FunnySports软件设计文档

苏琰梓

2016.10.31

文档更新记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 版本描述 | 日期 |
| V1.0 | 苏琰梓 | 完成文档 | 2016.10.31 |
|  |  |  |  |

目录

[1.引言 2](#_Toc466147944)

[1.1编制目的 2](#_Toc466147945)

[1.2词汇表 2](#_Toc466147946)

[1.3参考资料 2](#_Toc466147947)

[2.产品概述 2](#_Toc466147948)

[3.体系结构模型 2](#_Toc466147949)

[3.1整体架构描述 2](#_Toc466147950)

[3.2显示模块的分解 4](#_Toc466147951)

[3.2.1显示模块的职责 5](#_Toc466147952)

[3.2.2显示模块的接口规范 5](#_Toc466147953)

[3.2.3显示模块的设计原理 5](#_Toc466147954)

[3.3筛选模块的分解 6](#_Toc466147955)

[3.3.1筛选模块的职责 6](#_Toc466147956)

[3.3.2筛选模块的接口规范 6](#_Toc466147957)

[3.3.3筛选模块的设计原理 6](#_Toc466147958)

[3.4关注模块的分解 6](#_Toc466147959)

[3.4.1关注模块的职责 6](#_Toc466147960)

[3.4.2关注模块的接口规范 7](#_Toc466147961)

[3.4.3关注模块的设计原理 7](#_Toc466147962)

[3.5读取模块的分解 7](#_Toc466147963)

[3.5.1读取模块的职责 7](#_Toc466147964)

[3.5.1读取模块的接口规范 7](#_Toc466147965)

[3.5.2读取模块的设计原理 8](#_Toc466147966)

[3.6界面模块的设计 8](#_Toc466147967)

[3.6.1系统界面 8](#_Toc466147968)

[3.7运行时组件 9](#_Toc466147969)

# 1.引言

## 1.1编制目的

本报告给出FunnySports软件的详细设计描述和模型图，在软件架构的基础上进行了进一步分析，为系统的维护、扩展及后续开发提供参考依据。

## 1.2词汇表

|  |  |
| --- | --- |
| display sports data | 显示运动数据 |
| display activity list | 显示活动列表 |
| filter activities | 筛选活动 |
| follow / unfollow friend | 关注/取消关注好友 |

## 1.3参考资料

1）IEEE标准。

2）FunnySports软件需求规格说明文档。

# 2.产品概述

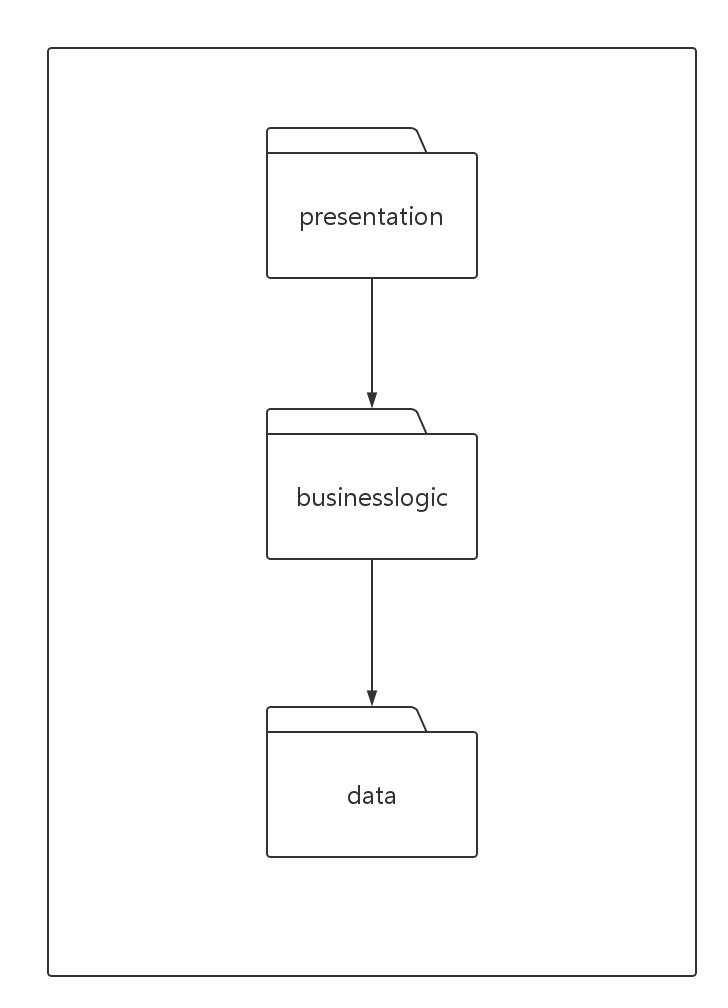
FunnySports系统通过基于可穿戴设备的数据采集，测试、收集、显示个人运动数据，并为用户提供历史运动数据的统计分析，供用户查看自己的运动数据、了解自己的运动水平。FunnySports支持用户线上交友，提供用户发布、参加活动、管理运动圈的功能。

