## 1 相关技术

#### 应用开发基本条件

编程语言： Objective-C

程序环境： 移动客户端 iPhone 5s 以上(包含 iPhone 5s)

后台服务器 Bmob云服务器

数据库 SQLite 3 及 MySQL

开发工具 移动客户端 Xcode 8.0 、iOS SDK 及 SQLite 3

后台服务器 Bmob SDK 及 MySQL

#### 相关技术介绍

* + 1. MVC 架构介绍

MVC(Model View Controller，即模型-视图-控制器的缩写，如图1-1 所示)，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。MVC 被独特的发展起来用于映射传统的输入、处理和输出功能在一个逻辑的图形化用户界面的结构中。

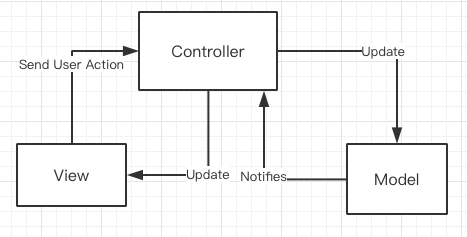


图1-1 MVC 架构示意图

* + 1. Objective-C 介绍

Objective-C，通常写作ObjC和较少用的Objective C或Obj-C，以 SmallTalk-80 语言为基础，扩充C的面向对象编程语言。它主要使用于Mac OS Objective-C，是在C的基础上，加入面向对象特性扩充而成的编程语言。目前，Objective-C主要应用于Mac OS X和[iOS](http://lib.csdn.net/base/ios" \t "_blank" \o "iOS知识库)这两个NeXTSTEP的衍生系统，而在NeXTSTEP和OpenStep中它更是基本语言。Objective-C可以在任何gcc支持的平台上进行编译，因为gcc原生支持Objective-C。

* + 1. iOS 开发介绍

iOS是由苹果公司为iPhone等设备开发的操作体系。它主要是给iPhone、iPod touch和iPad运用。就像其根据的Mac OS X操作体系相同，它也是以Darwin为根底的。iOS的体系架构分为四个层次：中心操作体系层（the Core OS layer），中心效劳层（the Core Services layer），媒体层（the Media layer），可轻触层（the Cocoa Touch layer）。如图1-2 所示。

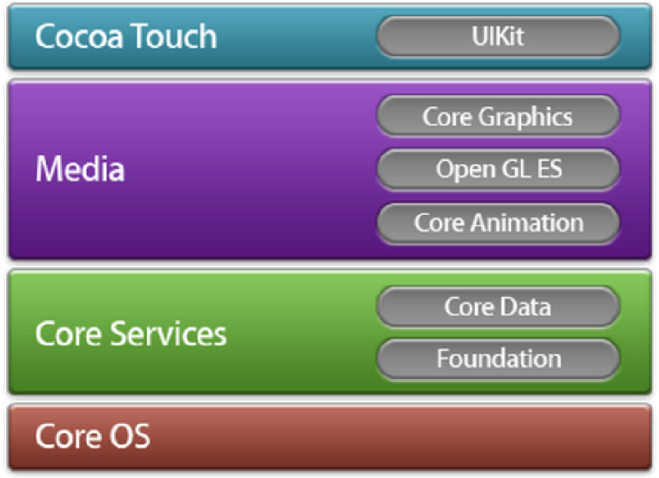


图 1-2 iOS 的体系架构

1. UIKit

所有的iOS应用程序都基于UIKit，你不能是应用程序脱离这个框架。UIKit提供了在屏幕上绘制的机制，捕获事件，和创建通用用户界面元素。UIKit也通过管理显示在屏幕上的组件来组织复杂的项目。UIKit 可以构建和管理你的用户界面，捕获触摸和基于移动的事件，呈现文字和 Web 内容，优化你的多任务程序，创建定制的用户界面元素。

1. Core Graphics

当你需要创建复杂的图形时，Core Graphics则提供了更底层的库来帮助你。使用 Core Graphics 可以创建基于路径的绘图，抗锯齿渲染，添加梯度、图片和颜色，创建、显示和分析 PDF 文档。

1. Core Animation

UIKit提供建立在Core Animation之上的动画。如果你需要比UIKit能力更高级的功能，可以直接使用Core Animation。Core Animation接口包含在Quartz Core框架里。使用Core Animation可以创建嵌套的对象，并且可以对它们操作、旋转、缩放和转换。使用Core animation，你可以创建动态的用户界面而不用使用更底层的图形API。

1. Core Data

  Core Data提供对象的管理，使用Core Data，你可以创建模型对象，并管理这些对象。你管理这这些对象间的联系并修改数据。Core Data提供的内建SQLlite技术可以高效的管理数据。

1. Foundation

UIKit和其它的框架都是建立在Foundation框架上面的。Foundation框架是用Object-C对Core Foundation框架里许多特性的封装。使用 Foundation 可以创建和管理集合，比如字典和数组，创建和管理字符串，提交和接收通知，创建日期和时间对象，执行异步代码等。

* + 1. SQLite 3 介绍

SQLite 3 是一款轻型的数据库，是遵守 ACID 的关系型数据库管理系统，它包含在一个相对小的 C 库中。它的设计目标是嵌入式的，而且目前已经在很多嵌入式产品中使用了它，占用资源非常低，可能只需要几百K的内存就够了。支持 iOS 系统，常用于在本地中存储大量的缓存数据。

* + 1. MySQL 介绍

MySQL是一种开放源代码的关系型数据库管理系统（RDBMS），MySQL数据库使用最常用的数据库管理语言—结构化查询语言（SQL）进行数据库管理。

#### 相关名词介绍

* + 1. SDK

国内外流行的社交平台，如：新浪微博、QQ空间、人人网、Facebook等，都开发了自己的开放平台，并为第三方应用接入准备了SDK，即软件开发工具包。使用SDK可以极大的简化接入这些社交平台的流程，并可以方便地使用他们所提供的API，从而丰富自己的APP。本项目中利用了一些SDK实现了社交平台的内容发布。

