跑Kemon-Go

软件需求规约

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <3/6/2019> | <1.0> | <初始版本> | <项目开发团队> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 定义、首字母缩写词和缩略语 4

1.3 参考资料 4

2. 整体说明 4

3. 具体需求 5

3.1 功能 5

3.1.1 <Use case 图> 5

3.1.2 <Use case 跑步 规约> 6

3.1.3 <Use case 捕捉精灵 规约> 6

3.1.4 <Use case 登录 规约> 7

3.1.4 <Use case 查看附近用户 规约> 7

3.1.5 <Use case 宠物对战 规约> 8

3.1.6 <Use case 查看好友信息 规约> 9

3.1.7 <Use case管理用户信息 规约> 9

3.1.8 <Use case 更改跑步规则 规约> 10

3.2 易用性 10

3.3 可靠性 10

3.4 性能 11

3.5 可支持性 11

3.6 设计约束 11

3.6.1 <设计约束一> 12

3.7 联机用户文档和帮助系统需求 12

3.8 接口 12

3.8.1 用户界面 12

3.8.2 硬件接口 12

3.8.3 软件接口 12

3.8.4 通信接口 12

3.9 适用的标准 12

软件需求规约 (简化版)

# 简介

## 目的

本文档的目的在于

1. 提供关于软件“跑Kemon-Go”的需求功能详述，作为用户和软件开发人员之间相互了解的基础。
2. 提供性能要求、初步设计和对用户影响的信息，作为软件人员进行软件结构设计和编码的基础
3. 作为软件总体测试的依据

## 定义、首字母缩写词和缩略语

AR：Augmented Reality，即增强现实。

app：application，特指手机上的应用软件。

非交互式功能：显示路线、以AR形式展现精灵形象

基本功能：用户登录、宠物养成、查看周围用户、好友功能、宠物对战

高级功能：更改跑步规则

## 参考资料

《软件工程原理》（作者：沈备军、陈雨亭、陈昊鹏 出版社：高等教育出版社 出版日期：2013年2月）

# 整体说明

## 产品总体效果

提供一个将跑步打卡软件和虚拟精灵养成游戏相结合的运动app，面向有硬性跑步指标以及有个人健身需求的在校大学生，用户可以在跑步的同时打卡领取道具，捕捉精灵，升级精灵，大大提升跑步的趣味性，使更多人积极地参与到每日的跑步任务中来。

## 产品功能

1. 用户登录：用户能以统一身份认证的方式登入本系统（例如：上海交通大学学生通过JACCOUNT登录）
2. 显示路线：登入系统，并开始跑步时，系统实时记录用户的跑步路线，并将其显示在app的地图上
3. 生成路径点：登入系统，并开始跑步时，系统自动生成虚拟精灵点位、道具补给点位，当用户经过对应点位时，将信息回传服务器
4. 以AR形式展现精灵形象：到达精灵点位，用摄像头对准点位即可看到虚拟宠物的AR形象，此时可操纵精灵球进行捕捉
5. 宠物养成：成功捕获精灵后，可通过道具或用户经验对宠物进行升级，道具通过补给点获取，用户经验通过每次跑步的距离和时长计算得到
6. 管理用户信息：服务器端对每位用户的历史跑步记录、虚拟宠物信息进行记录和管理
7. 更改跑步规则：服务器端可对跑步规则进行更改，可规定每次跑步的最大里程数、合法速度、必经点位以及路线范围
8. 查看周围用户：用户可以查看附近的用户，并添加为好友
9. 好友功能：用户可以查看好友的历史跑步记录以及虚拟宠物信息
10. 宠物对战：好友之间可以进行虚拟宠物对战
11. 其它：如论坛功能，用户可在论坛进行交流分享

## 用户特征

1. 本系统主要适用于大一大二、有学校布置的硬性跑步指标的在校大学生
2. 本系统也同时面向对自己有一定健身要求、同时希望在健身过程中获得娱乐享受的大学生以及社会其他人士

