

需求文档

Echo运动社交网站



生宸

141250107

目录

[1.引言 2](#_Toc464403829)

[1.1 目的 2](#_Toc464403830)

[1.2 范围 2](#_Toc464403831)

[1.3 参考文献 2](#_Toc464403832)

[2. 总体描述 2](#_Toc464403833)

[2.1 商品前景 2](#_Toc464403834)

[2.1.1 背景与机遇 2](#_Toc464403835)

[2.1.2 业务需求 2](#_Toc464403836)

[2.2 商品功能 3](#_Toc464403837)

[2.3 用户特征 3](#_Toc464403838)

[2.4 约束 3](#_Toc464403839)

[2.5 假设与依赖 3](#_Toc464403840)

[3.详细需求描述 4](#_Toc464403841)

[3.1 对外接口需求 4](#_Toc464403842)

[3.1.1 用户界面 4](#_Toc464403843)

[3.1.2 通信接口 4](#_Toc464403844)

[3.2 功能需求 4](#_Toc464403845)

[3.2.1 运动数据同步 4](#_Toc464403846)

[3.2.2 运动信息记录查看 5](#_Toc464403847)

[3.2.3 活动管理 5](#_Toc464403848)

[3.2.4 个人信息管理 8](#_Toc464403849)

[3.2.5 好友系统管理 8](#_Toc464403850)

[3.2.6 动态信息管理 9](#_Toc464403851)

[3.2.7运动数据统计分析 10](#_Toc464403852)

[3.2.8 用户举报 10](#_Toc464403853)

[3.2.9 举报信息处理 11](#_Toc464403854)

[3.2.10 用户权限管理 12](#_Toc464403855)

[3.3 非功能需求 12](#_Toc464403856)

[3.3.1 安全性 12](#_Toc464403857)

[3.3.2 可维护性 13](#_Toc464403858)

[3.3.3 易用性 13](#_Toc464403859)

[3.3.4 可靠性 13](#_Toc464403860)

[3.3.5 业务规则 13](#_Toc464403861)

[3.3.6 约束 13](#_Toc464403862)

[3.4 数据需求 13](#_Toc464403863)

[3.5 其他需求 14](#_Toc464403864)

# 1.引言

## 目的

本文档描述了Echo网站的功能需求和非功能需求。本文档说明的内容可能在项目实施过程中发生变更。

## 范围

Echo网站主要是针对那些希望有效的统计和监控自己跑步信息数据的用户，这些用户可以在本网站获取自己平时的运动信息以及相关的统一分析，用户还可以通过活动功能与身边的朋友等进行类似竞赛的项目，此外用户还能分享自己的运动情况的动态给好友。

## 参考文献

1. IEEE标准

# 总体描述

## 商品前景

### 背景与机遇

随着时代的不断发展，人们越来越注重健康对生活的影响，更多的人们会选择在日常中的闲暇时光来跑步或者锻炼以提供个人身体素质，有规律的、系统的、科学的健身方式更是人们追求的方面，所以需要一种能够解决此问题的软件，既能够帮助他们分析每天的运动情况，也能够帮助他们与朋友之间形成互相鼓励、互相监督的运动习惯，并且能够有效地指导运动时间、运动强度等。

Echo系统就是这样一个运动数据分析&社交网站，适合上述的用户使用，有很好的前景和价值意义。

### 业务需求

BR1：网站投入使用6个月后，总体的用户量能够超过150人

BR2：网站投入使用6个月后，用户回馈的好评率超过60%

BR3：网站投入使用6个月后，注册的用户使用本网站的人均频率超过0.8次/天

## 商品功能

SF1：运动数据同步

SF2：运动信息记录查看

SF3：活动管理

SF4：个人信息管理

SF5：好友系统管理

SF6：动态信息管理

SF7：运动数据统计分析

SF8：用户举报

SF9：举报信息处理

SF10：用户权限管理

## 用户特征

|  |  |
| --- | --- |
| 用户种类 | 特征 |
| 普通用户 | 本系统主要的用户，关注每天的运动情况与近阶段的运动变化情况，希望能借助本系统有效地分析运动数据，并且希望该系统能够提供一个平台供他们与自己的好友进行交流、比赛等 |
| 系统管理员 | 系统的维护人员，并且处理来自普通用户的举报信息，删除不正规的活动或者恶意使用该系统的用户，能监控所有的用户和活动的具体信息等内容 |

