**华南理工大学软件学院**

**《iOS平台应用开发》课程报告（论文）**

**医患帮：医生患者沟通的平台**

**年级专业：2017软件工程**

**班级： 2017级4班**

**任课教师：冯玉翔**

组长： 李海涛 学号：201630664710

2019—2020学年第一学期

教师考核评分：

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 班级 | 学号 | 评分 |
| 李海涛 | 17级4班 | 201630664710 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 团队总评：  指导老师： 日期： | | | |

**医患帮：医生患者沟通的平台**

**摘要**：现在的互联网越来越发达，全球化对人们的生活的影响也越来越广泛。在互联网时代中，人们已经不仅仅只满足于在医院中了解自己的病情，人们更加的趋向于再网络上面，在家庭里面了解自己的病情，并且能够当自己的私人医生，做到足不出户而身心健康。因此，我们开发一种能够让患者足不出户并且在平时就能够了解自己病情的一种平台，这样能够提高国民普遍的身心健康，并且能够增强国民的医学常识教育。

**关键词**：互联网 全球化 平台 医患

Table of Contents

[一．需求分析 6](#_Toc534578529)

[1.1任务概述 6](#_Toc534578530)

[1.2 功能需求 6](#_Toc534578531)

[1.3 性能需求 7](#_Toc534578532)

[1.4 运行环境描述 8](#_Toc534578533)

[1.5 其他需求 8](#_Toc534578534)

[1.6 市场现状 8](#_Toc534578535)

[二．功能介绍 8](#_Toc534578536)

[2.1总用例图 8](#_Toc534578537)

[2.2 登录注册 9](#_Toc534578538)

[2.2.1 用例图 9](#_Toc534578539)

[2.2.2 用例描述 9](#_Toc534578540)

[2.3 医生选择 11](#_Toc534578541)

[2.3.1 用例图 11](#_Toc534578542)

[2.3.2 用例描述 11](#_Toc534578543)

[2.4 即时聊天 12](#_Toc534578544)

[2.4.1 用例图 12](#_Toc534578545)

[2.4.2 用例描述 13](#_Toc534578546)

[2.5 分享病例 13](#_Toc534578547)

[2.5.1 发送病例 13](#_Toc534578548)

[2.5.2 删除病例 15](#_Toc534578549)

[2.5.3查看消息 16](#_Toc534578550)

[三．项目计划与进度 17](#_Toc534578551)

[3.1项目计划 17](#_Toc534578552)

[四．软件架构 21](#_Toc534578553)

[4.1软件框架 21](#_Toc534578554)

[4.1.1软件总框架 22](#_Toc534578555)

[4.1.2登录界面跳转到注册界面 26](#_Toc534578556)

[4.1.3登录界面跳转到主页面 26](#_Toc534578557)

[4.1.4 从注册界面跳转到主页面 26](#_Toc534578558)

[4.1.5主页面跳转到看病页面（用户类型为：患者） 26](#_Toc534578559)

[4.1.6主页面跳转到看病页面（用户类型为：医生） 27](#_Toc534578560)

[4.1.7主页面跳转到健康知识页面 27](#_Toc534578561)

[4.1.8主页面跳转到用户信息管理页面 27](#_Toc534578562)

[4.1.9主页面跳转到主页内容中的 医学新闻推送页面 27](#_Toc534578563)

[4.1.10主页面跳转到主页内容中的 病情分类科普页面 27](#_Toc534578564)

[4.1.11健康知识面跳转到其中的 健康知识详细介绍页面 28](#_Toc534578565)

[4.1.12用户信息管理页面跳转到 用户名更改页面 28](#_Toc534578566)

[4.1.13用户信息管理界面跳转到 修改密码页面 28](#_Toc534578567)

[4.1.14用户信息管理页面跳转到手机号更改页面 28](#_Toc534578568)

[4.1.8用户信息管理页面（退出登录）跳转到 登录页面 28](#_Toc534578569)

[4.2各种类说明 29](#_Toc534578570)

[4.2.1 Logon类 29](#_Toc534578571)

[4.2.2 Userdata类 29](#_Toc534578572)

[4.2.3 News类 30](#_Toc534578573)

[4.2.4 km类 30](#_Toc534578574)

[4.2.5 Homepage类 30](#_Toc534578575)

[4.2.6 Health类 31](#_Toc534578576)

[4.2.7 User类 31](#_Toc534578577)

[4.2.8 Register类 31](#_Toc534578578)

[4.2.9 Asking类 31](#_Toc534578579)

[4.2.10 Answer类 32](#_Toc534578580)

[4.3资源列表 34](#_Toc534578581)

[五．系统实现 36](#_Toc534578582)

[5.1登录界面的实现 36](#_Toc534578583)

[六．系统测试说明 46](#_Toc534578584)

[6.1测试项目及其说明 46](#_Toc534578585)

[6.1.1测试方案 46](#_Toc534578586)

[6.1.2测试项目 46](#_Toc534578587)

[6.2测试结果 47](#_Toc534578588)

[6.2.1测试结果及其相应的说明 47](#_Toc534578589)

[七．用户手册 48](#_Toc534578590)

[7.1系统说明 48](#_Toc534578591)

