

**《软件工程》课程作业**

Keep-Sporting健身系统

|  |  |
| --- | --- |
| 组长姓名 | 李金泽-20172202955 |
| 同组学生 | 杨立飞-20172202996  于家宝-20172202965  许开鑫-20172202968 金健琛-20172202991  卢成浩-20172202933 |
| 学 院 | 信息与电气工程学院 |
| 专 业 | 软件工程 |
| 班 级 | 201701 |
| 任课教师 | 钟丽 |

## 项目分工

|  |  |  |  |
| --- | --- | --- | --- |
| **学 号** | **姓 名** | **个人完成任务** | **答辩成绩** |
| **20172202955** | **李金泽** | 项目需求分析、报告汇总、总结 |  |
| **20172202996** | **杨立飞** | 体系结构 |  |
| **20172202965** | **于家宝** | 系统测试 |  |
| **20172202968** | **许开鑫** | 代码管理 |  |
| **20172202991** | **金健琛** | 系统原型、交互设计 |  |
| **20172202933** | **卢成浩** | 系统实践与设计模式 |  |

目录

**[Keep-Sporting 健身系统 1](#_Toc23986)**

**[1.项目需求分析 3](#_Toc20668)**

[1.1项目背景 3](#_Toc23424)

[1.2需求理解 3](#_Toc18749)

[1.3用例图 8](#_Toc26845)

**[2. 系统原型 14](#_Toc22181)**

[2.1工具选择 14](#_Toc20933)

[2.2主要需求原型设计 14](#_Toc21419)

**[3. 交互设计 18](#_Toc29205)**

[3.1 产品设计原则 18](#_Toc28851)

[3.2 产品设计效果 19](#_Toc21684)

[3.3 交互设计应用 22](#_Toc212)

**[4. 体系结构 23](#_Toc19710)**

[4.1体系结构设计考量 23](#_Toc11279)

[4.2体系结构设计图 23](#_Toc25763)

**[5. 系统实现与设计模式 24](#_Toc6206)**

[5.1系统核心功能实现 24](#_Toc2390)

[5.2设计模式应用 25](#_Toc26247)

**[6. 代码管理 27](#_Toc245)**

[6.1代码管理平台与角色分工 27](#_Toc4439)

[6.2 Git使用记录 27](#_Toc8224)

[6.3部分源码 29](#_Toc22313)

**[7. 系统测试 31](#_Toc24148)**

[7.1 单元测试记录 31](#_Toc174)

[7.2 性能测试记录 34](#_Toc31745)

**[8. 结论 36](#_Toc1297)**

# 1.项目需求分析

## 1.1项目背景

Keep虽然可以拥有庞大的用户群体，也有着旺盛的消费需求，但因为健身是非常线下的活动，纯线上的App如何联系线下又成为难点。而没有稳定的变现模式，也就没有办法进一步发展。

其次，Keep不得不面对低活跃的现实问题。从健身内容来讲，处于内容同质化时期。谁能增加内容的丰富性，输出更多优质的UGC，谁才有核心竞争力。同时，KeepUp产品小组通过用户调研报告（附件2），得出优化计步功能以及直播是大众所期待的，且Keep4.0也已经对计步做出了优化。因此KeepUp产品小组认为Keep下一个迭代方向是新增直播功能。

众所周知，视频教学要比图文教学更加直观，效率更加高效。而直播教学则更佳高效，经过我组头脑风暴，直播健身是个不错的选择。之所以选择直播健身节目的模式，也跟当下直播行业火热的趋势相契合，想要抓住直播这个风口。通过直播健身节目，做“在线健身房”，实现商业化。

## 1.2需求理解

##### 1.2.1功能需求

### 1、直播路线

与其他直播平台不同的是，我们对于keep直播的定位是健身社交。例如：

* 帮助你合理制定健身计划的规划师。
* 曾经看不着脖子看不着腰，现在简直是十六块腹肌人鱼线马甲线什么都有的励志帝。
* 帮你合理搭配膳食越吃越瘦的暖心营养师。

### 2、一个活动： 线下健身

线下活动，我们决定联合优质健身房策划一次keep线下体验活动，选择一家优质的健身房（最好是连锁健身品牌），聚集健身房周围的keep用户，进行一次健身之旅活动。

健身之旅主要包含：

1. 健身房明星教练（最好还是keep用户）讲授健身心得
2. 教练和keep用户互动环节
3. 身体机能检测，并同步至keep
4. 理疗体验（用户根据需求可选）
5. 健身房会员转化

### 3、一些优化： 用户体验提升

* 增加播放音乐入口，使用户可以在跑步时播放keep内置音乐。
* 优化卡路里显示，将消耗的卡路里转化为人们共识度高的食物增加在卡路里数字下面显示辅助理解。男生（炸鸡腿）女生（奶油蛋糕）。
* 增加日排行榜统计，统计当日运动量。
* 采用一些阅读App的奖励机制，对于在线时长以及每日打卡活动给予一定的金币奖励，可以兑换相应的现金金额。

### 4、直播

直播具有强社交性和互动性，变现模式成熟，目前不乏拥有直播模块的竞品。通过对竞品的观察，我们认为通过直播对于用户活跃度、产生平台级明星是有利的，难点在于产出的直播内容符合KEEP一直以来的社区风格与品牌形象。基于该问题，当前除设置直播门槛以外，建议还应进行以下运营工作，在直播社区内形成鲶鱼效应：

（1）.KOL管理

从已有KOL中发掘直播模块的种子用户进行用户运营，在版本上线前后为其进行培训与一定的帮助，并为内容提供建议，提高直播质量。

（2）.官方设置高质量的直播课程

邀请KEEP强大的课程研发团队成员、国内外不同领域的健身权威、优质私教、KEEP社区达人等参与官方直播课程，课程内容可包含但不限于以下内容：

增肌、塑性、跑步、瑜伽等健身训练

健康饮食

健身器材的使用与注意事项

运动装备挑选等

### 5、跑步音乐

基于该版本跑步的音乐设置，我们目前需要进行以下两点工作：

（1）.丰富曲库歌曲

方案一：自行寻找合适歌曲并签入版权

方案二：寻求第三方机构（如网易云音乐等）进行合作

对比以上两种方案，我们认为方案二具有短期内成本低、耗费精力少、效率高的特点，因此建议实施方案二，寻求与第三方音乐机构的合作。

（2）.对曲库歌曲进行官方分类

后续可根据用户需求与反馈，决定是否可让用户自行创建歌单，并进行音乐版块的精细运营，同时达成丰富社区内容与提高用户活跃度的目的。

### ****6、跑步、骑行音乐设置****



### 7、一些优化： 用户体验提升

* 增加播放音乐入口，使用户可以在跑步时播放keep内置音乐。
* 优化卡路里显示，将消耗的卡路里转化为人们共识度高的食物增加在卡路里数字下面显示辅助理解。男生（炸鸡腿）女生（奶油蛋糕）。
* 增加日排行榜统计，统计当日运动量。
* 采用一些阅读App的奖励机制，对于在线时长以及每日打卡活动给予一定的金币奖励，可以兑换相应的现金金额

