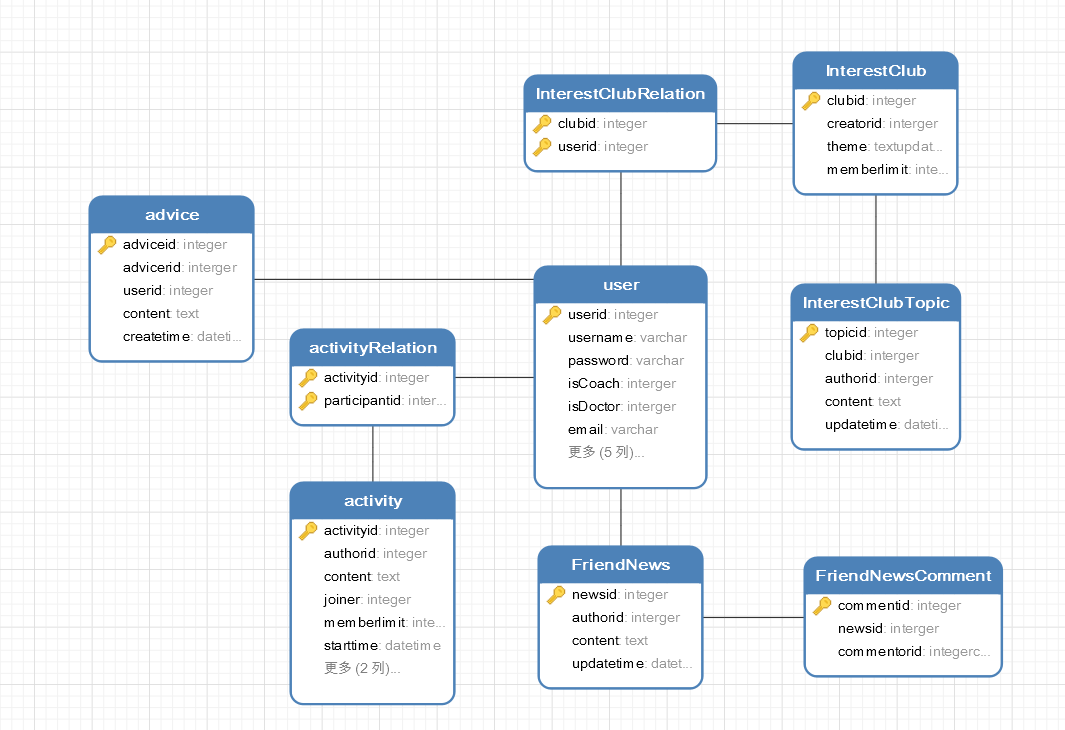


1. index.php 文件作为前端控制器，初始化运行 CodeIgniter 所需的基本资源；
2. Router 检查 HTTP 请求，以确定如何处理该请求；
3. 如果存在缓存文件，将直接输出到浏览器，不用走下面正常的系统流程；
4. 在加载应用程序控制器之前，对 HTTP 请求以及任何用户提交的数据进行安全检查；
5. 控制器加载模型、核心类库、辅助函数以及其他所有处理请求所需的资源；
6. 最后一步，渲染视图并发送至浏览器，如果开启了缓存，视图被会先缓存起来用于 后续的请求。

## 3.4软件架构

Restful软件架构

# 4数据库设计



# 5详细设计

## 5.1用户管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

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

## 5.2个人管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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设备管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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 |

## 5.4活动管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 | ActivityController |
| 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 |

需要的服务（需接口）

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

## 5.5健康管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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 |

## 5.6身体管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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 |

## 5.7朋友圈

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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 |

## 5.8兴趣组管理

接口规范

|  |  |  |
| --- | --- | --- |
| 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 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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 |

## 5.9登录/注册

接口规范

|  |  |  |
| --- | --- | --- |
| 登录界面 | url | Login/ |
| 方法类型 | Get |
| 方法 | getLogin() |
| 参数 | 无 |
| 控制器 | AuthController |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回登录界面 |
| 登录界面 | url | Login/ |
| 方法类型 | Post |
| 方法 | postLogin(Request$request) |
| 参数 | 用户名，密码 |
| 控制器 | AuthController |
| 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 |

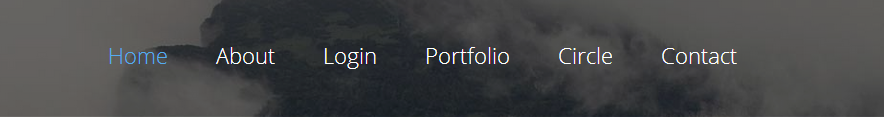
|  |  |  |
| --- | --- | --- |
| 注册界面 | url | /SignUp |
| 方法类型 | Get |
| 方法 | AuthController |
| 参数 | getRegister() |
| 控制器 | 无 |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册界面 |
| 注册请求 | url | /SignUp |
| 方法类型 | Post |
| 方法 | AuthController |
| 参数 | postRegister(Request$request) |
| 控制器 | 邮箱，用户名，密码（密文） |
| 前置条件 | 用户未登录 |
| 后置条件 | 返回注册结果 |

需要的服务（需接口）

|  |  |
| --- | --- |
| 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 |

# 6导航设计

## 6.1首页导航



## 6.2整体导航



## 6.3个人信息导航



## 6.4训练导航