[7.2用途 49](#_Toc534578592)

[7.2.1功能 49](#_Toc534578593)

[7.2.2性能 49](#_Toc534578594)

[7.2.2.1精度 49](#_Toc534578595)

[7.2.2.2时间特性 49](#_Toc534578596)

[7.2.2.3灵活性 49](#_Toc534578597)

[7.2.3安全保密 49](#_Toc534578598)

[7.3使用过程 50](#_Toc534578599)

[7.3.1安装与初始化 50](#_Toc534578600)

[7.3.2注册&登陆 50](#_Toc534578601)

[7.3.3首页 50](#_Toc534578602)

[7.3.4看病 50](#_Toc534578603)

[7.3.5健康知识推文 50](#_Toc534578604)

[7.3.6用户信息管理 50](#_Toc534578605)

[八．关键代码 50](#_Toc534578606)

[九．附录 54](#_Toc534578607)

[9.1小组分工 54](#_Toc534578608)

[9.2版本记录 54](#_Toc534578609)

[9.3设计心得 54](#_Toc534578610)

[9.4参考文献 55](#_Toc534578611)

# 一．需求分析

### 1.1任务概述

**项目背景**

现在的互联网越来越发达，全球化对人们的生活的影响也越来越广泛。在互联网时代中，人们已经不仅仅只满足于在医院中了解自己的病情，人们更加的趋向于再网络上面，在家庭里面了解自己的病情，并且能够当自己的私人医生，做到足不出户而身心健康。因此，我们开发一种能够让患者足不出户并且在平时就能够了解自己病情的一种平台，这样能够提高国民普遍的身心健康，并且能够增强国民的医学常识教育。

**实现目标**

软件设计目标能够为全国医生患者之间提供一个健康问题交流环境，主要功能包括以下方面：

1 登录注册

使用该软件的用户可以很方便的注册一个属于自己的账号并登录。

2 患者排队求医

患者登陆界面并选择医生对该医生提出问题，如果医生正在回答问题则该患者排队并显示该患者的队列情况。

3 医生接待患者

医生可以将自己的状态改成上班，回答患者问题并获取相应的奖励。

4 好友管理

可以添加删除好友，查找本地好友。

### 1.2 功能需求

**功能划分**

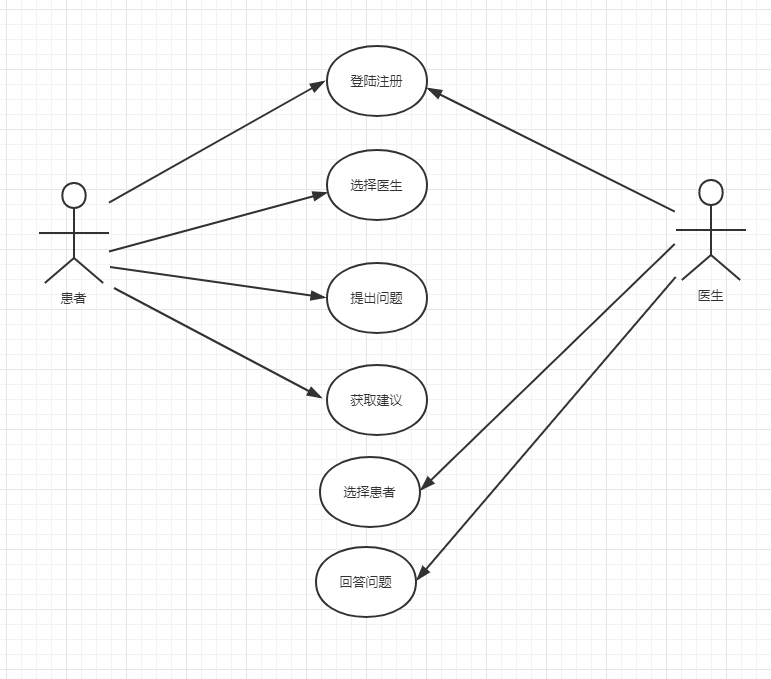


图1

**功能描述**

系统主要有以下几个模块：登录注册，好友管理，即时聊天和分享信息

功能模块表

|  |  |  |
| --- | --- | --- |
| 功能名称 | 功能标识 | 功能详细描述 |
| 注册 | 注册按钮 | 用户可以通过提供用户名和密码进行注册 |
| 登录 | 登录按钮 | 用户可以通过提供用户名和密码进行登录 |
| 医生名单管理 | 选择按钮 | 选择要提问的医生 |
| 医生回答问题，患者收到建议 | 发送按钮 | 该功能就是医生发出的建议患者能够看见并且能够继续跟问。 |

表1

### 1.3 性能需求

**时间要求**

登录、注册、发送消息、接收消息的时间不能超过1s.