## 约束

1. 学校虽然实现了Wifi和4G信号的全面覆盖，但是在某些缺少基站的地方（如湖边）还是可能遇到信号断流的问题，因此系统需要本地处理跑步信息的能力，即使在离线状态下依然能保证一次跑步的正常完成，重新获得网络连接之后再上传跑步信息。
2. 本系统的设计初衷是增加校园跑步的趣味性，不宜使用户在跑步过程中做太多停留，否则容易打乱跑步节奏，达不到锻炼效果，因此捕捉精灵的过程要尽量耗时少，操作简单。

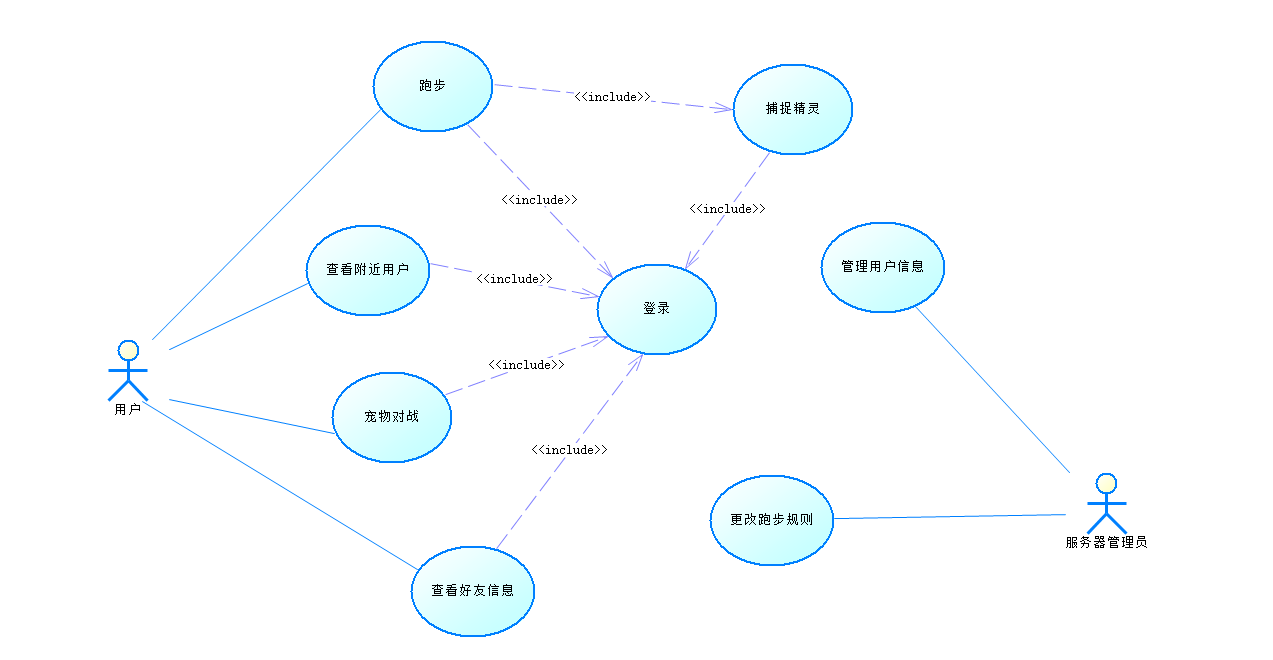
## 假设和依赖关系

AR技术是此次软件开发的一大亮点也是一大难点，鉴于参与开发的四位同学先前都无AR开发经验，此功能可能无法按时完成。假如AR开发过程遇到阻碍，我们会考虑先放掉这一功能，在日后有余力时再进行二次开发。

# 具体需求

## 功能

### <Use case 图>



·Actor有用户和服务器管理员

·用户的usecase包括跑步、查看附近用户、宠物对战、查看好友信息，这四个用例都依赖于登录，usecase“跑步”包含了“捕捉精灵”，因为捕捉精灵发生在跑步过程中