##### 1.2.2非功能需求

1. 卡路里提示图像化表达



1. 每日训练排名



##### 1.2.3数据需求

1、累计观看人次：整个直播过程中用户点击进入的次数，统一用户多次重复进入观看，也会计人次数，比如同一用户重复进入3次，累计观看人数就为3次, 按节目,按日的纬度进行统计。

2、累计用户量：直播过程中累计观看人数的统计，同一用户多次进入直播不会重复统计，比如同一用户重复进入4次，累计用户量为1, 按节目,按日的纬度进行统计。

3、累计观看时长：直播过程中所有观众累计起来的观看时长统计。

4、累计直播时长：直播方下所有活动累计的直播时长, 按节目,按日的纬度进行统计。

5、最高并发峰值：并发数表示同时在线观看的人数，最高并发峰值表示整个直播过程中 同时在线观看人数最多时刻的人数统计, 按节目,按日的纬度进行统计。

6、累计IP数：同个局域网内，无论多少用户观看直播，记录的IP数只会记录一个，不同局域网，IP不同，会累计多个IP。打个比方，办公室内5个用户均在同个局域网内，这5个观7、众观看直播，在IP统计中只会记录1个IP，按节目、按日的纬度进行统计。

## 1.3用例图

表1、所有用户登录系统

|  |  |
| --- | --- |
| 用例名称 | 登录系统 |
| 简要描述 | 根据用户输入的用户名和密码判断用户的身份，赋予相应的权限。 |
| 参与者 | 软件开发人员，测试人员，使用用户等 |
| 前置条件 | 系统正常运行并且主要功能应该实现。 |
| 后置条件 | 用户可登陆进行使用功能 |
| 基本事件流 | 1 输入用户名，检验用户名是否存才  2 输入密码，校验密码是否正确。  3 用户登陆界面并使用功能 |
| 备选事件流 | 用户名错误或密码错误  系统显示用户名错误或密码错误的提示信息，并进入(2)步  用户可以重新输入用户名和密码进行登录，也可以选择直接退出系统  如若用户忘记了密码，可以通过一定的方式找回密码 |
| 补充约束 | 系统允许用户重试三次登录操作，超过三次后系统自动结束，不允许用户重试 |
| 相关图 | 说明: http://yzhtml01.book118.com/2016/11/07/21/29482584/25.files/file0001.jpg |

表2、用户修改个人信息

|  |  |
| --- | --- |
| 用例名称 | 用户修改个人信息 |
| 用例描述 | 用户登陆后，查看个人信息，选择修改项，进行信息修改。 |
| 参与者 | 使用用户 |
| 前置条件 | 具有账户的用户登录到系统 |
| 后置条件 | 用户可以使用各种功能 |
| 基本流程 | 用户登录软件。  切换到个人信息界面。‘  选择修改项，将项目进行修改。  输入密码，进行保存  修改成功，查看修改后的信息 |
| 备选事件流 | 如果密码错误，要重新输入密码。  如果有非法修改，则无法修改。 |
| 相关图  相关图 | 说明: 1ad5ad6eddc451da107e7db2b5fd5266d11632da |

表3、添加健身好友

|  |  |
| --- | --- |
| 用例名称 | 添加健身好友 |
| 简要描述 | 用户登录后，切换到搜索好友页面，可以选择输入账号添加，也可以输入查找条件进行查找。 |
| 参与者 | 用户等使用人员 |
| 前置条件 | 用户应使用账号登陆 |
| 后置条件 | 添加好友，好友应该存在 |
| 基本事件流 | 1、用户使用密码登陆  2、找到软件的搜索功能  3、输入查找信息  4、进行添加好友，等到对方同意双方可成为好友 |
| 备选事件流 | 搜索好友不存在，或用户无法添加，可以进行其他添加方式，如二维码添加，面对面添加。  对方不同意时，可以选择关注。 |

表4、制定健身计划表

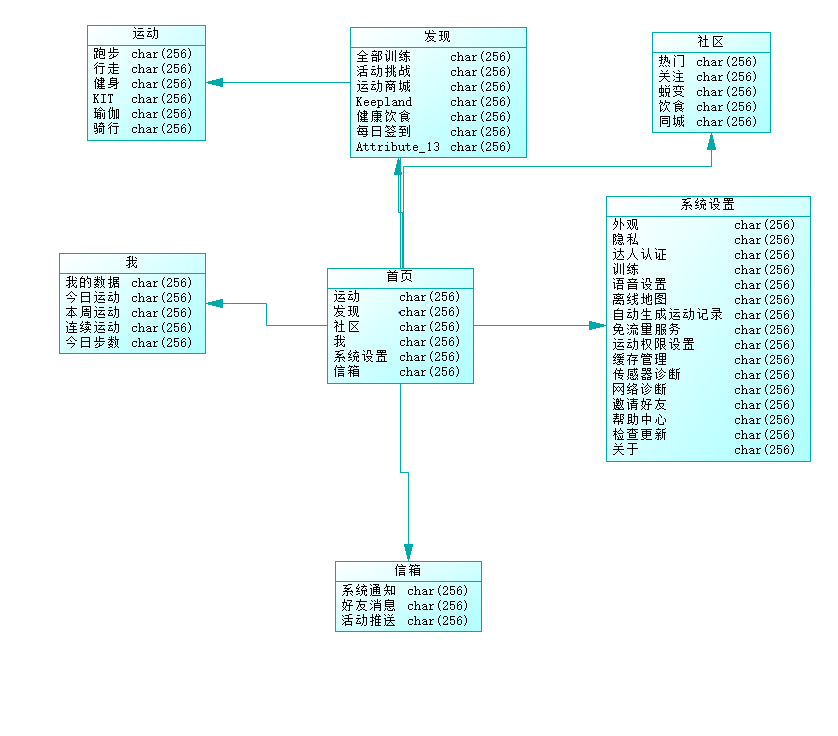
|  |  |
| --- | --- |
| 用例名称 | 制定健身计划表 |
| 用例描述 | 通过制定健身计划表，合理安排健身计划，科学的健身可以保证健康的身体 |
| 参与者 | 使用者，或其他用户 |
| 前置条件 | 用户登录健身软件 |
| 后置条件 | 无 |
| 基本事件流 | 用户登录软件，转移到健身计划训练营  参考系统推荐或健身达人的训练计划，合理制定自己的训练计划  保存计划，一键执行 |
| 备选事件流 | 暂无 |