**适应性**

可运行在iOS系统智能手机的环境下。

### 1.4 运行环境描述

**硬件**

**软件**

开发环境：VMware 15下的macOS 10.14

开发平台：xcode 11

开发语言：swift

数据库 ：MySQL

测试平台：iPhone 11

网络环境：局域网

### 1.5 其他需求

暂时无其他需求

### 1.6 市场现状

目前市场上有许多聊天工具，但是没有一个专门的医生患者健康问题的及时交流平台。

# 二．功能介绍

系统主要有以下几个模块：登录注册，患者选择医生，医生回答建议，医生受到悬赏。

### 2.1总用例图

如图所示：

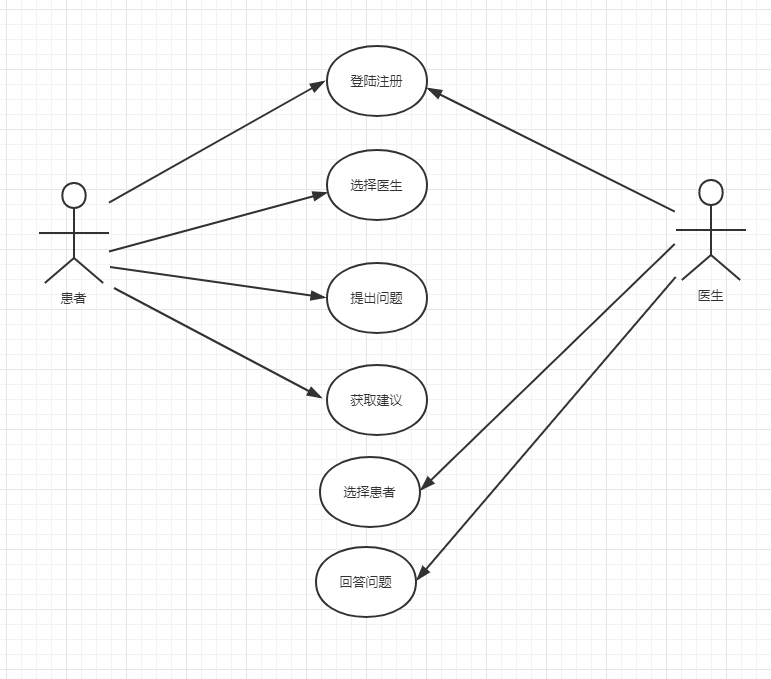


图2

### 2.2 登录注册

### 2.2.1 用例图

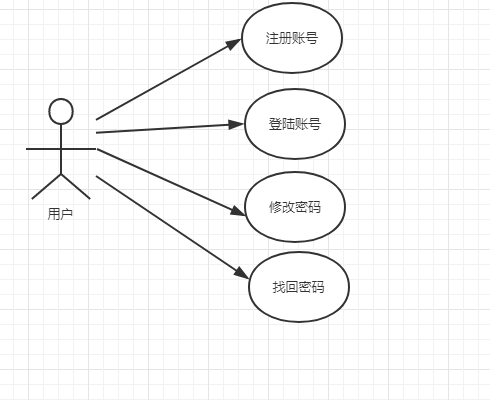


图3

### 2.2.2 用例描述

**2.2.2.1用例简要描述:**

该用例主要描述用户使用软件的时候要进行的登录或注册的行为，同时还有记住密码，寻回密码，修改密码等功能。

**2.2.2.2用例角色:**

用户（医生，患者）

**2.2.2.3用例前置条件:**

用户成功启动app并连接到正确url按照指定规则完成访问

**2.2.2.4用例后置条件:**

登录（注册）成功

**2.2.2.5用例事件流**

基本事件流

1. 登录已有账户；

·用户输入用户名与密码

·手机端获取用户输入内容并传送至服务器

·后台数据库验证用户密码是否匹配

·匹配成功则返回登陆成功并提示首页，失败则返回提示信息

1. 注册新账户；

·用户输入注册信息

·手机端获取输入内容并传送至服务器

·服务器在后台创建新用户并更新数据库

1. 修改账户密码；

·用户输入旧密码

·用户输入新密码

·手机端获取密码并发送至服务器

·后台验证，匹配成功则更新后台服务器

备选事件流

1. 记住账号密码；
2. 登录成功，界面跳转到首页推送；
3. 登录失败，输入账号或密码错误，弹出提醒框
   1. 账号错误提示重输；
   2. 密码输入错误达到三次账户暂时锁定，需要通过邮箱验证解锁；
4. 运行过程程序终止
   1. 重启软件