·服务器管理员只有两个用例：管理用户信息和更改跑步规则

### <Use case 查看宠物图鉴 规约>

|  |  |
| --- | --- |
| 用例名称 | 查看宠物图鉴 |
| 用例描述 | 用户在APP中点击“查看宠物图鉴”进入宠物页面 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录 |
| 后置条件 | 用户跳转至宠物界面 |
| 基本事件流程：   1. 点击查看宠物图鉴 2. 显示所有宠物 | |
| 候选事件流程：  2.1 用户点击只看已拥有,隐藏未拥有的宠物  2.2 用户点击查看所有,显示所有宠物 | |
| 备注 | 无 |

### <Use case 查看宠物信息 规约>

|  |  |
| --- | --- |
| 用例名称 | 查看宠物信息 |
| 用例描述 | 用户在APP中点击“查看宠物图鉴”进入宠物页面,再点击具体宠物查看宠物信息 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录 |
| 后置条件 | 用户跳转至具体宠物界面 |
| 基本事件流程：   1. 点击查看宠物图鉴 2. 点击具体宠物 3. 显示目标宠物信息 | |
| 候选事件流程：  无 | |
| 备注 | 无 |

### <Use case 添加好友 规约>

|  |  |
| --- | --- |
| 用例名称 | 添加好友 |
| 用例描述 | 用户在APP中点击“查看附近用户”后可以选着添加该用户为好友 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录 |
| 后置条件 | 目标用户同意 |
| 基本事件流程：   1. 点击查看附近所有用户 2. 点击用户头像 3. 点击添加他为好友 | |
| 候选事件流程：  无 | |
| 备注 | 无 |

### <Use case宠物训练 规约>

|  |  |
| --- | --- |
| 用例名称 | 宠物训练 |
| 用例描述 | 用户在APP中点击“训练宠物”后进行训练 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录 |
| 后置条件 | 宠物状态变更 |
| 基本事件流程：   1. 点击查看宠物图鉴 2. 点击已有宠物 3. 点击训练 4. 选择训练方式 | |
| 候选事件流程：   * 1. 选择运动训练,开始跑步   2. 选择消耗品训练,消耗消耗品,宠物经验状态改变 | |
| 备注 | 无 |

### <Use case 跑步 规约>

|  |  |
| --- | --- |
| 用例名称 | 跑步 |
| 用例描述 | 用户在APP中点击“开始跑步”后进行跑步 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录 |
| 后置条件 | 上传跑步信息 |
| 基本事件流程：   1. 点击开始跑步 2. 经过必经点位 3. 结束跑步 | |
| 候选事件流程：  1.1未点击“开始跑步”就开始，系统停留在第一步，且不记录任何跑步信息  2.1未经过必经点位就结束跑步，提示跑步无效，重新进入第二步  2.2经过必经点位，并且经过道具点位，用户获取道具  2.3经过必经点位，并且经过精灵点位，用户捕获精灵 | |
| 备注 | 无 |

### <Use case 捕捉精灵 规约>

|  |  |
| --- | --- |
| 用例名称 | 捕捉精灵 |
| 用例描述 | 用户在精灵点位打开摄像头，用精灵球捕捉精灵 |
| 主参与者 | 用户 |
| 前置条件 | 用户处于精灵点位，用户已点击“开始跑步”，且跑步有效 |
| 后置条件 | 本地存储精灵信息，在跑步结束后上传 |
| 基本事件流程：   1. 打开摄像头，对准精灵点位 2. 对准精灵扔出精灵球 3. 一定几率成功捕捉精灵 4. 结束 | |
| 候选事件流程：  1.1未对准精灵点位，精灵不出现，系统提示将摄像头对准精灵点位  2.1精灵球道具不够，系统提示缺少道具，并跳到第四步  3.1捕捉精灵失败，系统提示是否重新捕捉，回答是则跳到第2步，不是则跳到第4步 | |
| 备注 | 无 |

### <Use case 获得道具 规约>

|  |  |
| --- | --- |
| 用例名称 | 获得道具 |
| 用例描述 | 用户在精灵点位打开摄像头，点击道具即可获得 |
| 主参与者 | 用户 |
| 前置条件 | 用户处于道具点位，用户已点击“开始跑步”，且跑步有效 |
| 后置条件 | 本地存储道具信息，在跑步结束后上传 |
| 基本事件流程：   1. 打开摄像头，对准道具点位 2. 点击道具 3. 结束 | |
| 候选事件流程：  1.1未对准精灵点位，精灵不出现，系统提示将摄像头对准精灵点位  2.1精灵球道具不够，系统提示缺少道具，并跳到第四步  3.1捕捉精灵失败，系统提示是否重新捕捉，回答是则跳到第2步，不是则跳到第4步 | |
| 备注 | 无 |