## 2. 需求分析

目前，iPhone 的使用人数不断增加，这部分用户中，想必也有许多喜欢锻炼的用户。RunFit 为喜欢锻炼的用户提供了计步、运动轨迹记录、数据图表展示、历史数据查看、运动数据分享、云端数据同步等一系列功能。给用户提供一款相对完善的iOS端的计步软件。

#### 2.1 可行性分析

2.1.1 技术可行性

a. 移动客户端

iPhone 手机上内置了运动协处理器，iOS 平台还提供了一系列框架用来访问相应的协处理器，配合相应的计算公式以及缓存机制，就可以得到用户的步数、卡路里消耗、运动时间、运动公里数等。配合相应的图表绘制、数据存储、网络请求链接远程数据库的编程，在技术上足以实现预期的功能。

b. 服务器

采用Bmob云服务器，提供了可视化的数据操作界面，提供一套完整的后台服务，以及轻量级的SDK，能够以最小的配置和最简单的方式使用对应的服务，而且后台由专门人员维护，开发者不必编写和维护服务器，而且是完全免费的，没有经济负担。

c. 数据库

在手机本地上使用 SQLite 存储用户的每日数据以及历史数据，由于这些数据并不是非常复杂，而且SQLite在本地所占用的资源比较小，所以采用SQLite即可满足需求。服务器方面同样存储的数据不是特别的复杂，但是存储的数据量相对较大，所以在服务器方面采用MySQL可以满足需求。

2.1.2 经济可行性分析

对于应用的开发和测试，仅需要PC一台、iPhone手机一部，且编译、部署应用的开发软件环境可以采用面向个人开发者的免费版，成本较低，经济可行性较强。

2.1.3 操作可行性分析

iOS系统对用户的交互做的很好，操作简单方便，按照Apple官方的设计要求以及参考一些成熟应用的开发经验，在用户交互方面尽量降低操作难度和用户学习使用App的时间，提供良好的用户体验和较强的操作可行性。

2.1.4 法律可行性

本应用所涉及的内容均不违反国家法律，在用户隐私方面考虑的比较全面，不会侵犯他人或者群体利益，不涉及版权侵犯问题，在法律层面上是完全可行的。

#### 2.2 客户端需求分析

2.2.1 计步及数据展示模块

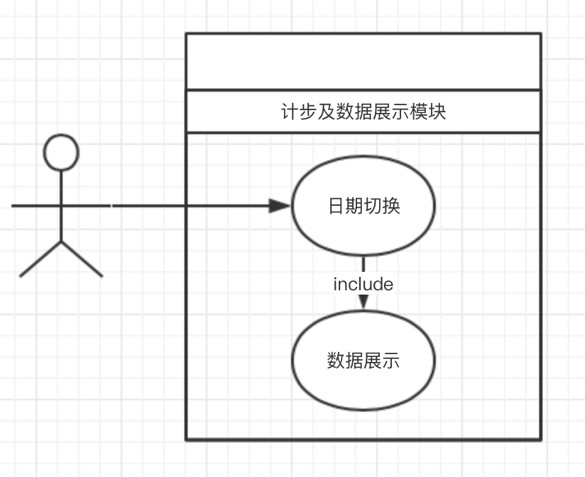


图2-1 计步及数据展示模块用例图

1. 数据展示

该部分作为主界面中步数、卡路里、活跃时间、行走的公里数的数据展示，还有一天内每个时段的步数数据的柱形图展示。根据日期展示对应的数据，数据存储于本地的SQLite数据库中。

1. 日期切换

利用系统提供的日期类实现了一个简单的日历，用户可以通过选择日期，查看对应日期的步数、卡路里消耗等数据。在用户点击对应的日期时，使用相应的日期对数据库进行检索，检索到数据之后，对数据展示界面的数据进行更新。

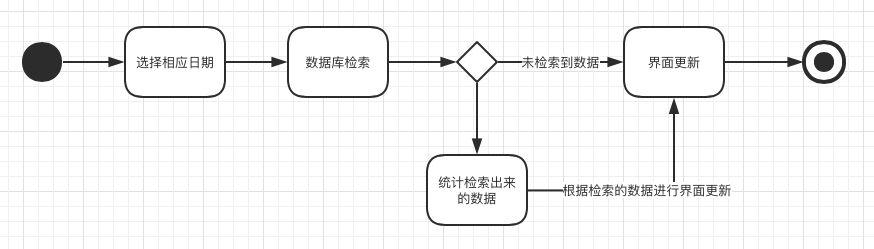


图2-2 日期切换活动图

2.2.2 跑步/骑行轨迹绘制模块

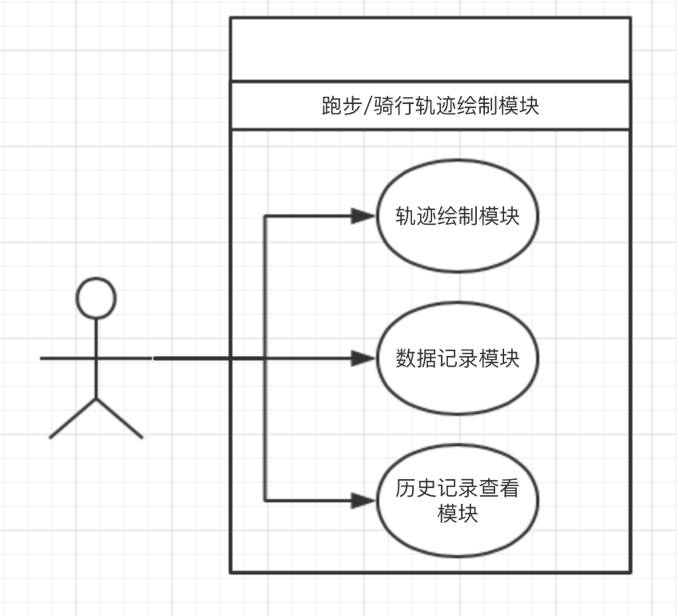


图2-3 跑步/骑行轨迹绘制模块用例图

1. 轨迹绘制模块

利用系统提供的SDK，在用户开始运动后，实时定位用户当前所在的位置，并将数据暂时存储在内存中，同时在地图上绘制出运动的轨迹。当用户结束运动时，可以选择是否保存数据。若保存，则数据存储在本地数据库中，若没有，则放弃当前记录的数据。

