# **基于运动状态识别及记录的运动类安卓应用项目概要设计说明书**

小组成员：20152100087黎建辉，20152100059余炳明

，20152100117阳应杰，20152100137黄成龙

目录

[1引言 2](#_Toc24203)

[1.1编写目的 2](#_Toc29656)

[1.2项目背景 2](#_Toc3115)

[1.3定义 4](#_Toc2160)

[1.4参考资料 4](#_Toc6675)

[2总体设计 5](#_Toc3910)

[2.1需求规定 5](#_Toc14595)

[2.2运行环境 5](#_Toc18621)

[2.3基本设计概念和处理流程 5](#_Toc14320)

[2.4结构 6](#_Toc32675)

[2.5人工处理过程 7](#_Toc22359)

[2.6尚未解决的问题 7](#_Toc14276)

[3接口设计 7](#_Toc17421)

[3.1用户接口 7](#_Toc10744)

[3.1.1用户的登录界面 7](#_Toc23957)

[3.1.2用户查看历史运动数据 8](#_Toc16786)

[3.2外部接口 8](#_Toc19537)

[3.2.1用户运动的数据输入 8](#_Toc11871)

[3.2.2GPS的位置信息输入 8](#_Toc19973)

[3.2.3零点报告 8](#_Toc30617)

[3.2.4服务器端的数据读取和写入 8](#_Toc4982)

[3.3内部接口 9](#_Toc8060)

[3.3.1运动数据的处理 9](#_Toc8259)

[3.3.2 GPS的信息处理 9](#_Toc9302)

[3.3.3零点的数据清算处理 9](#_Toc26137)

[4运行设计 10](#_Toc21324)

[4.1运行模块组合 10](#_Toc24306)

[4.1.1系统结构图 10](#_Toc16945)

[4.1.2注册功能模块组合 10](#_Toc17933)

[4.1.3登录功能模块组合 10](#_Toc12573)

[4.1.4查看及修改用户信息功能模块组合 11](#_Toc19887)

[4.1.5运动识别分析功能模块组合 11](#_Toc22949)

[4.1.6分享数据功能模块组合 11](#_Toc24565)

[4.2运行控制 11](#_Toc8414)

[4.2.1注册功能控制 11](#_Toc11076)

[4.2.2登录功能控制 11](#_Toc2720)

[4.2.3查看及修改用户信息功能控制 11](#_Toc32530)

[4.2.4运动识别分析功能控制 12](#_Toc32665)

[4.2.5数据分享功能控制 12](#_Toc25630)

[4.3运行时间 12](#_Toc17709)

[4.3.1注册功能模块运行时间 12](#_Toc29529)

[4.3.2登录功能模块运行时间 12](#_Toc15678)

[4.3.3查看及修改用户信息功能模块 13](#_Toc18690)

[5系统数据结构设计 13](#_Toc30032)

[5.1逻辑结构设计要点 13](#_Toc7625)

[5.2物理结构设计要点 13](#_Toc1954)

[5.3数据结构与程序的关系 14](#_Toc7675)

[6系统出错处理设计 14](#_Toc9777)

[6.1出错信息 14](#_Toc5883)

[6.2补救措施 15](#_Toc9927)

[6.3系统维护设计 15](#_Toc5527)

# 1引言

## 1.1编写目的

本设计书是基于运动状态识别及记录的运动类安卓应用项目的研发概要设计，将项目开发进程中或者项目结束后提供给双方人员使用，同时也可以作为实施后期的维护人员使用。

## 1.2项目背景

随着平均生活水平的提高，人们越来越注重身体健康问题，而传统的运动健身市场，主要是集中于健身房中以及运动健身产品的购买消费。就前者来说，动辄上千元的健身卡消费门槛，对于许多普通人来说过于专业的健身设备以及场地的限制，传统的健身房市场的发展受到了较大的限制。

近些年来，智能手机的广泛普及和发展，使得运动健身与移动互联网有了结合的契机，再加上手机内置的众多不同的传感器和GPS功能，因而从14年开始，运动健身类应用开始大爆发，并最终催生出了现在占有市场的一系列的运动健身类的移动应用。