表5、删除用户

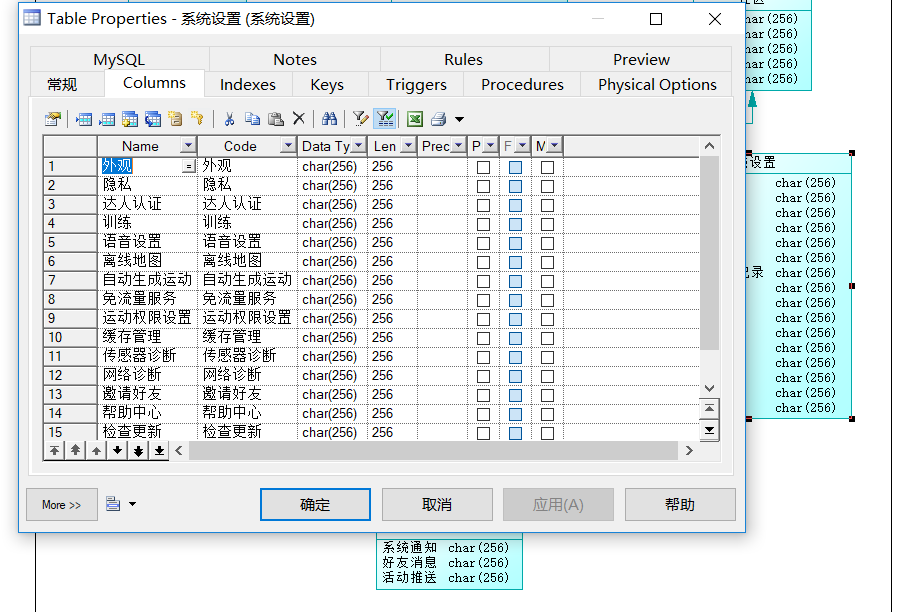
|  |  |
| --- | --- |
| 用例名称 | 删除用户 |
| 用例描述 | 维护人员或后台人员删除违规用户 |
| 参与者 | 后台人员 |
| 前置条件 | 用户在软件上有违规操作，例如发送不良信息，恶意充值，以极少的金额刷会员等 |
| 后置条件 | 暂无 |
| 基本事件流 | 后台人员在维护过程中对用户进行查看是否有不良行为  找到后进行封号，删除用户等惩罚 |
| 备选事件流 | 后台人员进行错误操作，用户可以游客的身份进行联系客服，协商解决 |
| 相关图（图片相关性不大） | 说明: http://www.toberp.com/upfile/eweb/2014311921953.jpg |

表5、充值会员

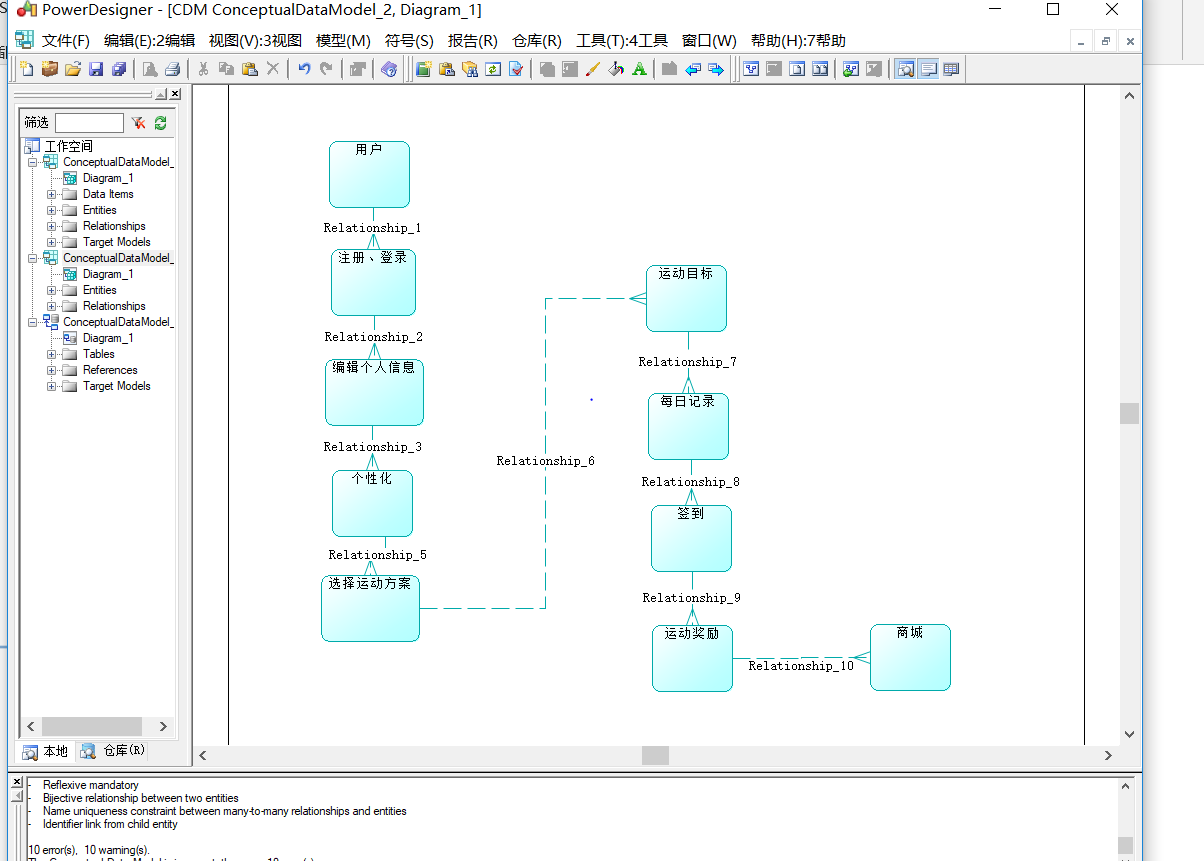
|  |  |
| --- | --- |
| 用例名称 | 充值会员 |
| 用例描述 | 用户充值会员 |
| 参与者 | 使用者 |
| 前置条件 | 使用者通过账户登录系统 |
| 后置条件 | 暂无 |
| 基本事件流 | 用户登录软件，点击充值功能  选择充值时间，选择充值金额，选定充值方式  输入密码，进行充值  享受充值带来的优惠 |
| 备选事件流 | 充值失败，请重新充值  如果充值金额减少，但会员服务未开启，请联系后台 |
| 相关图 | 无 |