**2.2.2.6用例非功能性需求:**

1. 界面通俗易懂，操作容易；
2. 医生状态显示通俗易懂；
3. 显示信息准确；
4. 高效执行管理操作
5. 及时显示交流内容

**2.2.2.7用例相关业务数据**

登录：账号，密码

注册：用户名，密码

选择：医生账号列表

提出建议：保存的聊天记录

### 2.3 医生选择

### 2.3.1 用例图

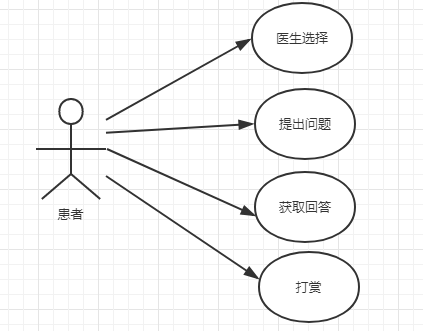


图4

### 2.3.2 用例描述

**2.3.2.1用例简要描述:**

该用例描述用户在使用软件的时候查找医生列表，并选择自己想要提问的医生对象。

**2.3.2.2用例角色:**

患者

**2.3.2.3用例前置条件:**

1. 登录成功；
2. 医生已经上班

**2.3.2.4用例后置条件:**

1. 成功选择医师
2. 成功提问；
3. 成功获取建议；
4. 成功发送打赏；

**2.3.2.5用例事件流**

基本事件流

1. 搜索医师；
2. 服务器返回满足条件的医师列表
3. 选择医师
4. 后台建立医师和患者的聊天通道。
5. 发送提问；
6. 收到回复；
7. 发送打赏；

备选事件流

1. 医师排队完成；
2. 患者排队队列；
3. 提士排到该医师；

**2.3.2.6用例非功能性需求:**

1.界面友好简洁；

2.交互迅速；

3.显示信息准确；

4.高效传递数据信息

**2.3.2.7用例相关业务数据**

1. 医师列表；
2. 患者排队列表；
3. 医患聊天信息；
4. 患者资料；

### 2.4 即时聊天

### 2.4.1 用例图

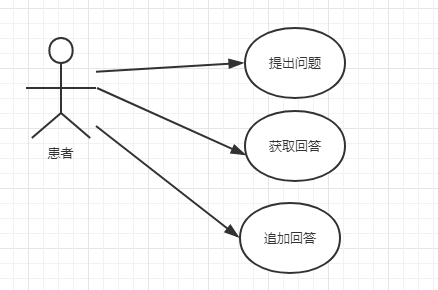


图5

### 2.4.2 用例描述

**2.4.2.1用例简要描述:**

该用例主要描述该应用在医患交流这部分的功能，包括医生和患者的列表，排队情况等等，每一个聊天记录会以文件形式保存在手机中。

**2.4.2.2用例角色:**

用户

**2.4.2.3用例前置条件:**

登录成功，选择医师成功

**2.4.2.4用例后置条件:**

查看聊天记录，聊天记录保存到手机文件中

**2.4.2.5用例事件流**

基本事件流

1. 选定医师新建聊天；
2. 实时连通；
3. 写入聊天内容
4. 客户端获取发送内容及目的地址，传送至服务器
5. 服务器处理后将内容传送至对应目的ip
6. 医/患接收消息
7. 结束聊天；
8. 查询聊天内容；
9. 删除聊天记录；