目前市场上的运动健身类APP主要分为三个大类：

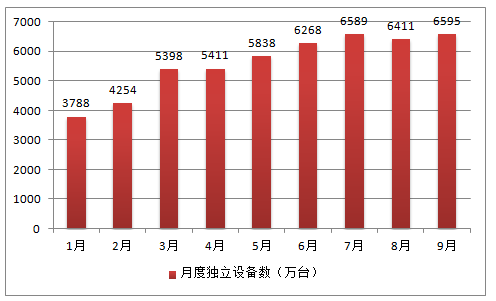
以跑步运动为主户外运动类APP，如：如咕咚、悦动圈等；

以身体塑型锻炼为主的室内锻炼类APP，如：Keep等；

单独锻炼某个部位的垂直锻炼应用，如：腹部运动等。

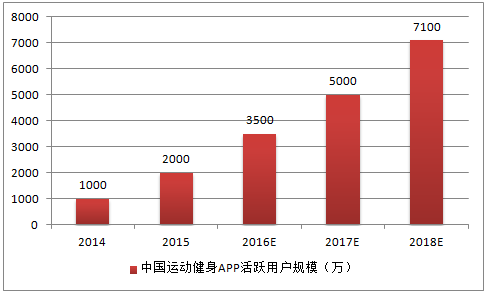
相对于传统的健身房健身来说，利用运动健身APP健身的门槛十分之低，无论是只想要每天进行一些简单锻炼的普通用户，还是身体素质较好，想要实现更好更专业的锻炼效果的健身爱好者，他们的需求都可以在不同的运动健身类APP上得到满足。低门槛的健身方式也是许多用户选择此类APP的重要原因。

从数据上来看，2016年1-9月，我国运动健身类APP的覆盖人数总体呈持续增长活跃的趋势，高达6500万的设备覆盖数，产业热度可见一斑。且今年的覆盖数量应该已经增长了许多，但是相关数据暂时还未完成统计。



而从月活跃用户数上来看，2016年运动健身类APP的月活跃用户数也已达到了3400万，且随之催生出了一系列的附加产业，甚至得以反馈到实体经济的诸如健身产品的销售上。

随着经济发展，人群亚健康的普遍存在以及雾霾等环境问题的出现，会有越来越多的人重视运动健康。2016年6月，国务院印发的《全民健身计划（2016—2020年）》指出，到2020年群众体育健身意识普遍增强，参加体育锻炼的人数明显增加，经常参加体育锻炼的人数达到4.35亿。运动人群数量增加的同时，人们对运动质量的要求也会越来越高，而运动类app可以通过对运动的记录和科学的计算，为用户提供最佳的运动体验。



从上面的图表可知，国内使用运动健身类app的用户逐年增加，显然未来使用运动健身类app的人也会越来越多。总而言之，运动健身类app的市场需求会越来越大。

## 1.3定义

分布式结构（MVC）——模型/视图/控制器结构（Model/View/Controller Architecture）

* 该结构是为同样的数据提供多个视图的应用程序而设计的，它将交互系统的组成分解成模型、视图、控制器三种部件。
* 视图是应用程序中用户界面相关的部分，即用户看到并与之交互的界面。
* 控制器工作就是根据用户的输入，控制用户界面数据显示和更新模型对象的状态。
* 模型是应用程序的主体部分，表示业务数据或者业务逻辑。
* 该结构适合于交互式系统，特别是同一个模型需要多个视图的情况。