### 3.1.8 <Use case 登录 规约>

|  |  |
| --- | --- |
| 用例名称 | 登录 |
| 用例描述 | 用户通过JAACOUNT登录系统 |
| 主参与者 | 用户 |
| 前置条件 | 无 |
| 后置条件 | 用户登录至系统 |
| 基本事件流程：   1. 点击登录 2. 跳转至统一身份认证界面 3. 输入账号密码 4. JAACOUNT登录成功，跳转回软件界面 5. 提示登录成功 | |
| 候选事件流程：  3.1输入账号密码错误，停留在第三步直到输入正确 | |
| 备注 | 无 |

### <Use case 查看附近用户 规约>

|  |  |
| --- | --- |
| 用例名称 | 查看附近用户 |
| 用例描述 | 用户可以在登录后随时查看附近用户 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录 |
| 后置条件 | 无 |
| 基本事件流程：   1. 点击“查看附近用户” 2. 上传用户地址至服务器，服务器回传附近用户信息 3. 查看用户信息 | |
| 候选事件流程：  3.1添加用户为好友 | |
| 备注 | 无 |

### <Use case 宠物对战 规约>

|  |  |
| --- | --- |
| 用例名称 | 宠物对战 |
| 用例描述 | 用户和好友进行宠物对战 |
| 主参与者 | 用户 |
| 前置条件 | 用户已登录，对战双方为好友 |
| 后置条件 | 胜负双方获得经验值 |
| 基本事件流程：   1. 选择好友 2. 点击开始对战 3. 选择出战精灵阵容 4. 对战胜利并结算 | |
| 候选事件流程：  3.1未选择出战阵容，系统停留在第三步并提示“请选择出战阵容”  4.1对战失败并结算 | |
| 备注 | 无 |

### <Use case 查看好友信息 规约>

|  |  |
| --- | --- |
| 用例名称 | 查看好友信息 |
| 用例描述 | 用户可查看好友的历史跑步信息和宠物信息 |
| 主参与者 | 用户 |
| 前置条件 | 用户查看对象已成为好友 |
| 后置条件 | 无 |
| 基本事件流程：   1. 选择好友 2. 点击查看信息，app显示好友的历史跑步信息 3. 切换至查看宠物信息，app显示好友的宠物和等级 4. 点击“退出查看”，系统返回 | |
| 候选事件流程：  3.1直接点击“退出查看”，系统返回  4.1不点击“退出查看”，系统一直停留在好友信息上 | |
| 备注 | 无 |

### <Use case管理用户信息 规约>

|  |  |
| --- | --- |
| 用例名称 | 管理用户信息 |
| 用例描述 | 管理员可查看和修改用户历史跑步记录和虚拟宠物信息 |
| 主参与者 | 服务器管理员 |
| 前置条件 | 无 |
| 后置条件 | 用户信息被保存 |
| 基本事件流程：   1. 点击某个用户，系统显示该用户的历史跑步记录和虚拟宠物信息 2. 点击修改 3. 更改历史跑步记录和虚拟宠物信息 4. 点击保存，用户信息完成修改 5. 系统退出并跳回到用户列表 | |
| 候选事件流程：  2.1不点击修改直接退出，跳转至第五步  4.1不点击保存直接退出，之前的修改全部无效 | |
| 备注 | 无 |

### <Use case 更改跑步规则 规约>

|  |  |
| --- | --- |
| 用例名称 | 更改跑步规则 |
| 用例描述 | 管理员可以更改规则，包括总里程、合法速度、必经点位、路线范围 |
| 主参与者 | 服务器管理员 |
| 前置条件 | 无 |
| 后置条件 | 新跑步规则应用至指定用户 |
| 基本事件流程：   1. 点击“修改跑步规则” 2. 更改总里程、合法速度、必经点位、路线范围 3. 点击“保存并应用至所有用户” 4. 退出系统 | |
| 候选事件流程：  3.1未点击保存就退出系统，此次修改全部无效 | |
| 备注 | 无 |