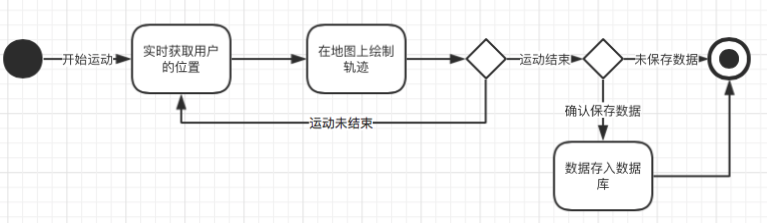


图2-4 轨迹绘制活动图

1. 数据记录模块

用户在运动期间，会在后台记录其运动的时间、速度、步数以及公里数。同时暂时保存用户的位置信息。用户运动期间可以很清楚地知道当前运动的时间、运动时的速度等数据。

1. 历史记录查看模块

该模块中会列出用户曾经确认保存数据的运动信息。用户可以在这个模块中看到过去的运动情况以及当时的一些运动数据等。在这个模块中，用户可以选择删除某次运动记录。

2.2.3 数据图表展示模块

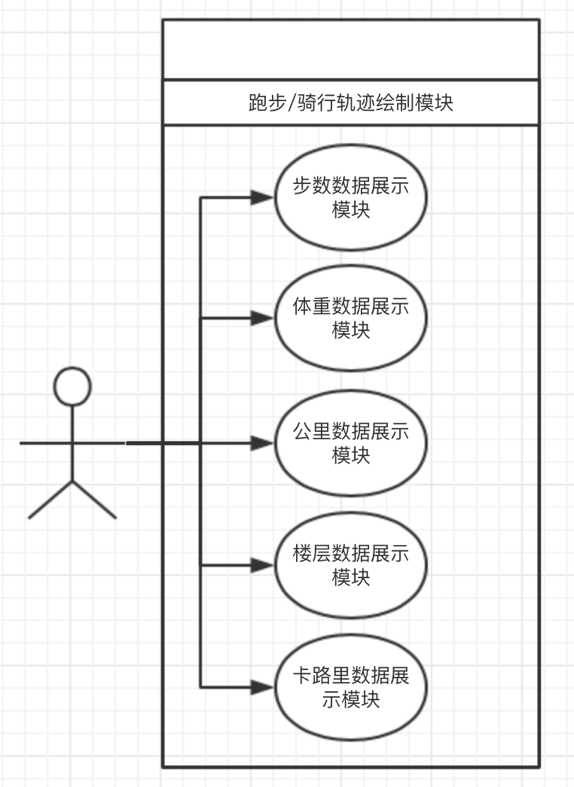


图2-5 跑步/骑行轨迹绘制模块用例图

1. 步数、公里、楼层模块

这三个模块的数据展示方式是相同的。即在这个模块中，会将数据库中所存储的对应的步数、公里、楼层数据进行统计，分为日、周、月、年各个时段，以柱型图的形式展现给用户，同时也提供了显示所有数据的功能，可以在这里面查看到每一天，对应模块的数据情况。

1. 体重、卡路里模块

这两个模块的数据展示方式也是相同的，但是不同于步数、公里、楼层模块。在此模块中，同样会将数据库中对应的数据统计，同样分为日、周、月、年各个时段，不同的是这两个模块中的数据是以折线图的形式展现给用户的。以折线图的形式更为直观的显示出体重、卡路里这类有对比性数据的每一个时段的数据，更为直观的看到其每个时段的差异性。

2.2.4 个人信息模块

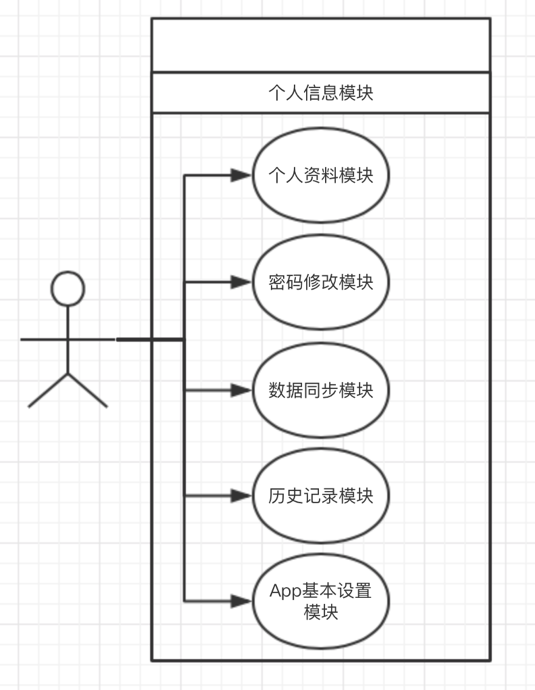


图2-6 个人信息模块用例图

1. 个人资料模块

为用户提供了修改用户头像、用户昵称、性别、体重、身高及步数目标这些基本信息的功能。这里的数据修改之后直接保存至后台数据库中，无需再次同步。

1. 密码修改模块

为用户提供了根据旧密码修改新密码的功能。在用户输入旧密码，两次新密码的输入，点击修改之后，会与后台交互并进行密码修改的操作，若修改失败，则提示修改失败并提示用户检查网络情况。在密码输入时会有密码格式检测，若检测出密码格式错误，则不会与后台交互并直接提示用户。