备选事件流

* + - 1. 在基础事件流1中，当医师正在和另外的患者交流的时候，需要显示排队状态。
      2. 在基础事件流3中，聊天内容：可以是语音，文件，文字

3． 在基础事件流4中，发送内容给对方需要设定最大发送时间T。即：当发送的时间大于T时，那么则认为发送失败，提醒用户重新发送，或者检查当前网络环境。

**2.4.2.6用例非功能性需求:**

1.界面友好简洁；

2.交互迅速；

3.显示信息准确；

4.高效执行管理操作

**2.4.2.7用例相关业务数据**

好友信息，聊天信息（包括文字，语音，文件）

### 2.5 分享病例

### 2.5.1 发送病例

**2.5.1.1 用例图**

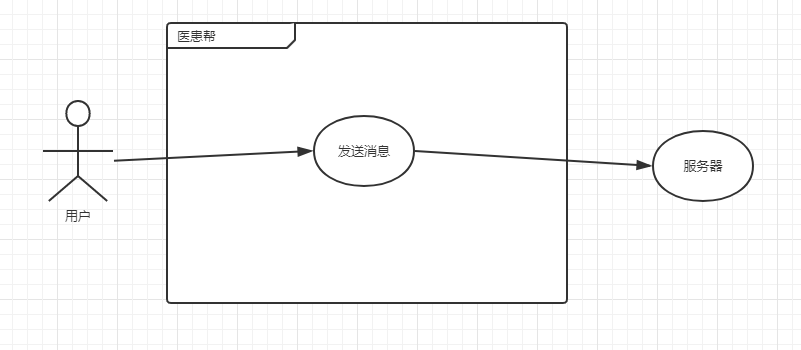


图6

**2.5.1.2 用例描述**

**2.5.1.2.1用例简要描述:**

用户可以在首页中分享自己的治疗效果和治疗经验。

**2.5.1.2.2用例角色:**

用户

**2.5.1.2.3用例前置条件:**

用户成功登录

**2.5.1.2.4用例后置条件:**

可发表信息

**2.5.1.2.5用例事件流**

基本事件流

1.用户1分享自己的治疗效果，系统接受请求后将病例发布在信息列表

2.用户1可以删除自己发布过的病例，也可以查看其他人发布的病例

3.用户2请求查看信息列表，系统刷新列表，显示用户1发布过的病例

4.用户2对用户1的病例进行提问或者评论

备选事件流

用户尝试发送动态时无法连接到服务器：

服务器或用户本地网络出错

将未发布的信息临时保存至本地

重启服务器或再次尝试连接网络

发布本地临时存储的信息

用户无法获得最新动态信息：

服务器或本地网络出错

显示最近一次更新的信息列表

再次尝试连接网络并获取信息

删除动态时无法提交到服务器：

提示用户网络错误未能删除

**2.5.1.2.6用例非功能性需求:**

1.界面友好简洁；

2.交互迅速；

3.显示信息准确；

4.高效执行管理操作

**2.5.1.2.7用例相关业务数据**

分享的文字信息

### 2.5.2 删除病例

**2.5.2.1 用例图**

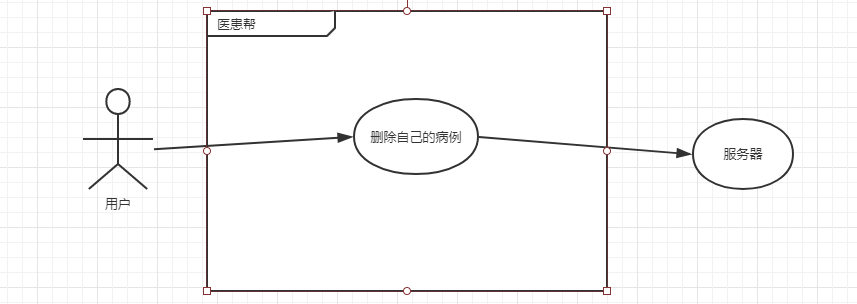


图7

**2.5.2.2 用例描述**

**2.5.2.2.1用例简要描述:**

用户可以删除自己之前发送的病例状态

**2.5.2.2.2用例角色:**

用户

**2.5.2.2.3用例前置条件:**

用户成功登录

**2.5.2.2.4用例后置条件:**

可删除信息

**2.5.2.2.5用例事件流**

基本事件流

1. 删除以往发送的病例；

备选事件

1.网络中断或程序非正常退出，用户重启程序。

**2.5.2.2.6用例非功能性需求:**

1.界面友好简洁；

2.交互迅速；

3.显示信息准确；

4.高效执行管理操作

**2.5.2.2.7用例相关业务数据**

删除的文字信息

### 2.5.3查看消息

**2.5.3.1用例图**

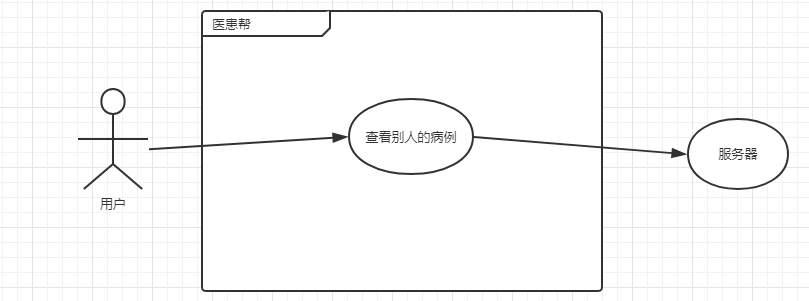


图8

**2.5.3.2 用例描述**

**2.5.3.2.1用例简要描述:**

用户可以查看热度最高的病例

**2.5.3.2.2用例角色:**

用户

**2.5.3.2.3用例前置条件:**

用户成功登录

**2.5.3.2.4用例后置条件:**

可查看别人发送的病例

**2.5.3.2.5用例事件流**

基本事件流

1. 查看好友发送的消息；

备选事件

1.网络中断或程序非正常退出，用户重启程序。

**2.5.3.2.6用例非功能性需求:**

1.界面友好简洁；

2.交互迅速；

3.显示信息准确；

4.高效执行管理操作

**2.5.3.2.7用例相关业务数据**

查看的文字信息

# 三．项目计划与进度

### 3.1项目计划

项目计划表

|  |  |  |  |
| --- | --- | --- | --- |
| 任务名称 | 工期 | 开始时间 | 完成时间 |
| **医患帮项目计划** | **84 个工作日** | **2019年9月7日** | **2019年12月20日** |
| **1、启动** | **30 个工作日** | **2019年9月7日** | **2019年10月13日** |
| 1.1 识别项目干系人 | 2 个工作日 | 2019年9月7日 | 2019年9月8日 |
| 1.2 制作项目章程 | 1 个工作日? | 2019年9月9日 | 2019年9月9日 |
| 1.3 小组召开启动会议 | 1 个工作日 | 2019年9月7日 | 2019年9月7日 |
| 1.4 项目范围规划结束 | 0 个工作日 | 2019年9月7日 | 2019年9月7日 |
| **2、计划** | **9 个工作日** | **2019年9月12日** | **2019年9月21日** |
| 2.1 制定项目开题报告与计划 | 2 个工作日? | 2019年9月14日 | 2019年9月15日 |
| 2.2 项目范围说明书 | 5 个工作日? | 2019年9月15日 | 2019年9月20日 |
| 2.3 项目成本估计 | 2 个工作日 | 2019年9月21日 | 2019年9月22日 |
| 2.2 项目成本计划 | 2 个工作日 | 2019年9月20日 | 2019年9月21日 |
| 2.5 编写WBS字典 | 4 个工作日 | 2019年9月20日 | 2019年9月23日 |
| **3、执行** | **66 个工作日** | **2019年9月22日** | **2019年12月10日** |
| **3.1需求分析** | **6 个工作日?** | **2019年9月22日** | **2019年9月29日** |
| 3.1.1 功能结构图 | 1 个工作日? | 2019年9月22日 | 2019年9月22日 |
| 3.1.2 设计用例 | 5.5 个工作日 | 2019年9月22日 | 2019年9月29日 |
| 3.1.3 非功能需求 | 1 个工作日 | 2019年9月22日 | 2019年9月22日 |
| **3.1.4 原型设计** | **4 个工作日** | **2019年9月23日** | **2019年9月28日** |
| 登录界面 | 1 个工作日? | 2019年9月23日 | 2019年9月23日 |
| 注册界面 | 1 个工作日? | 2019年9月26日 | 2019年9月26日 |
| 交流界面 | 1 个工作日? | 2019年9月27日 | 2019年9月27日 |