**图 1**



**图2**



**图3**

# 2. 系统原型

## 2.1工具选择

**Xcode10（基于苹果系统的APP设计软件）**

## 2.2主要需求原型设计

**（1）登录、注册界面（包含第三方登录）**

**注：如图所示的界面，是在苹果虚拟机运行的结果。（iphone-X）**



**（2）个人中心**



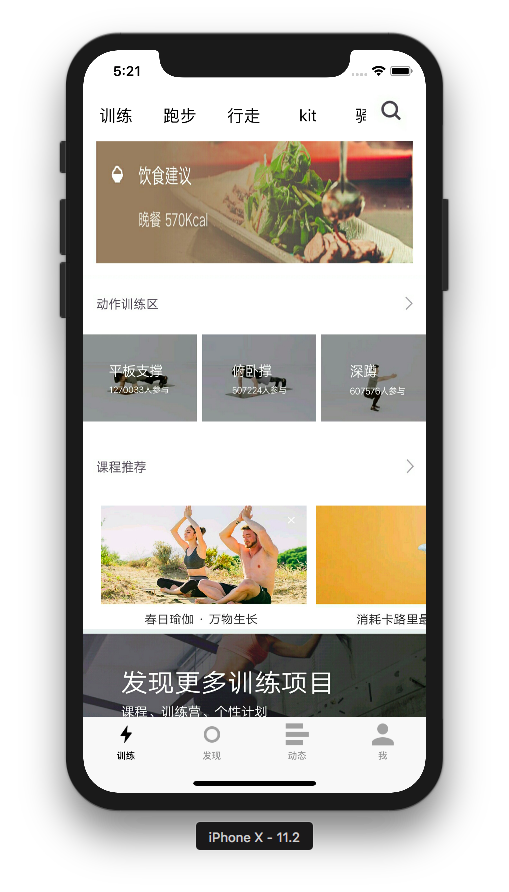
**（3）开始训练**



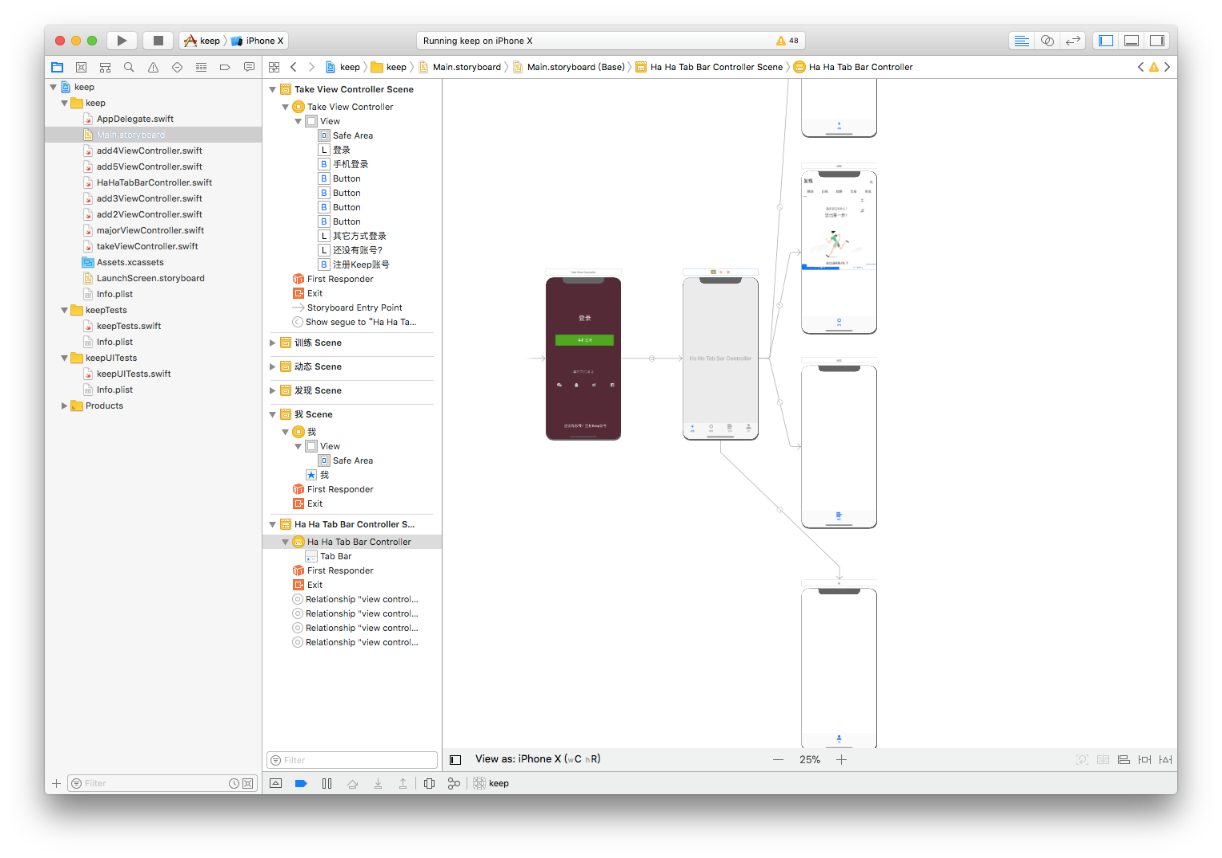
**（4）进度查看**

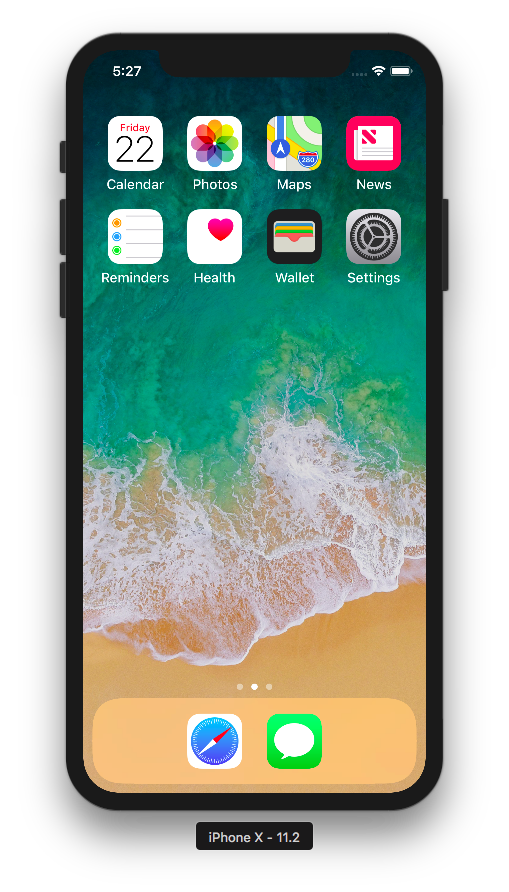


**（5）推荐方案**



**（6）系统整体设计方案和虚拟机展示**





# 3. 交互设计

## 3.1 产品设计原则

1，重点关注产品核心部分的交互点。如训练安排、个人喜好等。

2，站在不同的人群设计产品，如健身教练、专业健身人士、初学者。

3，实现新颖功能的引入，如交友设计、空间动态。

4，重点使用面向对象的设计原则：SRP职责单一原则。核心思想是一个类只做一件事，实现功能的分离以便于代码调试。

5. 基于AI的推荐系统，根据浏览记录推荐相应的课程。

## 3.2 产品设计效果

1，注册、登录界面（主要细节如图所示）

**登录部分：**

（1）点击手机登录按钮——>跳转至相应界面输入相关信息；

（2）点击手机号输入框——>弹出键盘，输入手机号；

（3）点击密码输入框——>弹出键盘，输入密码；

（4）点击确认密码输入框——>弹出键盘，再次输入密码。

**注：**

**1.在相应登录界面内，也可以选择验证码登录。（如图）**

**2.在初始软件界面，亦可选择第三方登录（如QQ、WeChat等）**

经系统后台验证后，分为以下几种情况：

密码不正确——>跳回原登录界面，并弹出提示信息；

密码正确——>跳至软件的主界面；

此时点击确定—>按钮，完成登录界面的交互。

**注册部分：**

（1）点击初始界面最下方注册按钮——>跳转相应界面输入相关信息；

（2）点击手机号注册——>弹出键盘，输入手机号；

（3）点击QQ注册——>跳转至QQ软件，进行授权；

（4）点击WeChat注册——>跳转至微信软件，进行授权；。

经系统后台验证后，分为以下几种情况：

手机号已存在——>弹出提示框，号码已存在可以直接登录；

第三方软件授权失败——>弹出提示框，稍后重新尝试，即可完成本界面的交互。



2，核心部分功能（开始训练和进度查看）

本部分是整个软件的核心功能，分为开始训练和进度查看。

（1）开始训练：

此界面的交互过程：用户点击“开始跑步”按钮，后台进行实时记录，主要包括里程数和训练时长：让用户在首次体验软件的时候能对该软件有大体的了解。

UI界面如图：



在该界面新增了两个功能：周围动态和排名查看。本部分为该软件新引入的功能，可以实现在跑步时，根据地理定位找出周围正在跑步的人，并与其进行数据对比（排名），在某些方面可以实现交友的目的。

UI界面如图：



（2）排名查看：

该界面的交互：