## 1.4参考资料

《基于运动状态识别及记录的运动类安卓应用项目可行性分析报告》

《基于运动状态识别及记录的运动类安卓应用项目需求分析》

# 2总体设计

## 2.1需求规定

（1）系统能够在各种安卓手机上正常运行

（2）各组件在不同尺寸的手机屏幕上都能正常显示和运行

（3）步行、跑步、骑行的数据记录准确率达到87%

（4）自动识别运动状态的准确率达到75%

（5）绘制的轨迹准确率达到90%

（6）用户能够正常登陆并获取用户历史数据（通过微信或者QQ）

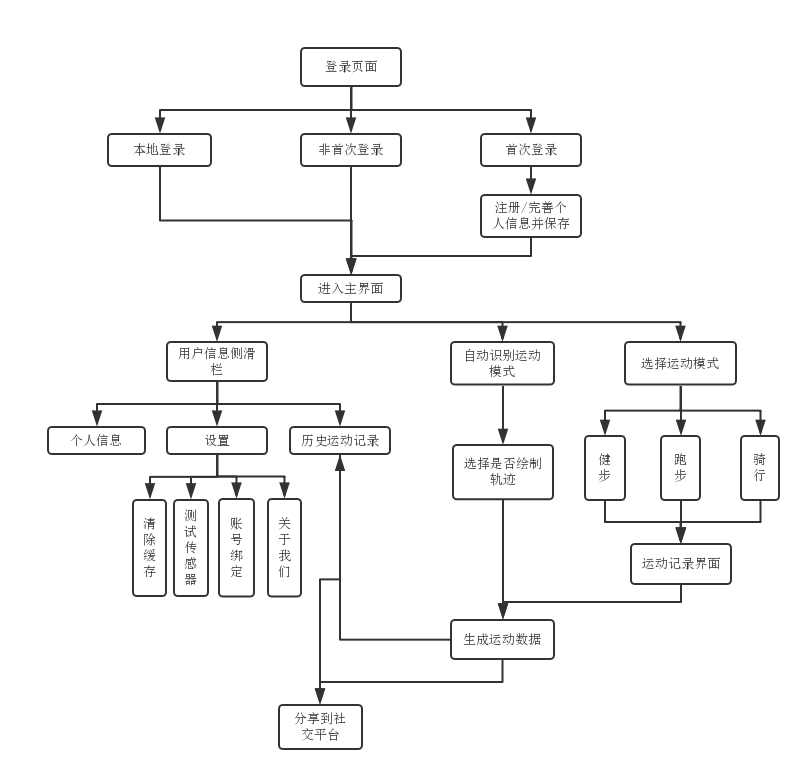
（7）服务端能够安全保存管理用户数据

## 2.2运行环境

* 操作系统：Android 5.0+系统
* 硬件要求：具备市面上智能手机常见的传感器

## 2.3基本设计概念和处理流程

系统控制流程如下图：



## 2.4结构

本项目采用分布式结构（MVC）



## 2.5人工处理过程

无

## 2.6尚未解决的问题

未确定功能需求与程序的关系

# 3接口设计

## 3.1用户接口

### 3.1.1用户的登录界面

输入：用户名、密码

输出：用户名与密码是否匹配的标识

功能描述：实现用户以账号的方式登陆（以便在不同的设备都能实现数据同步）

### 3.1.2用户查看历史运动数据

输入：时间

输出：与时间对应的详细运动数据

功能描述：实现用户查看最近的运动数据

## 3.2外部接口

### 3.2.1用户运动的数据输入

功能描述：输入传感器（主要是加速度传感器）的数据

### 3.2.2GPS的位置信息输入

功能描述：输入GPS的位置信息

### **3.2.3零点报告**

功能描述：让软件知道已经过了一天了要清算数据

### **3.2.4服务器端的数据读取和写入**

功能描述：读取与写入服务器中已注册账号的各种信息

**3.2.5通过微信和QQ的登录处理**

功能描述：通过微信和QQ作为软件的账号登录

**3.2.6手机的后台保活**

功能描述：通过与手机系统的内存管理交互，达到软件在后台也能工作的目的

## 3.3内部接口

### 3.3.1运动数据的处理

输入：传感器（主要是加速度传感器）的数据

输出：用户的运动状态和运动记录数据

功能描述：实现用户的运动状态识别及记录主要的运动数据

### 3.3.2 GPS的信息处理

输入：GPS的位置信息

输出：用户的运动轨迹

功能描述：实现绘制用户运动轨迹

### 3.3.3零点的数据清算处理

输入：零点标志

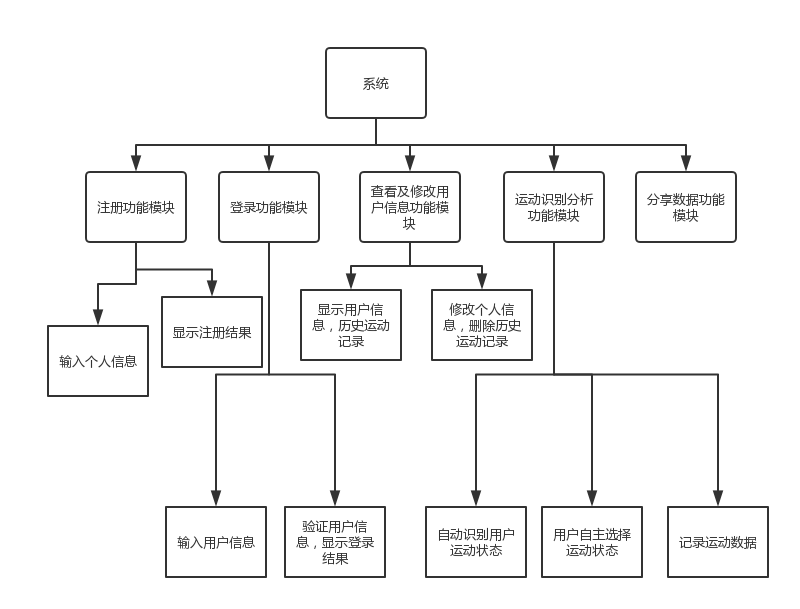
输出：数据清算处理的标识

功能描述：实现自动记录一天的运动数据

# 4运行设计

## 4.1运行模块组合

### 4.1.1系统结构图



### 4.1.2注册功能模块组合

（1）用户输入个人信息，输入用户名及密码功能模块组合

（2）系统显示用户注册成功，转向登录界面功能模块组合

### 4.1.3登录功能模块组合

（1）用户输入用户名，密码及验证用户信息功能模块组合

（2）系统显示登录成功，进入系统主界面功能组合

### 4.1.4查看及修改用户信息功能模块组合

（1）系统显示用户个人信息，历史运动记录功能模块组合

（2）用户修改个人账号信息，清除运动历史记录功能组合

### 4.1.5运动识别分析功能模块组合

（1）系统自动识别用户当前运动模式，记录运动数据，分析处理后显示功能模块组合