## 易用性

学生用户无需培训即可使用, 80%用户在初次接触该产品的数分钟后能掌握该产品的基本功能

管理员通过浏览器访问图形化界面, 80%管理员通过30分钟的培训后，能够掌握该产品的基本功能

## 可靠性

系统应在99%的时间处于正常可用状态, 90%的BUG修改时间不超过1个工作日，其他不超过2个工作日, 普通的软件缺陷修复可以在１～２天内完成

在GPS信号差的条件下（用户在跑步过程中GPS信号中断），能继续使用，记录大致的跑步路线

在服务器端提供数据备份功

在网络环境差的条件下（用户在跑步过程中中断网络连接），能在结束后数据同步，记录跑步过程

系统的GPS定位应精确到10米

最高错误或缺陷率:每千行代码错误容忍率为1%

错误或缺陷率:小错误容忍度为10个且至少在最近3次更新中修复,大错误容忍度为1个且必须在下次更新前修复,严重错误(用户数据丢失,系统无法启动,部分功能完全损坏等)容忍度为0个.

用户注册以及登录信息加密传输

管理员登陆服务器后端加密

## 性能

服务器后端采用负载均衡设计，能承受500个用户同时访问的流量

每秒可处理1000个事务

响应时间:

登录: 平均1秒 最长3秒

地图获取: 平均1秒 最长3秒

地图更新: 平均0.5秒 最长2秒

各种页面跳转: 平均0.2秒 最长1秒

宠物AR形象加载: 平均1秒 最长3秒

好友列表获取:平均1秒,最长3秒

好友信息获取: 平均0.5秒 最长1.5秒

对战界面加载:平均2秒 最长5秒

资源利用情况

内存:

客户端:100M

服务器端: 500M

磁盘:

客户端预计大小50M

服务器端预计大小300M

## 可支持性

手机客户端支持Android 7.0以及更高版本的操作系统

服务器后端支持Ubuntu 16.04以及更高版本的操作系统

服务器后端通过docker镜像快速部署，在十分钟内可以运行

## 设计约束

### 数据库设计及详细设计工具

Power Designer

Visio

### 编程语言

APP端：Android native

后端：Java

### 编程工具

JetBrains Android Studio

JetBrains IntelliJ IDEA

### 框架

APP前端：React native on Android

Web前端：React Web

后端：Spring Boot

ORM：Mybatis/hibernate

Gateway：Eureka/Zuul

前端管理框架：Gradle

后端管理框架：Maven

代码版本管理框架：Git/GitHub remote

后端微服务框架：Eureka/Kong

后端代码自动测试框架：Jenkins/circleci

### 数据库

Mysql

Mongodb

### 编程语言规范

[**Google Style Guides**](https://google.github.io/styleguide/)

## 联机用户文档和帮助系统需求

用户可以在线或离线阅读帮助文档，离线帮助文档在Wifi环境下能够自动与在线帮助文档同步。

80%的用户，阅读帮助文档的时间短于30分钟，可掌握基本功能及高级功能的完整使用方法。

## 接口

### 用户界面

#### 普通用户界面

登录界面：用户登录时所展现的界面。

跑步界面：使用跑步功能时所展现的界面。

用户地图界面：查找附近用户时所展现的界面。

宠物信息界面：查看宠物信息时所展现的界面。

宠物对战界面：宠物对战时所展现的界面。

好友信息界面：查看好友列表及好友信息时所展现的界面。

#### 管理员界面

用户管理界面：监视、变更用户状态时所展现的界面。

跑步规则界面：制定、应用、修改、废弃跑步规则时所展现的界面。

### 硬件接口

本软件无需硬件接口支持。

### 软件接口

与Jaccount系统：调用其认证接口。

与GPS系统：通过Android调用其定位服务接口。

### 通信接口

## 适用的标准