## 约束

CON1：所有角色使用图形用户界面与系统交互

CON2：在开发中，开发者要严格完成软件需求规格说明文档，详细设计文档

CON3：服务器采用php语言编写，使用Laravel框架，部署在Linux系统下

CON4：前端使用 HTML5+CSS+Javascript 采用Materialize的CSS库

CON5：数据库采用SQlite3，表设计要符合关系型数据库的基本要求

## 假设与依赖

AS1：用户拥有可穿戴设备(如运动手环)或者可以记录运动数据的移动设备

AS2：用户有可以连接至Internet的PC或者移动设备

# 3.详细需求描述

## 3.1 对外接口需求

### 3.1.1 用户界面

用户界面以网页的形式呈现给用户，要求符合Material Design

界面简洁，操作方便，颜色搭配合理，支持用户用鼠标和键盘进行操作，支持常规热键

界面要求要有导航栏，方便用户操作。

界面内要有一定数量的链接跳转，包括不同界面之间的跳转和同界面不同组件的跳转

### 3.1.2 通信接口

php服务器与SQlite3数据库的连接使用Laravel封装的连接方式

前端与服务器的数据访问采用Restful风格的数据通信，采用ajax发送http请求的方式

## 3.2 功能需求

### 3.2.1 运动数据同步

#### 3.2.1.1 特性描述

系统允许用户将可穿戴设备或者移动端设备中记录的运动信息同步到本系统中，以用于运动信息展示和统计数据分析

#### 3.2.1.2 刺激/响应序列

刺激：用户请求将运动数据同步到本系统中(默认为同步到当天为止的所有数据)

响应：系统将到当天为止的所有数据读取并存储

#### 3.2.1.3 相关功能需求

|  |  |
| --- | --- |
| DataSync.request | 系统允许用户进行运动数据更新 |
| DataSync.cancel | 系统允许用户取消运动数据更新操作 |
| DataSync.handle | 当用户在向系统发起数据更新同步请求后，系统开始同步并更新该用户的运动数据 |
| DataSync.finish | 当系统完成数据的更新后，会通知用户数据同步完成 |

### 3.2.2 运动信息记录查看

#### 3.2.2.1 特性描述

系统应该允许用户查看日常的运动信息记录信息，包括当天的运动数据详情以及近期的运动量变化信息情况

#### 3.2.2.2 刺激/响应序列

刺激：用户请求查看运动信息记录

响应：系统返回最近同步过的一天的运动数据详情给用户

刺激：用户选择具体的日期查看当日内的运动数据详情

响应：系统获取用户选择的日期内的数据详情并返回给用户

刺激：用户进行运动信息同步

响应：系统将当天以及之前的运动数据同步并储存，并且将当天的运动详情返回给用户

刺激：用户可以将日常的运动信息发布到好友圈内

响应：更新好友们的好友圈信息

#### 3.2.2.3 相关功能需求

|  |  |
| --- | --- |
| RecordView.request | 系统允许用户查看近期的运动信息记录 |
| RecordView.info.latest | 系统将用户最近同步过的运动数据详情返回给用户 |
| RecordView.info.datepick | 系统应该允许用户选择某个特定的日期 |
| RecordView.info.align | 系统根据用户选择的具体日期，将当天的运动数据详情返回给用户 |
| RecordView.sync | 系统应该允许用户在查看运动信息记录时进行运动数据的更新 |
| RecordView.sync.info | 当用户在请求同步运动数据之后，系统进行运动数据的更新与存储，更新结束之后将当天的运动数据详情返回给用户 |
| RecordView.share | 系统应该允许用户将当天的运动信息数据发布到好友圈 |
| RecordView.share.update | 在用户发布完运动信息动态之后，系统将好友圈内信息更新 |

### 3.2.3 活动管理

#### 3.2.3.1 特性描述

系统应该允许用户进行活动管理，其中包括参加其他用户创建的活动，查看活动详情信息，用户自己新建活动，用户编辑修改自己创建的活动的一些信息，用户查看已经结束的活动的结果，用户退出参与的活动

#### 3.2.3.2 刺激/响应序列

刺激：用户查看推荐的活动列表

响应：系统返回推荐给该用户的活动列表

刺激：用户选择活动的分类(比如活动的种类，活动的时间长度，活动的方式等)

