知康——个人健康Web应用

设计文档

131250218 王振聪

南京大学 软件学院

**更新历史**

|  |  |  |
| --- | --- | --- |
| 日期 | 变更原因 | 版本号 |
| 2015/10/15 | 初始化设计文档 | V0.1草稿版 |
|  |  |  |
| 2015/10/30 | 审核校对后的版本 | V1.0正式版 |

**目录**

**未找到目录项。**

1. **引言**
   1. 编制目的

本文档详细完成对“知康”——个人健康Web应用的体系结构设计和模块的详细设计，达到指导后续软件构造的目的，同时实现和测试人员及用户的沟通。

本文档面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

* 1. 词汇表

|  |  |  |
| --- | --- | --- |
| 词汇含义 | 词汇名称 | 备注 |
| Healther | 知康 | 应用名称 |
|  |  |  |

* 1. 参考文献

《知康——个人健康Web应用 需求规格说明书》

1. **产品概述**

“知康”是一款关注个人健康的Web2.0应用，它结合了当下流行的互联网社交元素，加上原本的常规运动健康项目，以此来关注个人健康。“知康”主要包括以下几个模块：运动健身和个人健康数据采集和数据统计分析；活动、朋友圈、兴趣组等社交行为；教练和医生提供健康建议；等等。

系统的用例图如图2-1所示。

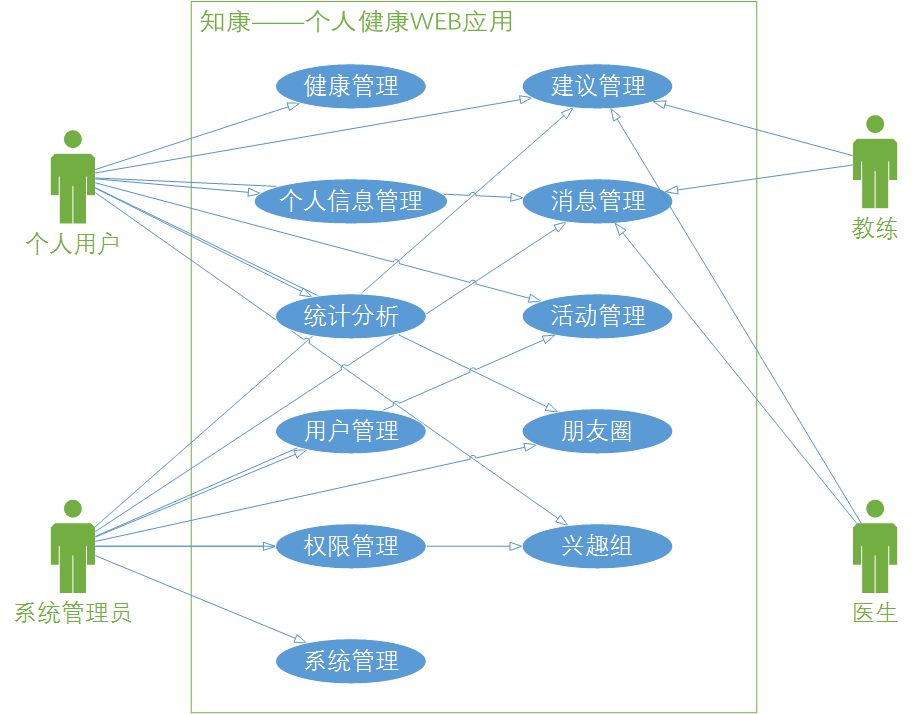


图2-1 用例图

系统的用例描述如表2-1所示。

表2-1 用例描述表

|  |  |
| --- | --- |
| 参与者 | 用例 |
| 个人用户 | 健康管理  活动管理  建议管理（查看）  统计分析  朋友圈  兴趣组  消息管理  个人信息管理 |
| 教练（在个人用户基础上） | 建议管理（发表）  消息管理 |
| 医生（在个人用户基础上） | 建议管理（发表）  消息管理 |
| 系统管理员 | 用户管理  权限管理  活动管理  建议管理（管理）  朋友圈（管理）  兴趣组（管理）  消息管理  系统管理 |