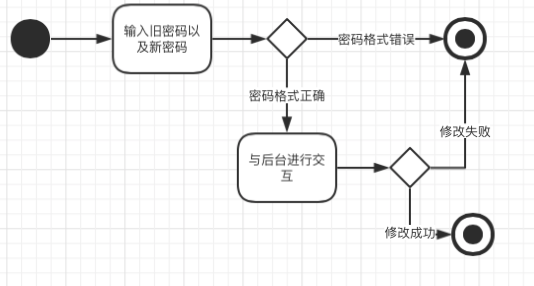


图2-7 修改密码活动图

1. 数据同步模块

用户可以通过数据同步模块将本地的数据同步到远程数据库中，也可以将远程数据库中的数据同步到本地的数据库中，以保证数据不会丢失。

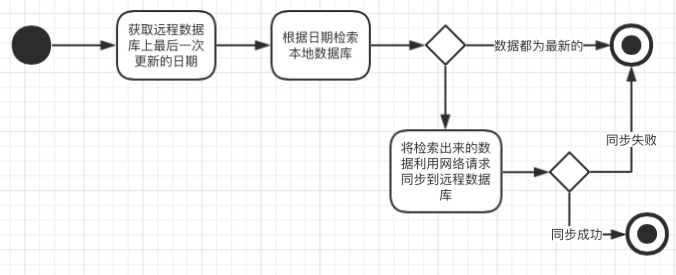


图2-8 本地数据库同步至远程数据库活动图

1. 历史记录模块

该模块中会列出用户保存数据的运动信息、体重信息等数据。用户可以在这个模块中看到过去的一些信息等。在这个模块中，用户可以选择删除某次数据记录。

1. App基本设置模块

包含了帮助、意见反馈、分享三个模块。帮助模块中列出了一些常见的问题以及解答，也提供了解答渠道，方便用户快速上手应用。意见反馈利用系统的邮件SDK，通过调用邮件来进行相应的问题反馈。分享提供了分享本APP的功能，如果觉得APP的功能还不错，可以分享推荐给好友使用。