本系统的目标是构建一个运动交友网站，使用户在了解自己运动数据的同时结交好友，从而更加热爱运动。

# 3.体系结构模型

## 3.1整体架构描述

为了更好地实现并行开发、提高开发速率，系统采用分层模型进行开发。传统分层模型包括表现层、业务层以及数据层3层。图1为传统分层模型图示。

图1 传统分层模型风格图

为了增强程序的灵活性和用户自由度，系统允许用户修改系统配置，为了更好地管理系统配置，特增加配置层以满足需求。图2为系统的分层架构。系统划分为以下4个层次

1. 表示层：用于前端界面展示，由html5+css构成
2. 业务层：包含业务控制和逻辑的层次，由php构成
3. 数据层：定义和存储系统中相关数据的层次，数据全部存储在sqlite数据库中
4. 配置层：管理界面配置及系统配置的层次

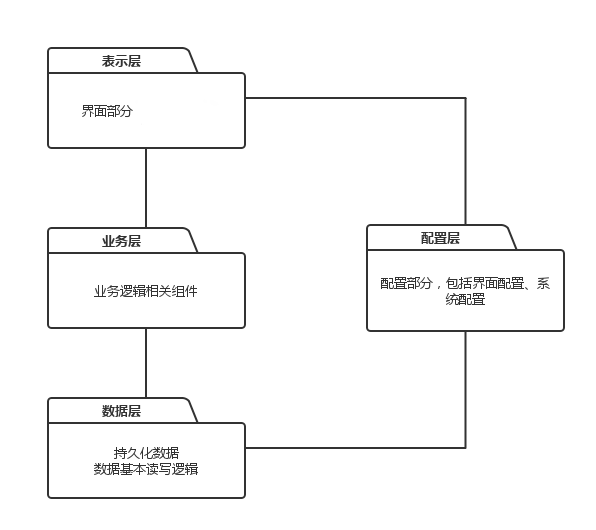


图2 系统分层架构

系统业务层分为以下四个模块，图3是业务层的分解示意图。

1. 显示模块：包括显示运动数据、显示活动列表和显示运动圈，负责实现与界面层的直接交互。
2. 读取模块：包括获取运动数据、活动信息以及运动圈数据，负责实现与数据层的直接交互。
3. 筛选模块：包括筛选活动。
4. 关注模块：包括关注好友和取消关注。

## 3.2显示模块的分解

显示模块分为运动数据、活动列表、运动圈三个模块，实现与界面层的直接交互，也是唯一与界面层交互的模块。

### 3.2.1显示模块的职责

显示运动数据负责显示用户各项运动数据的信息；

显示活动列表负责分类显示所有活动，并显示各项活动的具体内容；

显示运动圈负责显示好友列表和好友发布的话题。

### 3.2.2显示模块的接口规范

表1显示运动数据的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| DisplaySportsDataService.show | 语法 | public function getIndex() |
| 前置条件 | 已创建一个SportsData对象，已添加运动记录 |
| 后置条件 | 返回用户的运动信息 |
| DisplaySportsDataService.getSportsData | 语法 | public function getIndex($date) |
| 前置条件 | 已创建一个SportsData对象，已添加运动记录 |
| 后置条件 | 返回指定日期的用户运动信息 |
| 需要的接口（需接口） | | |
| 服务名 | | 服务 |
| SportsDataService.find() | | 返回所有运动数据 |
| SportsDataService.find($date) | | 根据日期返回运动数据 |