响应：系统根据用户选择的分类将推荐的列表进行筛选并返回给用户活动列表

刺激：用户查看活动的详细信息

响应：系统将用户选定的活动的详情返回给用户

刺激：用户参加活动

响应：系统将用户添加至该活动的成员中，更新该活动的详情并且更新用户参与的活动

刺激：用户创建活动

响应：系统返回活动创建需要填写的信息项

刺激：用户填写活动的具体内容信息并发布

响应：系统更新用户参与的活动

刺激：用户查看自己报名参加的未结束的活动

响应：系统将该用户参加的未结束的活动列表返回给用户

刺激：用户查看报名参加的未结束的活动的成员列表

响应：系统将该活动的成员列表返回给用户

刺激：用户查看自己参加已经参加过的并且结束了的活动

响应：系统将该种类的活动列表返回给用户

刺激：用户查看自己参加过的并且已经结束了的活动的结果排名和奖励

响应：系统将该活动的结果信息和奖励信息返回给用户

刺激：用户查看自己创建的未开始的活动

响应：系统将该种类的活动列表返回给用户

刺激：用户编辑自己创建过的未开始的活动

响应：系统返回给用户信息项让其填写修改项内容

刺激：用户编辑活动内容并且提交

响应：系统将修改的内容通知给参与该活动的成员

刺激：用户查看自己参加的未开始的活动列表

响应：系统将该用户参加的未开的活动列表返回给用户

刺激：用户查看自己参加的未开始的活动的详情和成员

响应：系统将该活动的详情信息和成员列表返回给用户

刺激：用户退出参与活动

响应：系统更新用户参与的活动列表和该活动的详情信息

#### 3.2.3.3 相关功能需求

|  |  |
| --- | --- |
| Activity.recommand.request | 系统允许用户查看推荐活动列表 |
| Activity.recommand.list | 系统将针对该用户的推荐的活动列表返回给用户 |
| Activity.recommand.classify | 系统应该允许用户对推荐的活动列表进行分类查看 |
| Activity.recommand.rerank | 系统根据用户的分类选择将推荐的活动列表进行重排后将推荐活动的列表返回给用户 |
| Activity.join | 系统应该允许用户参加特定的活动 |
| Activity.join.update | 在用户选择参加活动后，系统需要更新该活动的详情列表(该活动的成员列表)以及该用户的活动列表信息 |
| Activity.create | 系统应该允许用户创建活动 |
| Activity.create.info | 在用户向系统提出创建项目的请求后，系统应该允许用户填写和选择活动的基本信息 |
| Activity.create.update | 在用户填写完成并提交活动基本信息后，系统应该更新用户的活动列表和存储的总体的活动列表信息 |
| Activity.view.ready | 系统应该允许用户查看该用户报名了的未结束的活动列表 |
| Activity.view.ready.list | 在用户向系统发出查看参加的活动列表的请求后，系统将用户报名了的活动列表返回给用户 |
| Activity.view.ready.info | 系统应该允许用户查看报名参加了的未结束的活动的详情，包括活动的基础信息和报名参加的成员列表 |
| Activity.view.complished | 系统应该允许用户查看该用户参加了的已经结束了的活动 |
| Activity.view.complished.list | 在用户请求查看自己参加了的已经结束了的活动后，系统应该返回该种种类的活动列表给用户 |
| Activity.view.complished.result | 系统应该允许用户查看已经结束了的活动的结果排名和奖励信息 |
| Activity.view.created | 系统应该允许用户查看自己创建过的活动列表 |
| Activity.view.created.list | 在用户向系统发送查看自己创建过的活动列表的请求后，系统返回该用户创建过的活动列表 |
| Activity.view.creatd.modify | 系统应该允许用户对自己创建过的未开始的活动进行基本信息的修改 |
| Activity.view.created.update | 在用户修改并且提交活动的修改项之后，系统更新活动信息并且将修改内容通知给所有参加了该活动的用户 |
| Activity.view.exit | 系统应该允许用户删除一些活动的参与记录或者退出某些活动 |

### 3.2.4 个人信息管理

#### 3.2.4.1 特性描述

系统应该允许用户查看个人信息并且允许用户修改一些基本信息

#### 3.2.4.2 刺激/响应序列

刺激：用户请求查看个人信息

响应：系统将该用户的个人信息返回

刺激：用户请求修改一些基本信息

响应：系统将个人基本信息的修改项返回给用户

刺激：用户填写修改项内容并保存提交

响应：系统根据用户提交的修改项更新用户个人信息

#### 3.2.4.3 相关功能需求

|  |  |
| --- | --- |
| PersonalInfo.view | 系统应该允许用户查看个人信息 |
| PersonalInfo.view.show | 在用户向系统请求查看个人信息后，系统应该返回个人信息列表给用户 |
| PersonalInfo.modify | 系统应该允许用户修改个人信息 |
| PersonalInfo.modify.content | 在用户向系统请求修改个人信息后，系统应该返回修改项给用户 |
| PersonalInfo.modify.update | 在用户保存并提交修改项后，系统更新该用户的个人信息 |