2.2.5 登陆模块

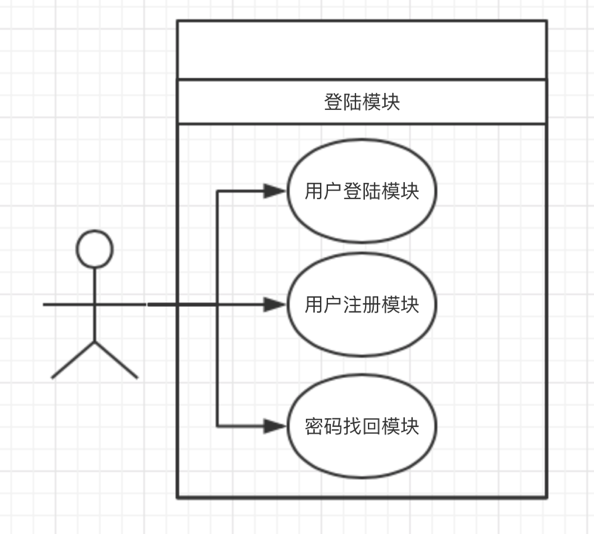


图2-9 登陆模块用例图

1. 用户登陆模块

用户利用已有的账号密码登陆APP。若账号不存在或者密码错误会有相应的错误提示反馈给用户。

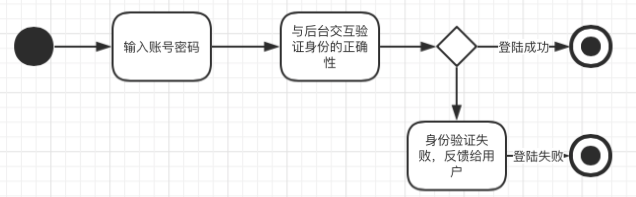


图2-10 用户登陆活动图

1. 用户注册模块

若用户第一次使用该应用，则需要注册一个账号。用户注册使用手机号码作为账号，通过手机验证码验证手机的正确性，然后用户设置密码。

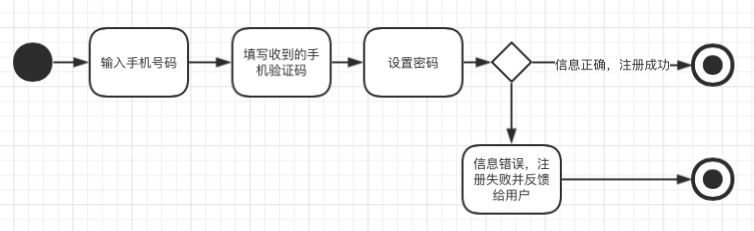


图2-11 用户注册活动图

1. 密码找回模块

若用户忘记密码导致登陆不上，可以通过该模块紧急修改密码。用户输入手机号并填写收到的手机验证码，重新设置密码。过程与注册相似。

2.2.6 数据分享模块

数据分享模块提供了将相应界面截图并分享到对应的App中，如QQ、微信、微博等。根据用户选择的App，将对应的图片发送到用户选择的App中。

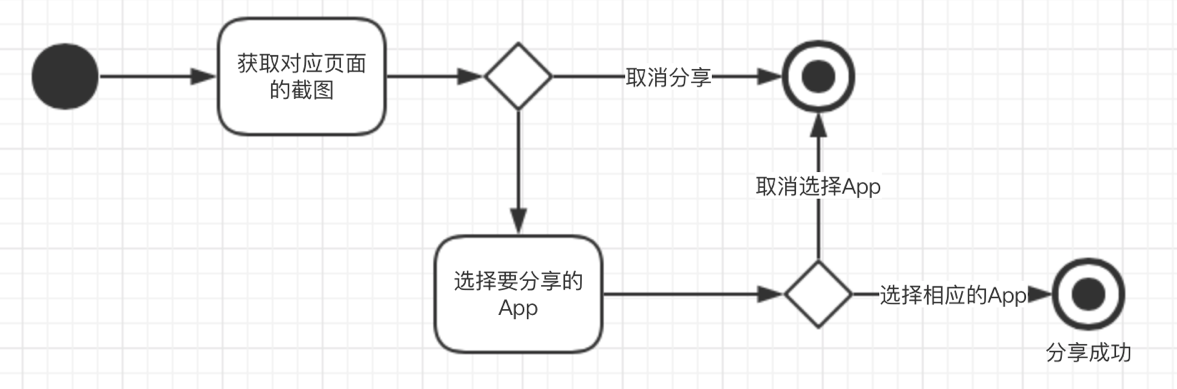


图2-6 数据分享状态图

#### 2.3 服务器端需求

2.3.1 用户注册接口

|  |
| --- |
| - (void)setUsername:(NSString \*)username; // 设置用户名 |
| -(void)setPassword:(NSString \*)password; // 设置密码 |
| -(void)setObject:(id)obj forKey:(id)key; // 根据字段设置对应的值 |
| -(void)signUpInBackgroundWithBlock:^(BOOL isSuccessful, NSError \*error)block;  // 注册用户，回调是否成功和错误信息 |

2.3.2 用户登陆接口

|  |
| --- |
| + (void)loginWithUsernameInBackground:(NSString \*)username  password:(NSString \*)password  block:(BmobUserResultBlock)block;  // 根据账号密码登陆，回调登陆是否成功信息 |

2.3.3 修改密码接口

|  |
| --- |
| - (void)updateCurrentUserPasswordWithOldPassword:(NSString \*)oldPass  newPassword:(NSString \*)newPass  block:^(BmobUser \*user, NSError \*error)block;  // 根据旧密码和新密码修改密码，回调修改之后的user，和错误信息 |

2.3.4 上传数据接口

|  |
| --- |
| - (void)saveInBackground:(BmobBooleanResultBlock)block  withProgressBlock:(BmobProgressBlock)progressBlock;  //异步进行数据传输，并回调传输的进度 |

#### 2.4 数据库需求

2.4.1 本地数据库

用户信息表：主键ID，时间，步数，卡路里，活跃时间，公里数，楼层，体重

用户历史记录表：主键ID，类型，时间，数值，时间，卡路里，速度，步数，位置坐标

2.4.2 远程数据库

用户表：主键ID，用户昵称，密码，手机号码，体重，目标，身高，头像

用户信息表：主键ID，时间，步数，卡路里，活跃时间，公里数，楼层，体重

用户历史记录表：主键ID，类型，时间，数值，活跃时间，卡路里，速度，步数，位置坐标

## 3. 项目设计

#### 3.1 系统总体设计

移动客户端根据用户每天的运动信息，记录用户的数据，并可选择地与远程进行同步，结合了iOS SDK，实现了计步、运动轨迹记录、数据图表展示、历史数据查看、运动数据分享、云端数据同步等基本功能。用户登陆、注册、数据同步需要请求后台服务器，服务器根据请求访问数据库，再将数据返回给客户端，客户端根据返回的数据进行解析，存入本地数据库中，同时在客户端上进行展示。

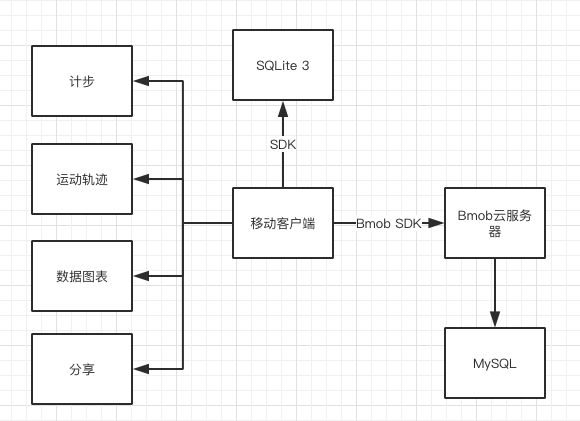


图3-1 系统总体结构图

#### 3.2 数据库设计

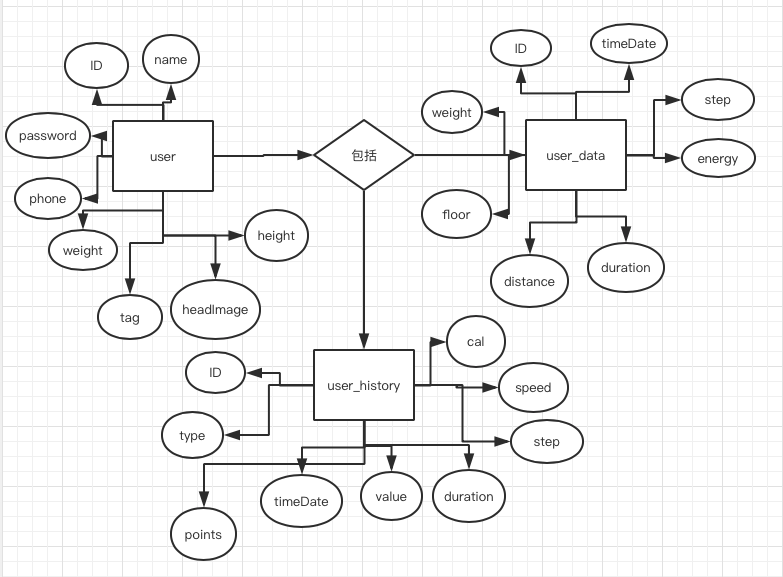


图3-2 数据库设计 E-R 图

#### 3.3 iOS 客户端设计

3.3.1 逻辑流程

在客户端上，大部分功能都是应用iOS SDK所给的API进行的，如计步功能的实现、运动轨迹的绘制、用户位置的获取等。与后台的交互采用的是Bmob SDK所提供的API，由API内自动进行服务器的请求。整体流程如图3-3所示。