（1）天数值变化情况：从数据库中查取到该用户的历史训练总天数，通过数据可视化技术将该用户的总天数显示到屏幕上。

（2）天数值排行榜：从数据库中查取到最近一周的全部用户天数值的前100名，通过数据可视化将用户的天数值，以列表的形式显示在UI上。

UI界面如图：



3，基于AI的推荐方案

本部分的交互过程：通过近期用户的相关浏览记录，然后借助AI的相关技术，可以实现在商城或者训练课程中推荐适合该用户的信息，使用户感到舒适。

UI界面截图如下：



## 3.3 交互设计应用

刚登录软件后，进入到主界面，在上方显示相关导航栏，下方分为训练、发现、动态、我的等四个主模块，点击训练模块，选择相应模式（户内或户外）即可进入相关训练，训练之后会显示训练时长和训练时间，同时可以在排名模块中，查看该用户在周围训练人群中的排名。

从开发者角度来说，交互设计，是定义、设计人造系统的行为的设计领域，它定义了两个或多个互动的个体之间交流的内容和结构，使之互相配合，共同达成某种目的。交互设计努力去创造和建立的是人与产品及服务之间有意义的关系，以“在充满社会复杂性的物质世界中嵌入信息技术”为中心。交互系统设计的目标可以从“可用性”和”用户体验“两个层面上进行分析，关注以人为本的用户需求。

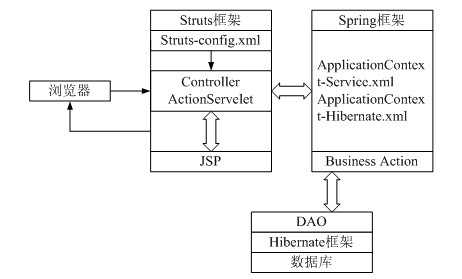
从用户角度来说，交互设计是一种如何让产品易用，有效而让人愉悦的技术，它致力于了解目标用户和他们的期望，了解用户在同产品交互时彼此的行为，了解“人”本身的心理和行为特点，同时，还包括了解各种有效的交互方式，并对它们进行增强和扩充。交互设计还涉及到多个学科，以及和多领域多背景人员的沟通。通过对产品的界面和行为进行交互设计，让产品和它的使用者之间建立一种有机关系，从而可以有效达到使用者的目标，这就是交互设计的目的。

# 4. 体系结构

## 4.1体系结构设计考量

基于B/S模式开发管理Keep-Sporting系统应用软件的实现原理和关键技术,提出了Keep-Sporting系统应用软件的设计方案,阐述了开发软件所采用的struct2技术及数据库触发器技术在Web应用程序中的应用,重点论述了根据实际问题设计数据库触发器的实现方法以及在软件功能中设计实现相关数据库触发器的点火条件,实现触发器的预定功能。使用java EE+MYSQL技术开发完成该系统的全部功能。

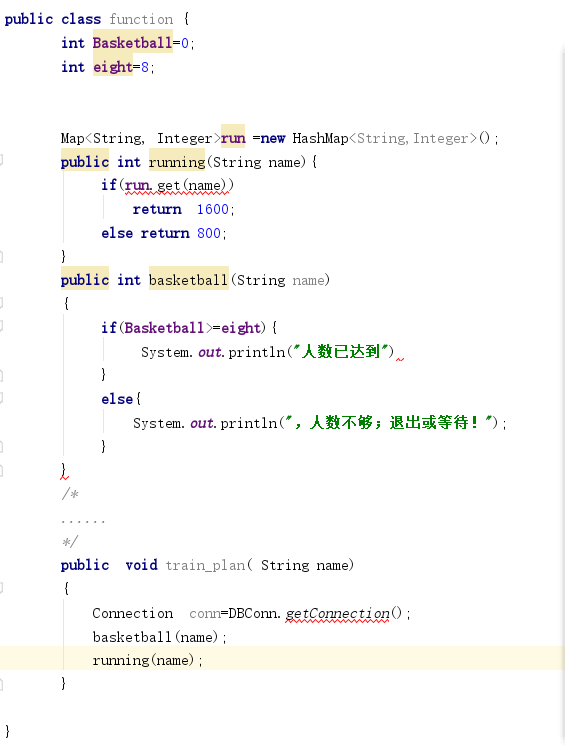
## 4.2体系结构设计图



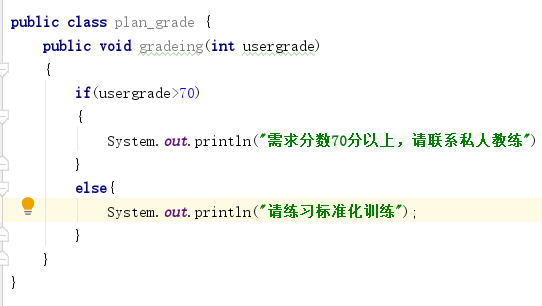
# 5. 系统实现与设计模式

## 5.1系统核心功能实现

1. 提供基础的、标准化的、可以随时开始的训练计划/课程，尽可能多的覆盖涉及运动的各场景；



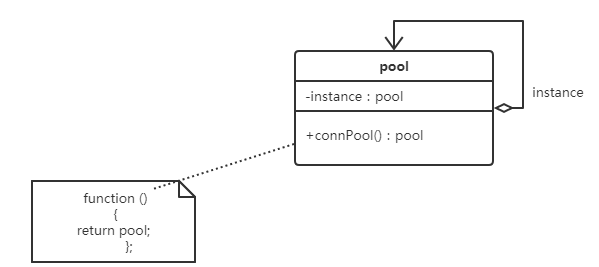
2）主要满足健身需求0-70分的群体；（70分以上需要定制化的，而非标准化的计划与服务）



## 5.2设计模式应用

**5.2.1**单例模式

功能：用于返回唯一的数据库连接池。



### 5.2.2策略模式

Context(应用场景):