### 3.2.5 好友系统管理

#### 3.2.5.1 特性描述

系统应该允许用户查看自己的粉丝和关注列表，并且能够在站内搜索用户，可以对感兴趣的用户进行关注，也可以取消对指定用户的关注

#### 3.2.5.2 刺激/响应序列

刺激：用户请求查看关注列表

响应：系统将该用户的关注列表返回给用户

刺激：用户请求查看粉丝列表

响应：系统将该用户的粉丝列表返回给用户

刺激：用户请求搜索其他用户

响应：系统将搜索结果返回给用户

刺激：用户可以主动将未关注的其他用户添加到自己的关注列表中

响应：系统更新该用户的关注列表，更新被关注用户的粉丝列表

刺激：用户可以对已经关注了的用户取消关注

响应：系统更新该用户的关注列表和被取关用户的粉丝列表

#### 3.2.5.3 相关功能需求

|  |  |
| --- | --- |
| FriendManage.follow | 系统应该允许用户查看自己的关注列表 |
| FriendManage.follow.list | 在用户向系统发出查看关注列表请求后，系统应该将关注列表返回给用户 |
| FriendManage.follow.unfollow | 系统应该允许用户取消关注已经关注的用户 |
| FriendManage.follow.update | 在用户完成取消关注或者关注未关注用户后，系统应该更新该用户的关注列表 |
| FriendManage.follower | 系统应该允许用户查看自己的粉丝列表 |
| FriendManage.follower.list | 在用户向系统发送查看粉丝列表请求后，系统应该将该用户的粉丝列表返回给该用户 |
| FriendManage.search | 系统应该允许用户在站内进行搜索 |
| FriendManage.search.result | 在用户向系统完成搜索请求后，系统应该将搜索的结果列表返回给用户 |

### 3.2.6 动态信息管理

#### 3.2.6.1 特性描述

用户可以查看自己好友的动态信息，其中动态信息包括日常的运动信息分享(类似扇贝英语打卡)、好友最近参加的活动动态、好友最近达成的成就动态，并且可以对好友圈内的动态信息进行点赞/举报

#### 3.2.6.2 刺激/响应序列

刺激：用户可以查看个人的好友圈

响应：系统将该用户的所有好友的动态信息返还给该用户

刺激：用户可以针对某些动态信息进行点赞或者举报

响应：系统更新该条动态的详情信息

刺激：用户可以查看好友圈内所有用户的运动量排名

响应：系统将好友圈内的用户的运动量进行排名并将结果返还给用户

#### 3.2.6.3 相关功能需求

|  |  |
| --- | --- |
| Post.view | 系统应该允许用户查看自己的好友圈动态 |
| Post.view.list | 在用户请求完查看好友圈动态之后，系统应该将好友圈的信息列表返回给该用户 |
| Post.view.praise | 系统应该允许用户对某条动态进行点赞 |
| Post.view.report | 系统应该允许用户对某条动态进行举报 |
| Post.view.update | 在用户进行完点赞或举报之后，系统应该更新该条动态的详细信息 |
| Post.rank | 系统应该允许用户查看好友圈内的运动量排行 |
| Post.rank.result | 在用户向系统请求查看好友圈排名后，系统应该讲好友圈排名信息返还给用户 |

### 3.2.7运动数据统计分析

#### 3.2.7.1 特性描述

用户可以查看运动数据的详细统计分析

#### 3.2.7.2 刺激/响应序列

刺激：用户请求查看运动数据统计分析

响应：系统将该用户的详细的运动数据统计分析结果返还给用户

#### 3.2.7.3 相关功能需求

|  |  |
| --- | --- |
| StatResult.view | 系统应该允许用户查看自己的运动数据统计分析 |
| StatResult.view.info | 在用户向系统请求查看运动数据分析后，系统将该用户的运动数据详细信息返还给该用户 |

### 3.2.8 用户举报

#### 3.2.8.1 特性描述

用户可以对一些动态信息进行举报

#### 3.2.8.2 刺激/响应序列

刺激：用户举报动态信息

响应：系统更新该条动态信息的详情，并在管理员处理完成之后对动态信息进行其他操作

#### 3.2.8.3 相关功能需求

|  |  |
| --- | --- |
| Report.post | 用户可以举报动态信息 |
| Report.post.update | 在用户完成举报请求后，系统更新该条动态的详情 |