1. **体系结构设计**
   1. 系统概述

“知康”是一款关注个人健康的WEB应用，分为浏览器端（前端）和服务器端（后端）两个部分。前端使用Html、CSS、JavaScript语言编写，采用Bootstrap前端框架，并且使用Ajax技术。后端使用PHP语言编写，采用Laravel 5.1框架，其中数据库方面使用SQLite 3。

**Laravel开发框架：**

Laravel是一款采用MVC架构的PHP应用开发框架，Laravel的应用目录结构如表3-1-1所示。

表3-1-1 Laravel应用目录结构

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

app文件夹包含了应用的核心代码，app目录结构如表3-1-2所示。

表3-1-2 Laravel 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 |

Laravel是model-view-controller（MVC）架构模式，其中Model在app目录下，View在/Resources/views目录下，Controller在/app/Http/Controllers目录下。Laravel的一次响应流程如图3-1-1所示。

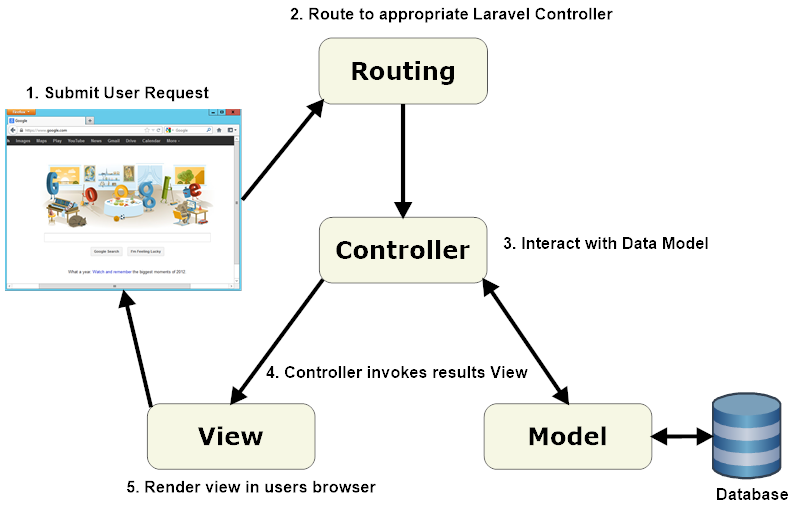


图3-1-1 Laravel的一次响应流程

当访问一个laravel应用程序时，浏览器发送一个请求，Laravel应用的所有请求入口都是public/index.php文件，所有请求都会被web服务器导向这个文件。接下来Laravel启动内核，Web服务器将请求传递到Laravel的路由引擎。该Laravel路由器接收到请求后，根据配置重定向到相应的控制器类的方法。然后由控制器类接管。在某些情况下，控制器将立即渲染一个视图，这是一个模板，将被转换成HTML并且发送回浏览器。更普遍的动态网站，控制器与模型进行交互，与数据库进行通信。调用模型后,控制器呈现最终视图(HTML、CSS和图像)并返回完整的web页面到用户的浏览器。