1. 需要使用ConcreteStrategy提供的算法。

2. 内部维护一个strategy的实例。

3. 负责动态设置运行时strategy具体的实现算法

4. 负责跟Strategy之间的交互和数据传递。

Strategy(抽象策略类)：

定义一个公共接口，各种不同的算法以不同的方式实现这个接口，context使用这个接口调用不同的算法，一般使用接口或者抽象类实现。

ConcreteStrategy(具体策略类)：

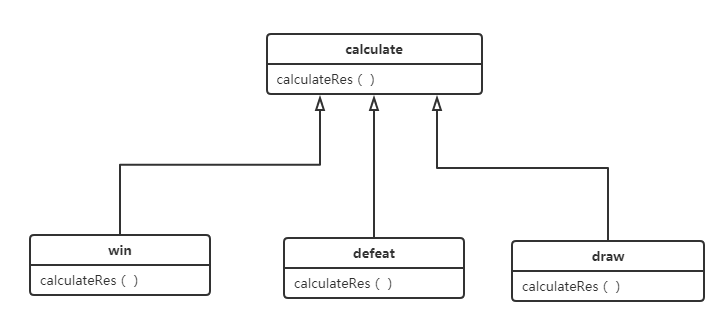
实现了Strategy定义的接口，提供具体的算法实现。

应用场景：  
　　1、 多个类只区别在表现行为不同，可以使用Strategy模式，在运行时动态选择具体要执行的行为。(例如健身产品促销)  
　　2、 需要在不同情况下使用不同的策略(算法)，或者策略还可能在未来用其它方式来实现。(具体优惠活动)  
　　3、 对客户隐藏具体策略(算法)的实现细节，彼此完全独立。

### 5.2.3代理模式(Proxy)

处理跟外部系统或硬件交互时都用到，例如打印（只是调用打印类的方法，而不是直接调打印机接口）、例如发送JMS消息、导出excel、导入数据到其他系统(ERP),都会用一个代理类操作。