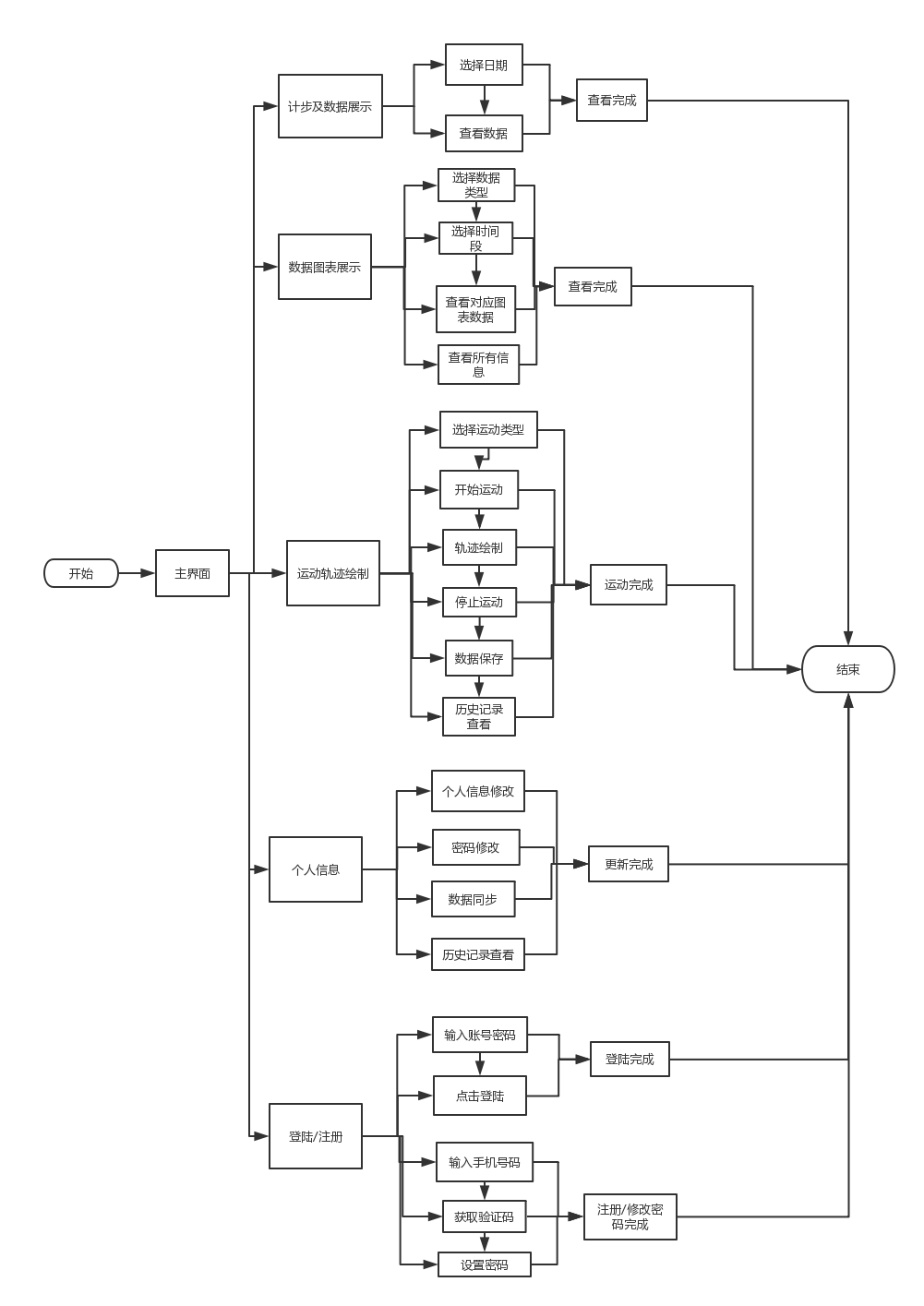


图3-3 逻辑流程图

3.3.2 设计方案

a. 计步及数据展示模块

含数据展示、日期切换、数据分享，共计3个功能。

1. 数据展示

首次进入APP时，通过协处理器及iOS系统提供的API获取七天内的数据保存至SQLite数据库中。并将当天的数据在界面上展示，展示内容分别为步数、卡路里消耗、活跃时间、步行公里数、每个时段的步数。

1. 日期切换

APP中编写了一个简单的日历，通过对日历的日期点击，获取到点击的日期，并根据这个日期前往SQLite数据库检索对应日期的数据，并使用检索到的数据更新数据展示界面。

1. 数据分享

通过对数据展示界面的截屏，得到图片数据，弹出选择应用的界面，根据用户所选择的应用，将数据传输到相应的APP中。该功能使用系统提供的应用间通信的API。

b. 数据图表展示模块

通过当天的时间，在数据库中检索以当天为标准的一天、一周、一月、一年内的数据，并使用相应的数据图表展示出来。

此模块包含步数、体重、公里数、楼层数、卡路里数，共计5个数据。提供一天内、一周内、一月内、一年内的数据的变化趋势，并使用图表展示出来。

图表的实现使用的是iOS系统中自带的绘图API，根据相应的数据绘制柱形图或者折线图。

c. 运动轨迹绘制

此模块包含轨迹绘制、数据记录、历史记录，共计3个功能。

1. 轨迹绘制

轨迹绘制使用的是iOS系统自带的地图应用中给出的轨迹绘制API，利用定位API获取到用户的位置改变坐标，在内存中缓存，并将这个坐标集合交给轨迹绘制API，系统便会根据坐标集合绘制相应的轨迹。