* 1. 体系结构模型

系统的体系结构高层抽象模型如图3-2-1所示。

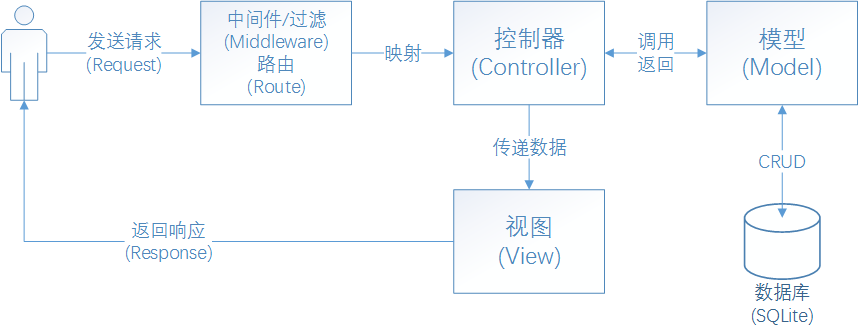


图3-2-1 系统体系结构高层抽象模型

在系统的体系结构高层抽象模型中，最主要的是控制器层、模型层和视图层。控制器层负责封装外界作用于模型的操作和对数据流向的控制，即负责分发和处理用户的请求；模型层负责封装数据和所有基于对这些数据的基本操作；视图层负责封装数据的显示，即用户界面层。

根据需求，继续细化系统的模块。系统分为前台(Front)和后台(Admin)两个部分，前台涉及的用户对象有个人用户、教练、医生、系统管理员，后台涉及的用户对象是系统管理员，其中教练和医生是在个人用户基础之上的（拥有个人用户的权限）。前台系统模块如表3-2-1所示，后台系统模块如表3-2-2所示。表中的用户认证模块(Auth)是指注册登录、用户权限认证模块，默认模块(Index)是指首页和其他不能单独成模块的小功能。

表3-2-1 前台系统模块

|  |  |  |
| --- | --- | --- |
| 模块 | 模块名称 | 涉及的用户对象 |
| Health | 健康管理 | 个人用户 |
| Activity | 活动管理 | 个人用户 |
| Advice | 建议管理 | 个人用户、教练、医生 |
| Article | 文章管理 | 个人用户、教练、医生 |
| Analysis | 统计分析 | 个人用户 |
| Message | 消息管理 | 个人用户 |
| Personal | 个人信息管理 | 个人用户 |
| Friend | 好友管理 | 个人用户 |
| Moments | 朋友圈 | 个人用户 |
| Group | 兴趣组 | 个人用户 |
| Auth | 用户认证\* | 个人用户、系统管理员 |
| Index | 默认模块\* | 个人用户、教练、医生、系统管理员 |

表3-2-2 后台系统模块

|  |  |  |
| --- | --- | --- |
| 模块 | 模块名称 | 涉及的用户对象 |
| User | 用户管理 | 系统管理员 |
| Activity | 活动管理 |
| Advice | 建议管理 |
| Article | 文章管理 |
| Message | 消息管理 |
| Moments | 朋友圈 |
| Group | 兴趣组 |
| System | 系统管理 |
| Auth | 用户认证\* |
| Index | 默认模块\* |