| 用户信息界面 | 1 个工作日? | 2019年9月28日 | 2019年9月28日 |
| 申请提醒界面 | 1 个工作日? | 2019年9月23日 | 2019年9月23日 |
| **3.1.5 编写需求分析文档** | **4 个工作日** | **2019年9月23日** | **2019年9月28日** |
| 3.1.5.1 编写系统功能需求 | 2 个工作日? | 2019年9月23日 | 2019年9月26日 |
| **3.1.5.2 编写系统功能详细描述** | **4 个工作日** | **2019年9月23日** | **2019年9月28日** |
| 登录注册 | 1 个工作日? | 2019年9月23日 | 2019年9月23日 |
| 好友管理 | 1 个工作日? | 2019年9月23日 | 2019年9月23日 |
| 即时交流 | 1 个工作日? | 2019年9月26日 | 2019年9月26日 |
| 查看病例 | 1 个工作日? | 2019年9月27日 | 2019年9月27日 |
| 3.1.5.3 编写系统非功能需求 | 2.5 个工作日? | 2019年9月26日 | 2019年9月28日 |
| 3.1.5.4 编写系统接口 | 1 个工作日? | 2019年9月28日 | 2019年9月28日 |
| 3.1.5.5 编写术语表 | 0.5 个工作日? | 2019年9月28日 | 2019年9月28日 |
| 3.1.6 需求分析结束 | 0 个工作日 | 2019年9月28日 | 2019年9月28日 |
| 3.1.7 需求分析评审 | 1 个工作日? | 2019年9月29日 | 2019年9月29日 |
| **3.2概要设计** | **9 个工作日** | **2019年10月6日** | **2019年10月15日** |
| 3.2.1 系统E-R图设计 | 3 个工作日 | 2019年10月6日 | 2019年10月8日 |
| 3.2.2 系统逻辑架构设计 | 2 个工作日 | 2019年10月6日 | 2019年10月7日 |
| 3.2.3 系统物理架构设计 | 4 个工作日 | 2019年10月6日 | 2019年10月10日 |
| 3.2.4 系统接口设计 | 4 个工作日 | 2019年10月11日 | 2019年10月14日 |
| 3.2.5 约定与假设 | 3 个工作日 | 2019年10月6日 | 2019年10月8日 |
| 3.2.6 整合系统数据模型文档 | 2 个工作日 | 2019年10月6日 | 2019年10月7日 |
| 3.2.6 编写概要设计文档 | 3 个工作日 | 2019年10月12日 | 2019年10月14日 |
| 3.2.7 概要设计阶段结束 | 0 个工作日 | 2019年10月12日 | 2019年10月12日 |
| 3.2.8 概要设计评审 | 1 个工作日 | 2019年10月15日 | 2019年10月15日 |
| **3.3详细设计** | **16 个工作日** | **2019年10月12日** | **2019年10月29日** |
| 3.3.1 系统简述 | 1 个工作日? | 2019年10月13日 | 2019年10月13日 |
| 3.3.2 术语表 | 1 个工作日? | 2019年10月13日 | 2019年10月13日 |
| **3.3.3 设计概述** | **12 个工作日** | **2019年10月13日** | **2019年10月26日** |
| 3.3.3.1 系统的复用计划 | 2 个工作日 | 2019年10月14日 | 2019年10月15日 |
| 3.3.3.2 系统的接口设计 | 2 个工作日 | 2019年10月13日 | 2019年10月14日 |
| 3.3.3.3 对象模型设计（类图设计） | 7 个工作日 | 2019年10月15日 | 2019年10月22日 |
| **客户端** | **7 个工作日** | **2019年10月31日** | **2019年11月7日** |
| **登录注册模块** | **5.5 个工作日?** | **2019年10月31日** | **2019年11月5日** |
| **登录** | **1 个工作日?** | **2019年10月31日** | **2019年10月31日** |
| 布局xml | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| Activity | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| **注册** | **1 个工作日?** | **2019年10月31日** | **2019年10月31日** |
| 布局xml | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| Activity | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| **找回密码** | **5.5 个工作日?** | **2019年10月31日** | **2019年11月5日** |
| 布局xml | 1.5 个工作日? | 2086年11月4日 | 2019年11月5日 |
| Activity | 3 个工作日? | 2019年10月31日 | 2019年11月2日 |
| **医师选择模块** | **1 个工作日?** | **2019年10月31日** | **2019年10月31日** |
| **选择成功** | **1 个工作日?** | **2019年10月31日** | **2019年10月31日** |
| 布局xml | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| Activity | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| **取消挂号** | **1 个工作日?** | **2019年10月31日** | **2019年10月31日** |
| 布局xml | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| Activity | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| **即时交流模块** | **2 个工作日?** | **2019年11月5日** | **2019年11月7日** |