（2）用户自主选择运动模式，记录运动数据，分析处理后显示运动数据功能模块组合

（3）用户选择运动轨迹绘制，系统绘制轨迹并显示功能模块组合

### 4.1.6分享数据功能模块组合

（1）选择要分享到的社交平台，系统生成分享数据并链接相应社交平台功能模块组合

## 4.2运行控制

### 4.2.1注册功能控制

（1）用户输入注册信息，系统对其进行注册

（2）用户注册成功后，系统自动转到登录界面

### 4.2.2登录功能控制

（1）用户输入登录信息，系统对信息进行验证，系统核对后才能进行相关操作

（2）用户登录成功后，系统自动转到软件主界面

### 4.2.3查看及修改用户信息功能控制

（1）用户选择查看个人信息及历史运动记录，系统从服务器获得数据，并显示为个人信息和历史运动记录

（2）用户选择修改个人信息，系统显示个人信息，使之处于可编辑状态，并将修改后信息保存至服务器

（3）用户选择删除历史运动记录，系统弹出提示框再次询问，确定后删除服务器中相应的历史运动数据

### 4.2.4运动识别分析功能控制

（1）系统获取传感器数据并分析，自动识别用户当前运动状态，并将记录的传感器数据做相应处理，然后记录并显示

（2）用户自主选择正在进行步行、跑步还是骑行运动，系统将记录的传感器数据做相应处理，然后记录并显示

（3）系统弹出对话框询问是否进行轨迹绘制，用户选择是则检查GPS是否已经打开，若打开则开始记录轨迹并绘制在轨迹绘制界面上，若未打开则提示用户打开GPS；用户选择不进行轨迹绘制则跳过后续步骤

### 4.2.5数据分享功能控制

（1）用户选择要分享的社交平台，系统生成要分享的数据并对链接相应平台。

## 4.3运行时间

### 4.3.1注册功能模块运行时间

将会占用数据库资源较多的时间，当用户输入注册信息后，系统要将输入的注册信息与用户数据库中的信息进行比较，确定没有重复后，再将此注册信息加入数据库中。此过程需要遍历数据库，需要一定的时间占用。

### 4.3.2登录功能模块运行时间

将会占用数据库资源较多的时间，当用户输入登录信息后，系统要将输入的登录信息与用户数据库中的信息进行比较，找到对应的数据后，才会显示登录成功，并将相应的个人信息保存至本地，然后跳转到系统主界面。此过程除了需要遍历数据库，还要从服务器中下载数据，时间占用较多。

### 4.3.3查看及修改用户信息功能模块

查看个人信息时只用显示已保存至本地的信息即可，占用时间较少。查看历史运动数据则需要从服务器中调取相应数据并显示，需要占用数据库一定的时间。修改个人信息和删除历史运动数据，则需要上传数据至服务器，需要占用数据库一定的时间。

**4.3.4运动识别分析功能运行时间**

开启自动识别运动状态或者手动选择正在进行的运动，都将在后台持续读取传感器数据，并占用手机一部分的计算资源，这是本系统最大的时间占用。同时，前者占用的资源和时间都会稍多于后者

开启轨迹绘制后，会通过GPS持续获得位置数据及对这些数据进行处理，获得数据将一直占用网络传输资源，数据处理会占用较多的计算资源，这些占用是不可忽略的。

# 5系统数据结构设计

## 5.1逻辑结构设计要点

* 用户信息表（user）
* 用户（用户编号，用户账号，密码，头像，性别）
* 运动历史详表(exercise\_history)
* 运动历史(用户编号，具体日期，运动时长，消耗能量，健走步数，骑行里程)
* 运动日记表(exercise\_diary)
* 运动日记(用户编号，具体日期，运动记录)