1. 数据记录

数据记录与计步类似，在后台开启一个计时器进行时间的记录，同时在运动过程中获取协处理器中的速度、步数等数据，并记录在内存中，待最后的数据保存时使用。

1. 历史记录

通过读取SQLite中历史记录表的数据，将过去保存的数据读取出来并进行展示。用户可以有选择性地进行删除。

1. 个人信息模块

此模块包含个人信息修改、密码修改、数据同步、历史记录、设置，共计5个功能。

1. 个人信息修改

用户可以在此模块中修改相应的信息并点击保存，通过Bmob SDK提供的数据更新API，将相应的数据与Bmob云服务器交互，更新云服务器里面的数据。可修改的信息包括用户头像、昵称、身高、体重、目标、性别，共计6个信息。

1. 密码修改

用户可以根据旧密码，来修改对应的密码。根据提供的API，进行相应的密码更新，若网络没有问题，则密码会对应修改成功。

1. 数据同步

此模块中，通过获取云服务器上最后一次更新的时间，将对应的数据通过提供的API存储到云服务器或者从云服务器上获取保存在本地的数据库中。在同步过程中提供同步进度供用户查看。

1. 历史记录

此模块同运动轨迹中的历史记录。同样是从本地SQLite中获取相应的数据并展示。

1. 设置

设置模块中包含帮助、意见反馈、分享，共计3个功能。

1. 帮助

提供常见的问题及帮助联系方式，用户可以根据帮助中的常见问题以及帮助联系方式来快速上手APP。

1. 意见反馈

通过系统提供的邮件API，通过邮件来进行反馈的提交。

1. 登陆注册模块

此模块包括登陆、注册、忘记密码，共计3个功能。

1. 登陆

采用手机号作为账号，根据用户输入的账号密码，首先检查格式输入是否错误，若没有错误，则采用Bmob SDK 提供的登陆API进行账号登陆。

1. 注册

根据手机号，来获取短信验证码，用户输入验证码并设置密码后，先检查输入格式是否错误，若没有错误，则采用Bmob SDK提供的注册API进行注册，并将手机号码作为用户的账号。

1. 忘记密码

操作过程同注册，只不过使用的API为根据手机号和短信验证码修改密码的API。

## 4. 项目实现

#### 4.1 系统实现

第一步：设计对应APP的UI界面，界面符合Apple公司的设计规范。

第二步：利用Bmob的可视化界面创建数据库以及设置对应的字段。

第三步：编写完善移动客户端的项目需求，并完成与后台的数据交互。

#### 4.2 数据库实现

#### 4.3 iOS客户端实现

第一步：根据UI设计图，完成各个页面的设计，并加入一些动画效果。

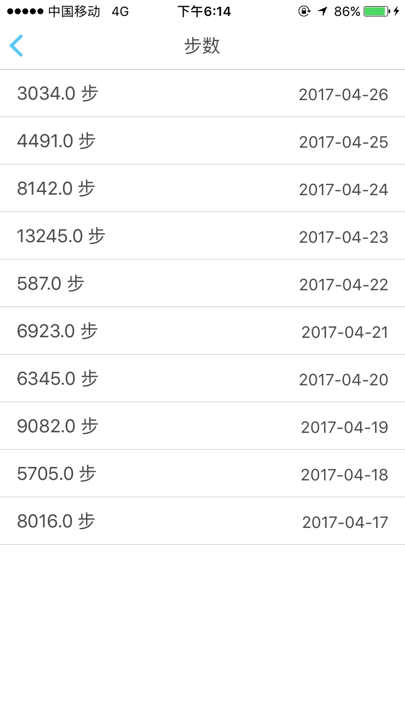
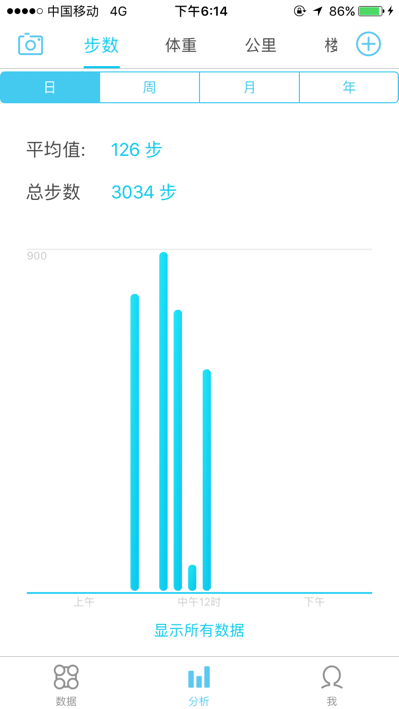
第二步：理清逻辑，根据各模块的需求设计，进行界面逻辑跳转及功能实现。