| 布局xml | 1 个工作日? | 2019年11月7日 | 2019年11月7日 |
| Activity | 1 个工作日? | 2019年11月5日 | 2019年11月5日 |
| **病例分享模块** | **1 个工作日?** | **2019年10月31日** | **2019年10月31日** |
| 布局xml | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| Activity | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| Socket | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| 对象类 | 1 个工作日? | 2019年10月31日 | 2019年10月31日 |
| **服务器** | **7 个工作日** | **2019年10月31日** | **2019年11月7日** |
| 对象类 | 2.5 个工作日? | 2019年10月31日 | 2019年11月2日 |
| 数据库接口 | 3 个工作日? | 2019年11月2日 | 2019年11月4日 |
| 数据库配置文件 | 1 个工作日? | 2019年11月4日 | 2019年11月4日 |
| Socket | 1.5 个工作日? | 2019年11月5日 | 2019年11月7日 |
| **类描述** | **5 个工作日** | **2019年11月9日** | **2019年11月13日** |
| **客户端** | **2 个工作日?** | **2019年11月9日** | **2019年11月10日** |
| 逻辑类 | 1 个工作日? | 2019年11月10日 | 2019年11月10日 |
| 对象类 | 2 个工作日? | 2019年11月9日 | 2019年11月10日 |
| Socket类 | 1 个工作日? | 2019年11月9日 | 2019年11月9日 |
| **服务器** | **2 个工作日?** | **2019年11月9日** | **2019年11月10日** |
| 逻辑类 | 1 个工作日? | 2019年11月9日 | 2019年11月9日 |
| 对象类 | 1 个工作日? | 2019年11月9日 | 2019年11月9日 |
| Socket类 | 1 个工作日? | 2019年11月10日 | 2019年11月10日 |
| 3.3.3.4 系统用例实现详细设计 | **2 个工作日** | **2019年10月20日** | **2019年10月22日** |
| 顺序图 | **6 个工作日** | **2019年11月4日** | **2019年11月10日** |
| 登录 | 1 个工作日? | 2019年11月4日 | 2019年11月4日 |
| 注册 | 1.5 个工作日? | 2019年11月5日 | 2019年11月7日 |
| 选择医师 | 1 个工作日? | 2019年11月4日 | 2019年11月4日 |
| 取消挂号 | 1.5 个工作日? | 2019年11月5日 | 2019年11月7日 |
| 病例分享 | 1 个工作日? | 2019年11月7日 | 2019年11月7日 |
| 聊天 | 2.5 个工作日? | 2019年11月8日 | 2019年11月10日 |
| 3.3.3.5 系统非功能设计 | 5 个工作日? | 2019年10月13日 | 2019年10月18日 |
| 3.3.4 编制详细设计文档 | **4 个工作日** | **2019年10月26日** | **2019年10月29日** |
| 文档整合 | 1 个工作日? | 2019年10月26日 | 2019年10月26日 |
| 格式调整 | 1 个工作日? | 2019年10月26日 | 2019年10月26日 |
| 审核内容 | 1 个工作日? | 2019年10月26日 | 2019年10月26日 |
| 3.3.5 详细设计阶段结束 | 1 个工作日 | 2019年10月29日 | 2019年10月29日 |
| 3.3.6 详细设计评审 | 0 个工作日? | 2019年10月29日 | 2019年10月29日 |
| **3.4编码阶段** | **37 个工作日** | **2019年10月26日** | **2019年12月9日** |
| **客户端** | **35 个工作日** | **2019年10月26日** | **2019年12月7日** |
| **登录** | **34 个工作日** | **2019年10月26日** | **2019年12月6日** |
| 编写布局文件 | 3 个工作日 | 2019年10月26日 | 2019年10月28日 |
| 在Activity中注册布局组件 | 7 个工作日? | 2019年10月26日 | 2019年11月2日 |
| 绑定点击监听器 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| 取出、传递参数（消息类） | 25 个工作日? | 2019年10月26日 | 2019年11月23日 |
| 消息发给服务器 | 25 个工作日? | 2019年10月26日 | 2019年11月23日 |
| 服务器返回信息判断 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| **注册** | **26 个工作日** | **2019年10月26日** | **2019年11月24日** |
| 编写布局文件 | 3 个工作日? | 2019年10月26日 | 2019年10月28日 |
| 在Activity中注册布局组件 | 7 个工作日? | 2019年10月27日 | 2019年11月3日 |
| 绑定点击监听器 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| 取出、传递参数（消息类） | 5 个工作日? | 2019年10月27日 | 2019年11月1日 |
| 消息发给服务器 | 9 个工作日? | 2019年10月27日 | 2019年11月5日 |
| 服务器返回信息判断 | 24 个工作日 | 2019年10月26日 | 2019年11月22日 |
| **选择医师** | **34 个工作日** | **2019年10月26日** | **2019年12月6日** |
| 编写布局文件 | 3 个工作日? | 2019年10月26日 | 2019年10月28日 |
| 在Activity中注册布局组件 | 7 个工作日? | 2019年10月27日 | 2019年11月3日 |