## 5.2物理结构设计要点

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 用户信息表（User） | | |  |  |  |
| 序号 | 字段名 | 字段含义 | 类型 | 字节数 | 允许空 |
| 1 | id | 用户唯一编号 | int | 4 | 否 |
| 2 | name | 用户账号名 | varchar(20) | 20 | 否 |
| 3 | password | 登录密码 | varchar(15) | 15 | 否 |
| 4 | head\_img | 头像链接地址 | varchar(n) |  | 是 |
| 5 | sex | 性别 | char(2) | 2 | 否 |
|  |  |  |  |  |  |
| 运动历史详表(exercise\_history) | | |  |  |  |
| 序号 | 字段名 | 字段含义 | 类型 | 字节数 | 允许空 |
| 1 | id | 用户唯一编号 | int | 4 | 否 |
| 2 | date | 具体日期 | date | 3 | 否 |
| 3 | duration | 运动时长/min | int | 4 | 是 |
| 4 | coms\_en | 消耗能量/千卡 | int | 4 | 是 |
| 5 | step\_cnt | 健走步数 | int | 4 | 是 |
| 6 | ride\_cnt | 骑行里程/km | int | 4 | 是 |
|  |  |  |  |  |  |
| 运动日记表(exercise\_diary) | | |  |  |  |
| 序号 | 字段名 | 字段含义 | 类型 | 字节数 | 允许空 |
| 1 | id | 用户唯一编号 | int | 4 | 否 |
| 2 | date | 具体日期 | date | 3 | 否 |
| 3 | record\_des | 运动记录描述 | varchar(40) | 40 | 是 |

## 5.3数据结构与程序的关系

数据结构为关系型数据库，所以在程序中可以用标准的SQL语句与数据结构进行交互，交互过程中采用通用的数据反问接口。为了保持良好的程序架构，对数据库访问采用DAO设计模式实现，提高维护性和扩张性。

# 6系统出错处理设计

## 6.1出错信息

1. 数据加载慢可能导致系统崩溃：
2. 现今手机客户端普遍存在着数据加载慢于web页面的问题。如果加载速度慢于一定的界限值，可能会导致系统出现死锁、堵塞甚至崩溃的问题。
3. 服务器严重的业务和工作负载会导致停顿事故：
4. 服务器过小或工作范围过小可能会导致服务器负载过限，从而造成服务器工作停顿甚至崩溃。
5. 闪退：
6. 内存管理错误造成的闪退。内存过低，APP运行所需内存超过设备的限制，导致APP跑不起来；程序逻辑错误。数组越界、堆栈溢出、并发操作、逻辑出差等问题导致的APP闪退；设备兼容。由于设备多样性，APP在一些设备上可能存在着不兼容的现象。
7. 网络错误：
8. APP在运行环境中出现Http错误和其他网络错误，会造成应用页面无法显示的情况，从而影响用户的体验。

## 6.2补救措施

1. 针对数据加载慢所做出的补救措施：优化程序代码，尽量去除不必要的渲染效果（动画之类），追求应用简洁而不失美感，提高应用的流畅性和实用性。
2. 针对服务器停顿所做出的补救措施：由于是学生开发，可能我们所使用的服务器不能满足过于繁多的数据，所以这方面我们可能考虑前期限制客户的使用量，如果反馈效果良好，那么我们可能会考虑后续维护和运营，到时候我们将会有一定的资金去使用更为强大的服务器。
3. 针对闪退所做出的补救措施：设置提醒功能。当用户设备内存过低时给予提醒或限制使用；优化程序，避免程序逻辑出现错误；增加APP测试量，实时优化代码，提高APP在众多设备中的兼容性。
4. 针对网络错误所做出的补救措施：严格遵守文明上网、合法上网的原则，规范网络的安全性，避免网络病毒的侵染。同时优化代码，减小APP对网络流量的需求，提高APP的实用性。

## 6.3系统维护设计

1. 改正性维护：
2. 软件交付使用后，必然会有一部分隐藏的错误被带到运行阶段，在某些特定的使用环境下就会暴露，为了识别和纠正软件错误、改正软件性能上的缺陷、排除实施中的误使用，应进行的诊断和纠正的过程，就叫做改正性维护。
3. 适应性维护：随着计算机的飞速发展，外部环境或数据环境可能发生改变，为了使软件适应这种变化，而去修改软件的过程就叫做适应性维护。
4. 完善性维护：
5. 在软件使用的过程中，用户往往会对软件提出新的功能与性能要求。为了满足这些要求，需要修改或再开发软件，以扩充软件功能、增强软件性能、改进加工效率、提高软件的可维护性。这种情况下进行的维护活动叫做完善性维护。
6. 预防性维护：
7. 除了以上三类维护之外，还有一种维护活动，叫做预防性维护。这是为了提高软件的可维护性、可靠性等，为以后进一步改进软件打下了良好的基础。