### 3.2.9 举报信息处理

#### 3.2.9.1 特性描述

系统管理员可以接受来自用户的举报，并且处理这些被举报的内容

#### 3.2.9.2 刺激/响应序列

刺激：管理员可以查看来自用户的举报通知

响应：系统将来自用户的举报信息返还给管理员

刺激：管理员可以查看举报通知所对应的动态信息的详情，查明情况后，可以选择将其删除

响应：系统将管理员选中的需要删除的动态列表内容删除

#### 3.2.9.3 相关功能需求

|  |  |
| --- | --- |
| ReportProcess.view | 系统应该允许管理员查看举报信息 |
| ReportProcess.view.list | 在管理员请求查看举报信息列表之后，系统将来自普通用户的举报信息返还给管理员 |
| ReportProcess.view.detail | 系统应该允许管理员根据举报信息查看被举报内容的详情 |
| ReportProcess.view.select | 在管理员查看完被举报内容的详情后，可以选择将要被删除的内容 |
| ReportProcess.view.delete | 在管理员选完将要被删除的动态信息列表后，系统将选中的信息删除 |

### 3.2.10 用户权限管理

#### 3.2.10.1 特性描述

根据不同等级的用户，系统开放的功能会有一些差别，具体表现在如下：是否有权限新建活动。参与活动并取得名次可以获得一定的站内货币奖励，达成成就可以获得一定的站内货币奖励，根据货币的持有数量可以将普通用户分等级，当站内货币达到一定数量之后，可以开放新建活动的权限，如果货币量没有达到上述要求，则不能够创建活动，只能够参与别人的活动

#### 3.2.10.2 刺激/响应序列

刺激：普通用户参与活动与取得名次

响应：系统根据用户获得的名次发放一定数量的站内货币作为奖励

刺激：普通用户在一定量的运动后达成运动成就

响应：系统根据用户达成的成就发放一定数量的站内货币作为奖励

刺激：用户新建活动

响应：系统会根据用户所持有的站内货币量进行区分，如果货币量达到规定标准则允许用户新建活动，否则不允许用户新建活动

#### 3.2.10.3 相关功能需求

|  |  |
| --- | --- |
| PowerManage.earn | 系统应该允许用户通过一定的途径获取站内货币 |
| PowerManage.earn.activity | 系统根据用户参加过的活动的名次向该用户发放一定数量的站内货币 |
| PowerManage.earn.acievement | 系统根据用户达成的成就向该用户发放一定数量的站内货币 |
| PowerManage.activity.create | 在用户创建活动的时候，系统会根据用户持有的站内货币数量判断用户是否有权限，若有权限，则允许创建 |
| PowerManage.activity.unable | 若用户在创建活动时货币没有达到规定数量，则系统不允许用户创建活动 |

## 3.3 非功能需求

### 3.3.1 安全性

Safety1：系统只允许经过验证和授权的用户访问

Safety2：系统内好友的定义是双方都互相关注的用户，单方面的关注不能开放好友圈功能

Safety3：系统内只有一个默认的管理员账号，该账号可以查看所有的普通用户和各种动态、活动等信息

Safety4：该系统不涉及真实的货币的交换，且虚拟的站内货币也不与真实货币挂钩，是一个财务安全的系统

### 3.3.2 可维护性

Modifiability1：当活动的种类发生改变后，系统需要在0.25个人月内完成

Modifiability2：当系统新增运动建议等方面的新功能后，系统需要在0.3个人月内完成

### 3.3.3 易用性

Usability1：主要的功能需求操作需要简单，用户可以在简单的摸索后掌握80%的功能操作

Usability2：尽量减少用户需要输入的地方，新建活动、修改活动内容等较为繁杂的业务过程也尽量采用选择项而非填空项的形式传达

### 3.3.4 可靠性

Reliability1：当20个以内的用户同时使用本系统时，系统不能崩溃

Reliability2：当系统崩溃后，需要在5分钟内快速重新启动

Reliability3：数据更新是服务器与数据库的数据交换过程，在用户进行数据同步后，数据同步发生在服务器端，用户断网不会影响到数据的更新与同步

### 3.3.5 业务规则

无

### 3.3.6 约束

Constriant1：使用本系统的用户默认有可以记录运动信息的可携带设备或者移动设备

## 3.4 数据需求

数据格式等定义会出现在设计文档中，在这里先不讨论

## 3.5 其他需求

无