另外，代理的模式能够在原来的方法上加入自己想要的逻辑。spring AOP就是使用了java的动态代理机制



# 6. 代码管理

## 6.1代码管理平台与角色分工

代码管理平台：GitHub

角色分工：金健琛：交互设计的实现、大部分界面

李金泽：项目需求

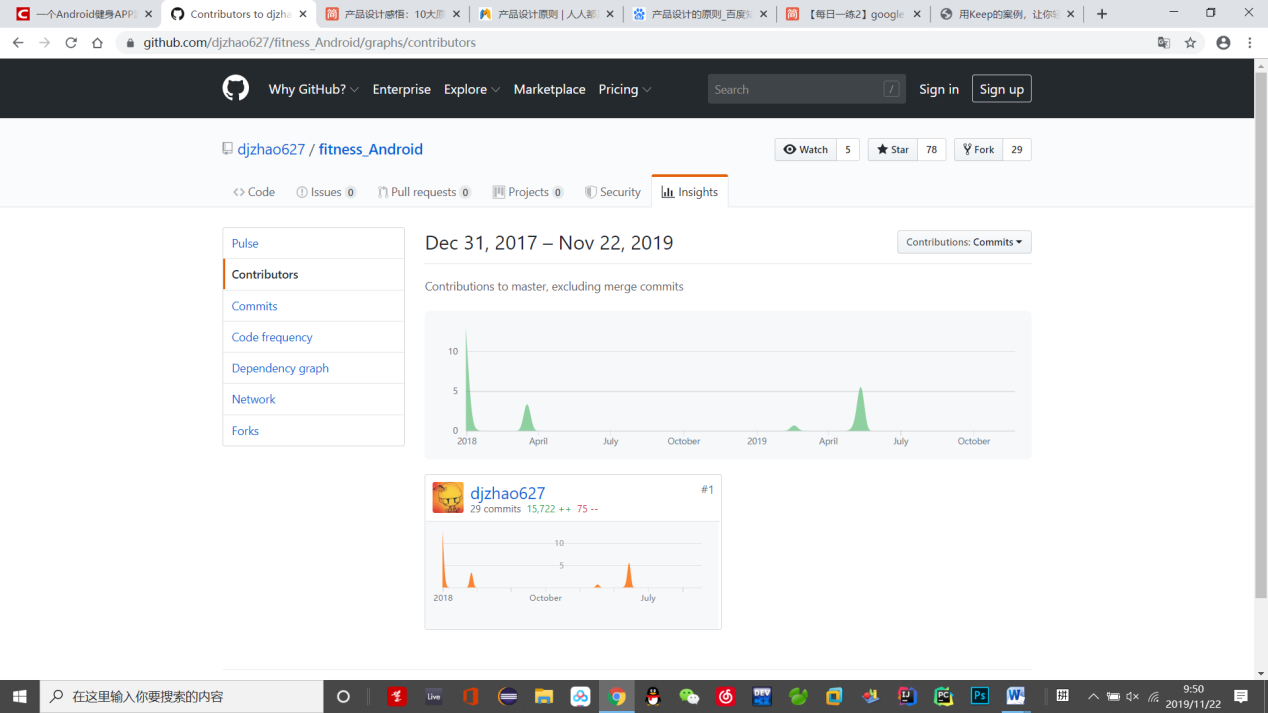
杨立飞：体系结构设计、设计者模式实现

许开鑫：用户登录界面、代码管理

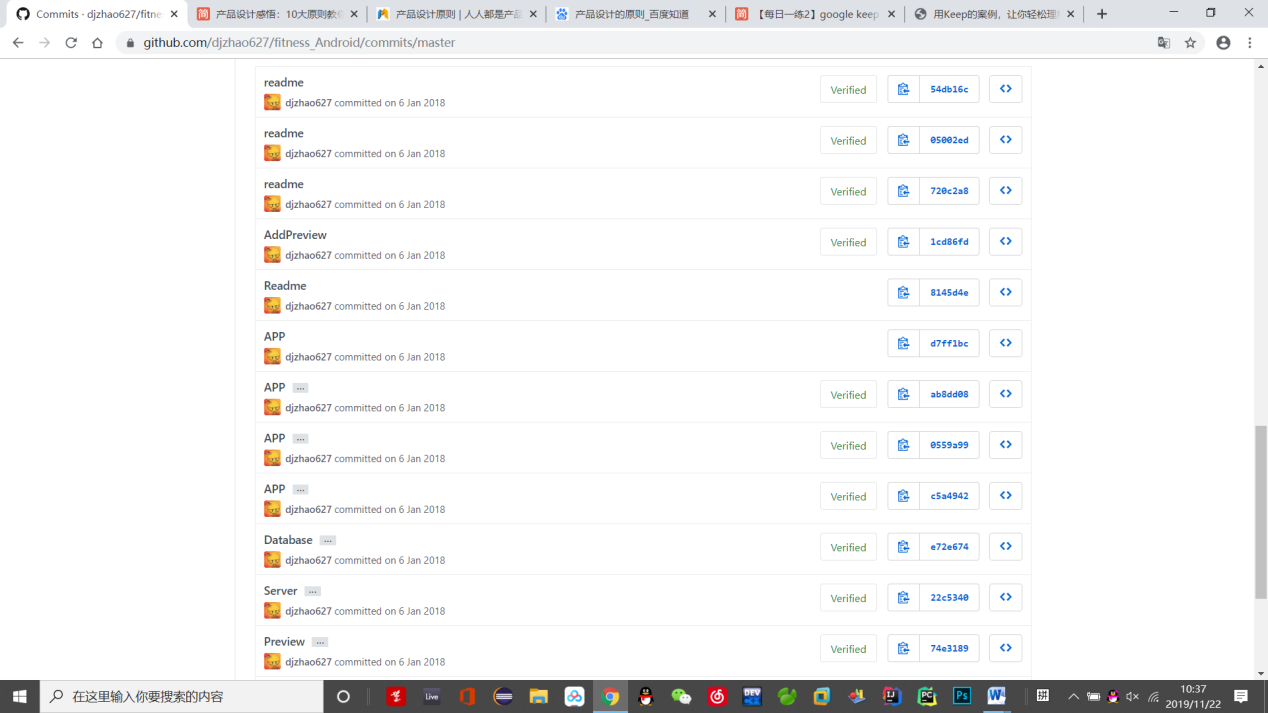
卢成浩：部分算法设计

于家宝：数据库设计及实现

## 6.2 Git使用记录







## 6.3部分源码

import UIKit

class add4ViewController: UIViewController {

    //1.视图之间的联动    first!

    //2.点击按钮，用代码实现交互。 second。尽力

    //3.侧滑

    //4.点击按钮背景图改变 不是长按，即非高亮！

    //5.segmentcontroller的选项的背景图！

    //6.在可以平！（竖的解决了）滑动的界面中的副界面加入这个segmentcontroller。 以后再说

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

    /\*

    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before navigation

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        // Get the new view controller using segue.destinationViewController.

        // Pass the selected object to the new view controller.

    }

    \*/

}

//

//  add4ViewController.swift

//  keep

//

//  Created by mac on 2018/4/15.

//  Copyright © 2018年 mac. All rights reserved.

import UIKit

class add4ViewController: UIViewController {

    //1.视图之间的联动    first!

    //2.点击按钮，用代码实现交互。 second。尽力

    //3.侧滑

    //4.点击按钮背景图改变 不是长按，即非高亮！

    //5.segmentcontroller的选项的背景图！

    //6.在可以平！（竖的解决了）滑动的界面中的副界面加入这个segmentcontroller。 以后再说

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view.

    }

    override func didReceiveMemoryWarning() {

        super.didReceiveMemoryWarning()

        // Dispose of any resources that can be recreated.

    }

    /\*

    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before navigation

    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {

        // Get the new view controller using segue.destinationViewController.

        // Pass the selected object to the new view controller.

    }

    \*/

}

import UIKit

class HaHaTabBarController: UITabBarController {

override func viewDidLoad() {

//var items: [UITabBarItem] = self.tabBar.items as! [UITabBarItem]

super.viewDidLoad()

self.tabBar.isTranslucent = false//避免受默认的半透明色影响，关闭

self.tabBar.tintColor = UIColor.black //选中这个item 这个item会变色！图片的全蓝色也会变成全黑色！ 这个是设置选该item的图片颜色！

//当在storyboard中设置的图片为未选中的时，颜色一律为灰色！ 当填充的图片为自己所截屏的图片时，有时会出现一片蓝！

//self.tabBar.barTintColor = UIColor.white//整个背景色！！

self.tabBar.barTintColor = UIColor(red: 248/255, green:248/255, blue:248/255, alpha:1)

// Do any additional setup after loading the view.

/\*var xun2SelectedImagine = UIImage(named: "xun2")

var fa2SelectedImagine = UIImage(named: "fa2")

var d2SelectedImagine = UIImage(named: "d2")

var wo2SelectedImagine = UIImage(named: "wo2")

items[0].image = UIImage(named:"xun1")

items[0].selectedImage = xun2SelectedImagine!.withRenderingMode(UIImageRenderingMode.alwaysOriginal)

items[1].image = UIImage(named:"fa1")

items[1].selectedImage = fa2SelectedImagine!.withRenderingMode(UIImageRenderingMode.alwaysOriginal)

items[2].image = UIImage(named:"d1")

items[2].selectedImage = d2SelectedImagine!.withRenderingMode(UIImageRenderingMode.alwaysOriginal)

items[3].image = UIImage(named:"wo1")

items[3].selectedImage = wo2SelectedImagine!.withRenderingMode(UIImageRenderingMode.alwaysOriginal)//这一段代码修改的是，选中该项时的图片！ 而默认的是在一开始storyboard中设置的！\*/

//(UIImageRenderingMode.alwaysOriginal)

}

override func didReceiveMemoryWarning() {

super.didReceiveMemoryWarning()

// Dispose of any resources that can be recreated.

}

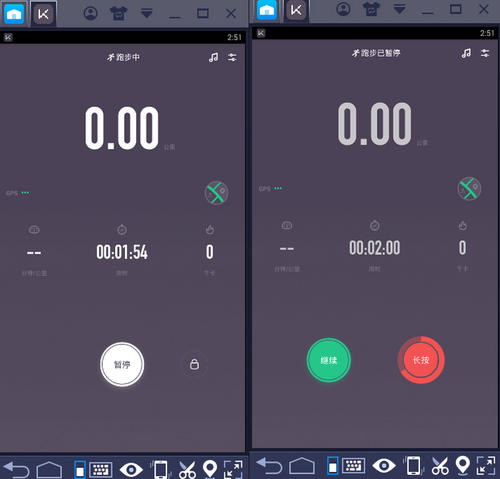
# 7. 系统测试

## 7.1 单元测试记录

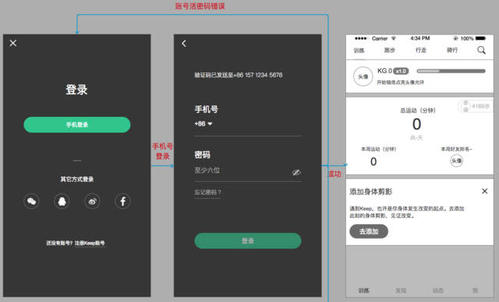
累计IP数：同个局域网内，无论多少用户观看直播，记录的IP数只会记录一个，不同局域网，IP不同，会累计多个IP。打个比方，办公室内5个用户均在同个局域网内，这5个观众观看直播，在IP统计中只会记录1个IP，按节目、按日的纬度进行统计。