系统的前台模块开发包图如图3-2-2所示。后台模块开发包图如3-2-3所示。

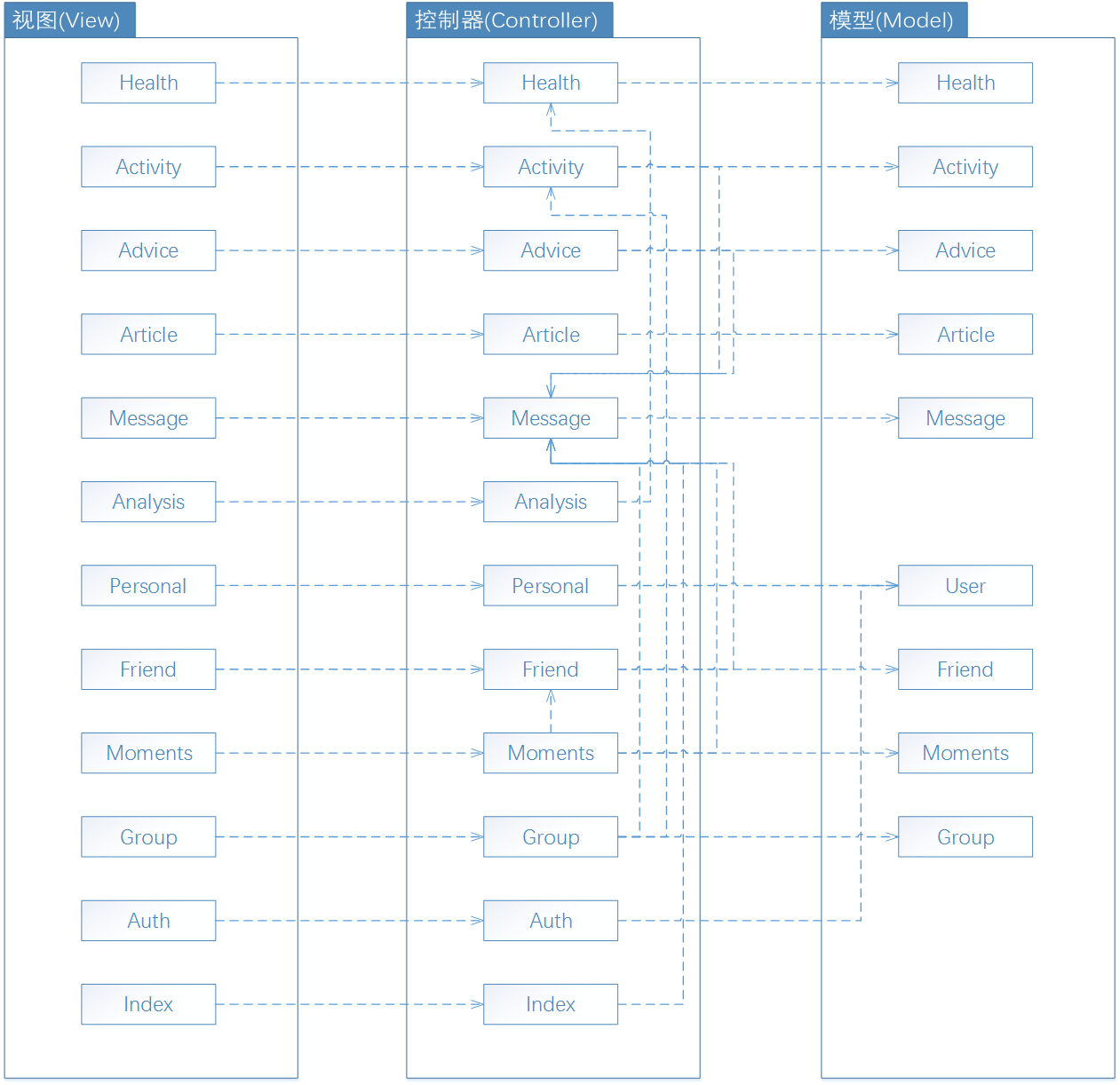


图3-2-2 前台模块开发包图

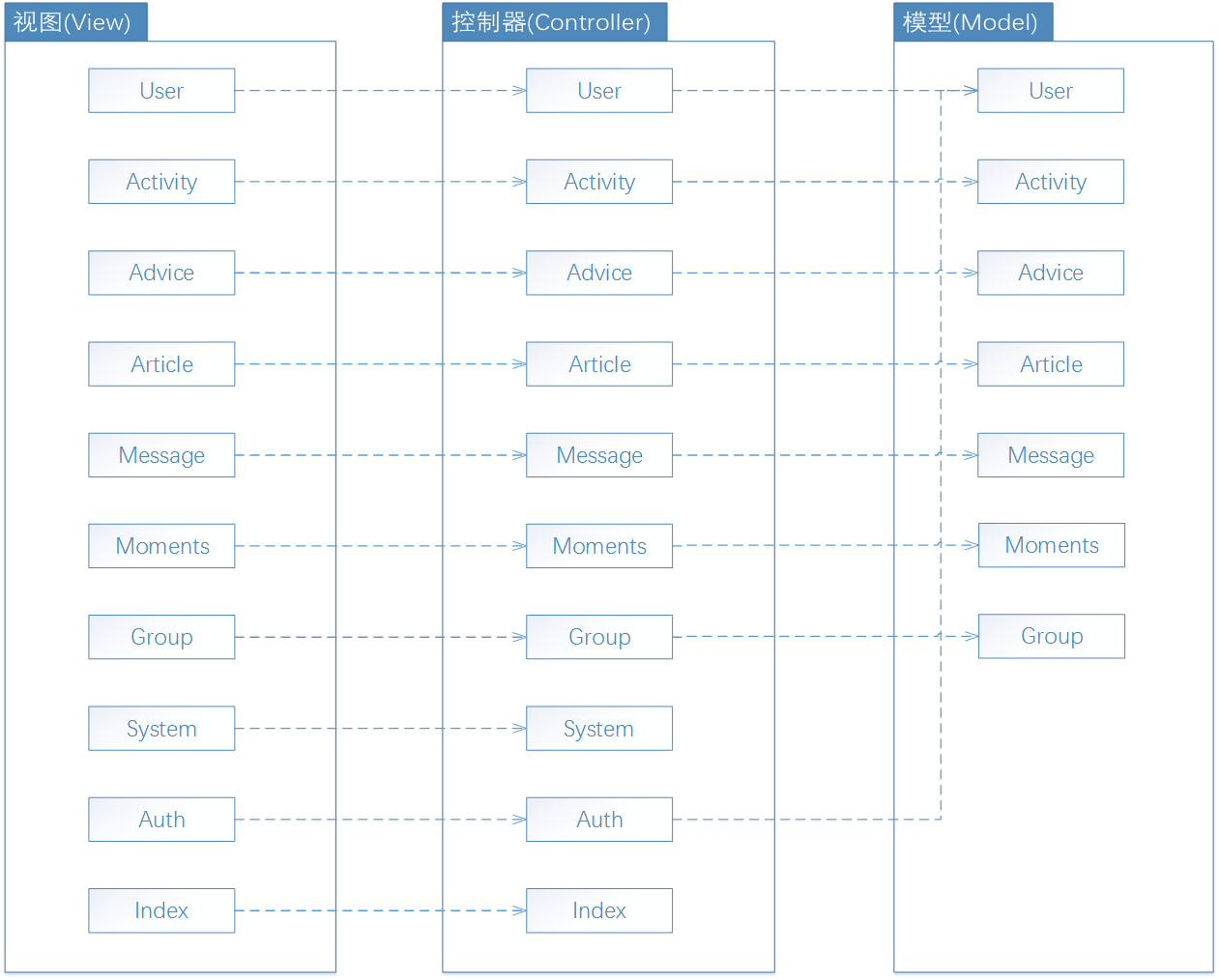


图3-2-3 后台模块开发包图

1. **数据库设计**

系统数据库设计的各表如下所示。

注：A\_I为Auto Increment（自增），PK为Primary Key（主键），FK为Foreign Key（外键）。

**user表 (用户表)：**

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 解释 |
| id | BIGINT, UNSIGNED, A\_I, PK | 用户ID |
| username | VARCHAR, UNIQUE | 用户名 |
| email | VARCHAR, UNIQUE | 邮箱 |
| phone\_number | VARCHAR, UNIQUE | 手机号 |
| password | VARCHAR | 密码 |
| real\_name | VARCHAR | 真实姓名 |
| nickname | VARCHAR | 昵称 |
| sex | TINYINT | 性别 |
| province | VARCHAR | 所在省份 |
| city | VARCHAR | 所在城市 |
| introduction | TEXT | 自我介绍 |
| interest | TEXT | 兴趣爱好 |

1. **界面原型与人机交互设计**
2. **详细设计**
3. **WEB性能优化**