第三步：完成相应的数据交互代码，并进行测试。

4.3.1 计步及数据展示模块



4.3.2 数据图表展示模块



4.3.3 运动轨迹绘制模块





4.3.4 个人信息模块

(1) 主界面



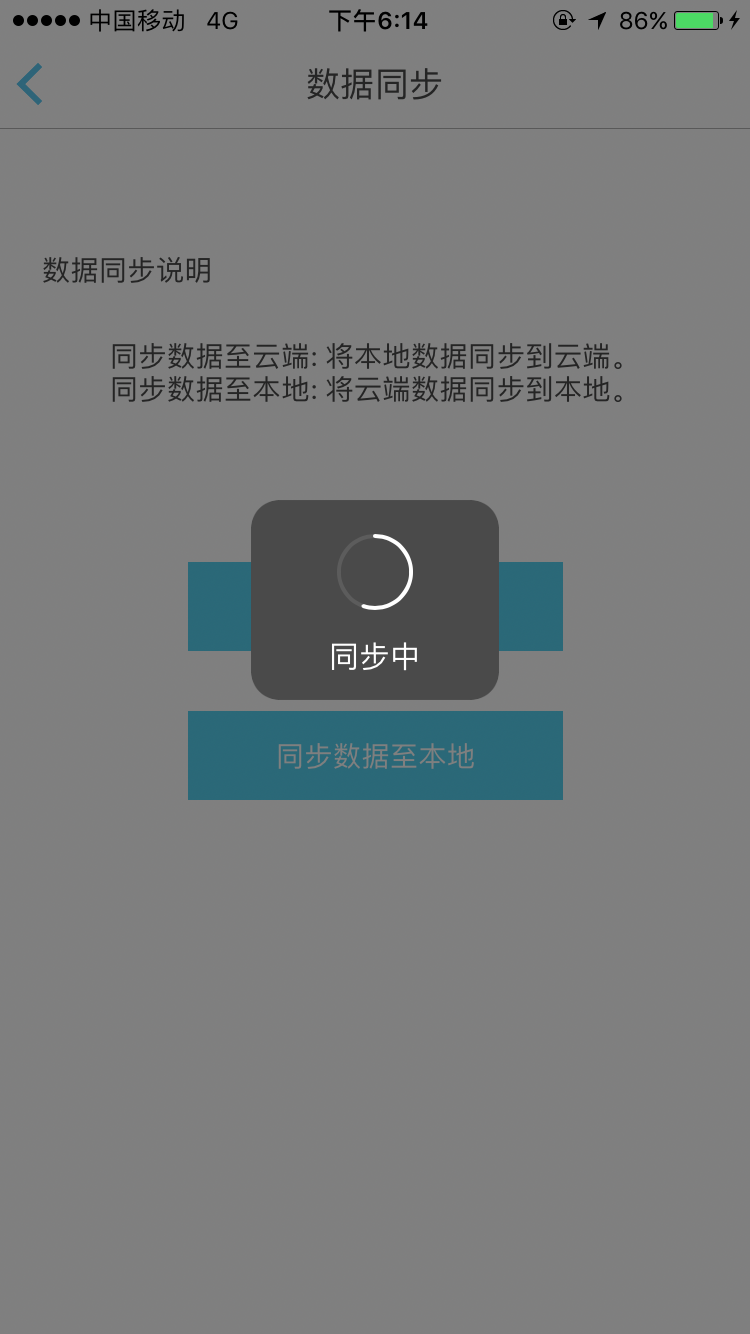
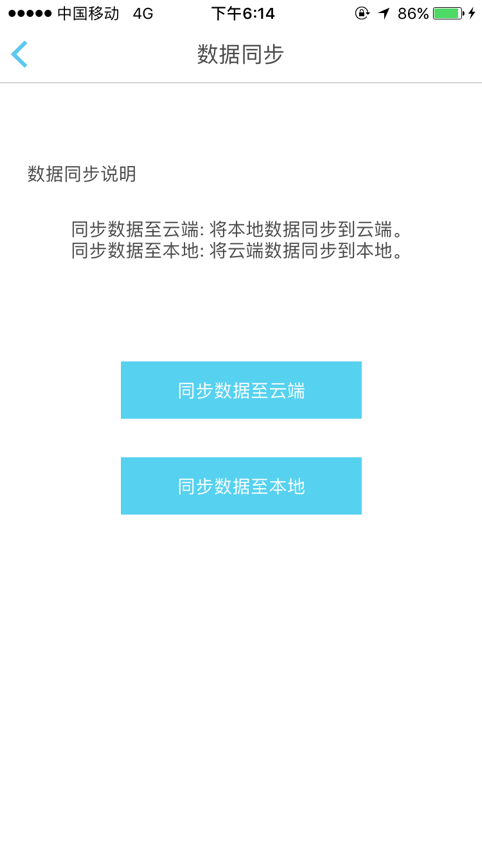
(2) 个人信息修改



(3)密码修改



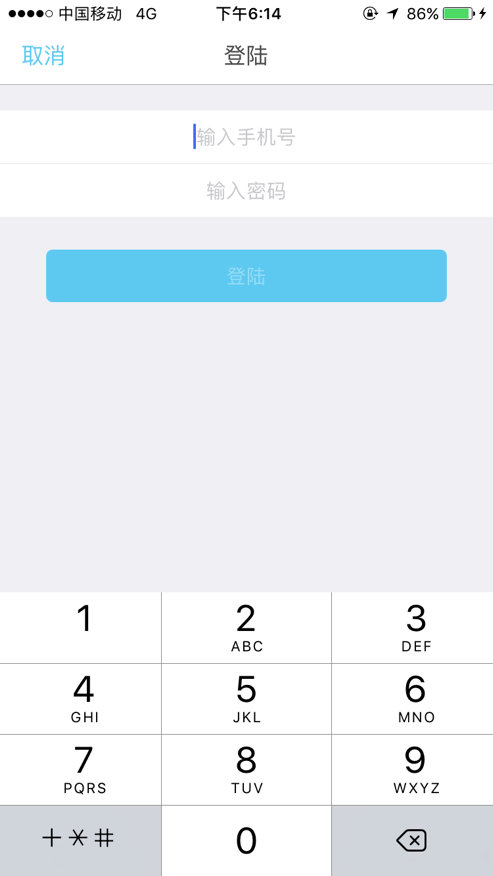
1. 数据同步



1. 设置



4.3.5 登陆/注册模块



4.3.6 分享模块