表2显示活动列表的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的接口（供接口） | | |
| DisplayActivityListService.  getActivities | 语法 | public function getActivities () |
| 前置条件 | 已创建一个ActivityList对象，已添加各项活动 |
| 后置条件 | 返回活动列表 |
| DisplayActivityListService.  getActivities | 语法 | public function getActivities($item) |
| 前置条件 | 已创建一个ActivityList对象，已添加各项活动 |
| 后置条件 | 返回指定标签的活动 |
| 需要的接口（需接口） | | |
| 服务名 | | 服务 |
| ActivityDataService.find() | | 返回活动列表 |
| ActivityDataService.find($item) | | 根据标签返回符合的活动 |

### 3.2.3显示模块的设计原理

显示模块可以显示运动数据、活动列表和运动圈，运动数据供用户了解自己的运动情况与健康信息，活动列表展现所有的活动，同时也提供进一步筛选和搜索活动，符合逐层递进的思维习惯，显示运动圈供用户线上交友以及管理自己的好友。

## 3.3筛选模块的分解

筛选模块主要是按标签筛选活动列表中的活动，并对结果进行排序显示。

### 3.3.1筛选模块的职责

筛选活动负责根据不同的标签对活动进行筛选，并对筛选结果进行排序。

### 3.3.2筛选模块的接口规范

表3 筛选活动的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| FilterActivityService.  getQualifiedActivity | 语法 | public function getQualifiedActivity ($item) | |
| 前置条件 | 用户从活动列表中输入筛选条件，启动一次活动筛选任务 | |
| 后置条件 | 系统选出所有符合用户输入筛选条件的活动 | |
| FilterActivityService.  sort | 语法 | public function sort($high,$low) | |
| 前置条件 | 系统筛选出所有符合条件的活动 | |
| 后置条件 | 所有被筛选出的活动按照符合条件的由大到小程度进行排序并显示 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| FilterActivityDataService.findQualifiedActivity ($item) | | | 根据筛选条件进行查找对应的所有活动对象 |
| ActivityDataService.find($item) | | | 根据标签返回符合的活动 |

### 3.3.3筛选模块的设计原理

为了增强用户体验，系统提供对活动列表的筛选操作。仅仅对活动列表进行筛选显然并不能很好地展现筛选效果，所以在显示结果前，对筛选结果进行排序。关于排序，参照经典文件资源管理器采用的方式，系统提供根据不同数据域进行排序，并可实现倒序。

## 3.4关注模块的分解

关注模块分为关注好友和取消关注，实现对用户好友的管理。

### 3.4.1关注模块的职责

关注好友及取消关注负责实现对好友列表的管理。

### 3.4.2关注模块的接口规范

表4 关注模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| **提供的服务（供接口）** | | | |
| SelfSelectFriendService.addSelfSelectFriend | 语法 | public function addSelfSelectFriend ($friendId) | |
| 前置条件 | 启动一次添加好友操作 | |
| 后置条件 | 将该用户加入好友列表中 | |
| SelfSelectFriendService.removeSelfSelectFriend | 语法 | public function removeSelfSelectFriend ($friendId) | |
| 前置条件 | 启动一次删除自选股操作 | |
| 后置条件 | 将该用户从好友列表中删除 | |
| **需要的服务（需接口）** | | | |
| 服务名 | | | 服务 |
| SelfSelectFriendDataService.addSelfSelectFriend ($userID,$friendID) | | | 根据用户账户加入用户的好友列表中 |
| SelfSelectFriendDataService.removeSelfSelectFriend ($userID,$friendID) | | | 根据用户账户从用户的好友列表中移除 |

### 3.4.3关注模块的设计原理

用户选择对感兴趣的人添加关注后才可以从好友列表中看到该好友，并可以在运动圈看到好友发布的信息，当用户取消关注后好友列表及运动圈都会清除非关注好友的信息。

## 3.5读取模块的分解

读取模块包括读取数据和解析数据。

### 3.5.1读取模块的职责

读取模块负责从数据库获取各项数据，实现与外界的交互。

### 3.5.1读取模块的接口规范

表5 读取模块的接口规范

|  |  |  |
| --- | --- | --- |
| **提供的服务（供接口）** | | |
| ReadActivityDataService.  getData（） | 语法 | public function getData() |
| 前置条件 | 无 |
| 后置条件 | 获取数据库中的信息 |

### 3.5.2读取模块的设计原理

读取模块实现系统与数据库的交互，负责读取数据。

## 3.6界面模块的设计

### 3.6.1系统界面

图4 首页界面



图5 运动数据界面



图 6活动列表界面



图7 运动圈界面



## 3.7运行时组件

无