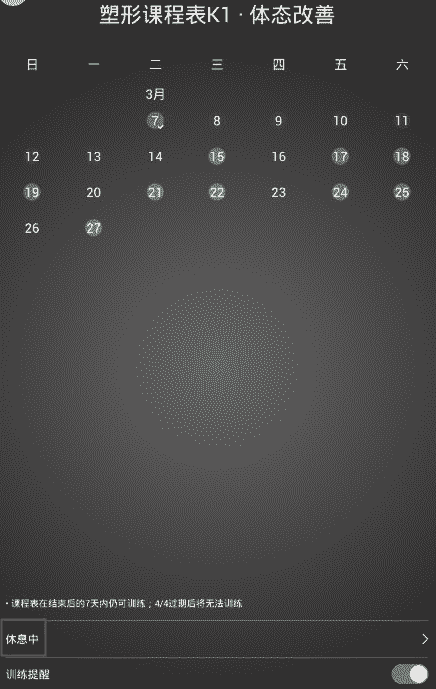
累计用户量：直播过程中累计观看人数的统计，同一用户多次进入直播不会重复统计，比如同一用户重复进入4次，累计用户量为1, 按节目,按日的纬度进行统计。

累计IP数达到一定量级会出现高延迟甚至页面丢失的情况，说明后端设计还有待提升、优化

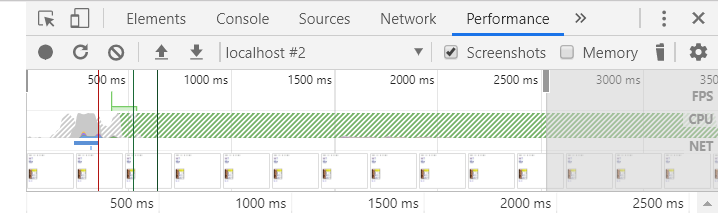


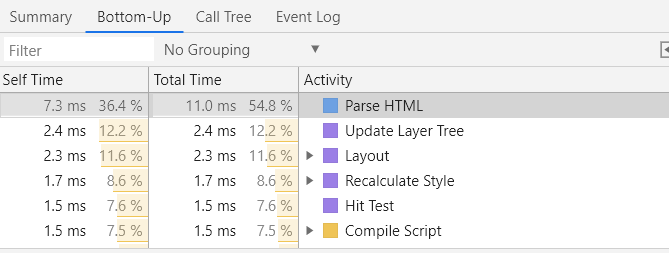
这是运动界面的运行效果，页面之间切换流畅，系统的可用性达到了初步要求，测试通过，在视觉上无明显延迟，加载时间较短无页面丢失或者请求无响应之类的状况。





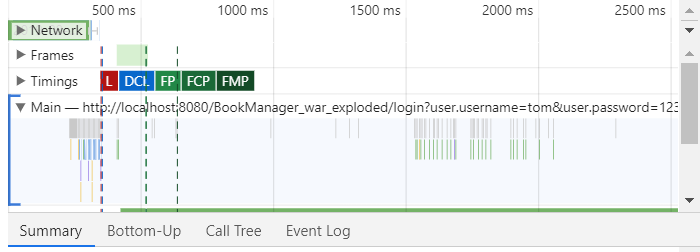
## 7.2 性能测试记录

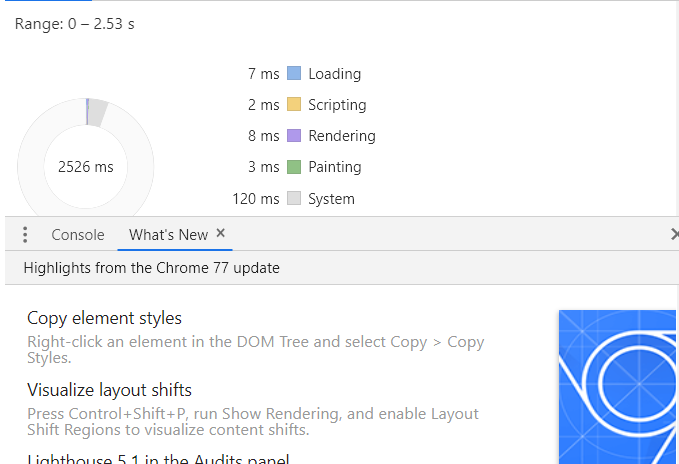




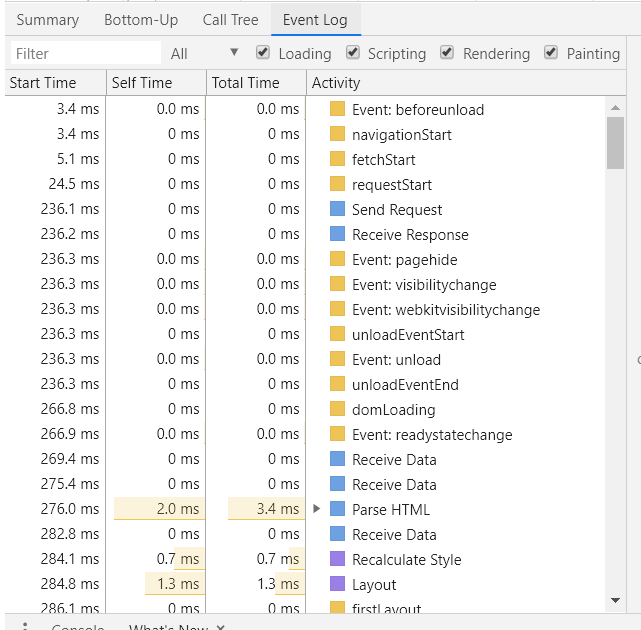
页面加载资源消耗主体PARSE HTML 总用时11ms 占比54.8%

本页面是基于structs2框架的Web页面由于在笔记本上运行idle所以系统启动与idle链接耗时较高达到2706ms，对此我们应该在ElementsUI框架里面进一步优化，对于没有用到的组件进行删除，按需引用，并在静态文件上面开启Gzip压缩，进一步减少传输文件大小。





而除呼叫idle服务外其他资源调用用时总和在理想范围内



由Event Log可见进程曲线基本平缓符合人们的心理预期并呈阶梯状上升

# 结论

Keep专注于运动领域，但并非传统的健身公司，而是以技术及数据来驱动的互联网公司。Keep最初是由健身工具切入运动市场， 以健身视频教学为媒介，为白领尤其是健身小白用户，提供线上免费健身训练服务，满足用户碎片化、随时随地健身的需求。Keep的最终目标是开发出一个能面向更多不同文化的人，影响更多人运动习惯的产品，打造一个以“KEEP”为核心的品牌，希望在用户群中树立“年轻、有活力、Cool”的品牌印象。就如同让用户想搜索时想到谷歌一下，运动时想“Keep”一下。

通过一个学期的软件工程的学习，从选题、需求分析、原型设计、交互设计、体系结构分析、Structs2框架的学习使用、以及系统测试的相关知识和软件的使用，我们顺利完成了Keep-Sporting系统简单的开发、制作，虽然在细节上无法和成熟的系统相媲美，但是在学习过程中出现了许多开发之初不曾料想到的问题，我们也通过自我学习克服了开发中的一些困难。通过此次项目我们学会了分工合作，加深了对所学知识的理解和应用，为以后的工作打下了良好的基础。同时也感谢老师的辛勤指导和我们组同学的刻苦付出。