| 绑定点击监听器 | 5 个工作日? | 2019年10月27日 | 2019年11月1日 |
| 取出、传递参数（消息类） | 13 个工作日? | 2019年10月27日 | 2019年11月10日 |
| 消息转发客户端 | 25 个工作日? | 2019年10月27日 | 2019年11月24日 |
| 服务器返回信息判断 | 21 个工作日 | 2019年10月26日 | 2019年11月17日 |
| 将返回的医师信息状态显示在界面中 | 3.5 个工作日? | 2019年10月26日 | 2019年10月29日 |
| **删除好友** | **34 个工作日** | **2019年10月26日** | **2019年12月6日** |
| 编写布局文件 | 3 个工作日? | 2019年10月26日 | 2019年10月28日 |
| 在Activity中注册布局组件 | 6 个工作日? | 2019年10月27日 | 2019年11月2日 |
| 绑定点击监听器 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| 取出、传递参数（消息类） | 25 个工作日? | 2019年10月27日 | 2019年11月24日 |
| 消息发给服务器 | 21 个工作日 | 2019年10月26日 | 2019年11月17日 |
| 服务器返回信息判断 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| 从界面中结束挂号 | 2 个工作日? | 2019年10月26日 | 2019年10月27日 |
| **即时聊天** | **32 个工作日** | **2019年10月26日** | **2019年12月2日** |
| 编写布局文件 | 3 个工作日? | 2019年10月26日 | 2019年10月28日 |
| 在Activity中注册布局组件 | 7 个工作日? | 2019年10月27日 | 2019年11月3日 |
| 绑定点击监听器 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| 取出、传递参数（消息类） | 25 个工作日? | 2019年10月27日 | 2019年11月24日 |
| 消息发给服务器 | 25 个工作日? | 2019年10月27日 | 2019年11月24日 |
| 服务器返回信息判断 | 30 个工作日? | 2019年10月27日 | 2019年12月1日 |
| **病例分享** | **22 个工作日** | **2019年10月26日** | **2019年11月18日** |
| 编写布局文件 | 3 个工作日? | 2019年10月26日 | 2019年10月28日 |
| 在Activity中注册布局组件 | 7 个工作日? | 2019年10月27日 | 2019年11月3日 |
| 绑定点击监听器 | 21 个工作日? | 2019年10月27日 | 2019年11月18日 |
| **服务器** | **32 个工作日** | **2019年10月26日** | **2019年12月2日** |
| **服务器搭建** | **9 个工作日** | **2019年10月26日** | **2019年11月4日** |
| mybatis.xml配置 | 8 个工作日? | 2019年10月27日 | 2019年11月4日 |
| **数据库** | **4 个工作日** | **2019年10月26日** | **2019年10月29日** |
| user表设计 | 1 个工作日? | 2019年10月28日 | 2019年10月28日 |
| friend表设计 | 1 个工作日? | 2019年10月27日 | 2019年10月27日 |
| **后台实现** | **23 个工作日** | **2019年11月2日** | **2019年11月29日** |
| 数据库访问接口 | 10 个工作日 | 2019年11月1日 | 2019年11月11日 |
| 客户端传来参数的判断 | 10 个工作日 | 2019年11月11日 | 2019年11月22日 |
| **3.5 测试** | **4 个工作日** | **2019年12月7日** | **2019年12月10日** |
| **功能性测试** | **5 个工作日** | **2019年11月21日** | **2019年11月25日** |
| **客户端** | **3 个工作日** | **2019年11月21日** | **2019年11月23日** |
| 登录注册模块 | 1 个工作日? | 2019年11月21日 | 2019年11月21日 |
| 医师选择模块 | 1 个工作日? | 2019年11月21日 | 2019年11月21日 |
| 即时交流模块 | 1 个工作日? | 2019年11月21日 | 2019年11月21日 |
| 结束挂号模块 | 1 个工作日? | 2019年11月21日 | 2019年11月21日 |
| **服务器** | **3 个工作日** | **2019年11月23日** | **2019年11月25日** |
| 接口测试 | 1 个工作日? | 2019年11月23日 | 2019年11月23日 |
| Socket | 1 个工作日? | 2019年11月23日 | 2019年11月23日 |
| **易用性测试** | **3 个工作日** | **2019年11月25日** | **2019年11月29日** |
| 按键测试 | 1 个工作日? | 2019年11月25日 | 2019年11月25日 |
| 界面切换测试 | 1 个工作日? | 2019年11月28日 | 2019年11月28日 |
| **可靠性测试** | **2 个工作日** | **2019年11月23日** | **2019年11月24日** |
| 服务器与客户端连接测试 | 1 个工作日? | 2019年11月24日 | 2019年11月24日 |
| 兼容性测试 | 1 个工作日 | 2019年11月30日 | 2019年11月30日 |
| **安全性测试** | **2 个工作日** | **2019年11月28日** | **2019年11月29日** |
| 测试用户名密码 | 1 个工作日? | 2019年11月28日 | 2019年11月28日 |
| 编制系统测试报告 | 1 个工作日? | 2019年12月5日 | 2019年12月5日 |

表2

# 四．软件架构

### 4.1软件框架

### 4.1.1软件总框架

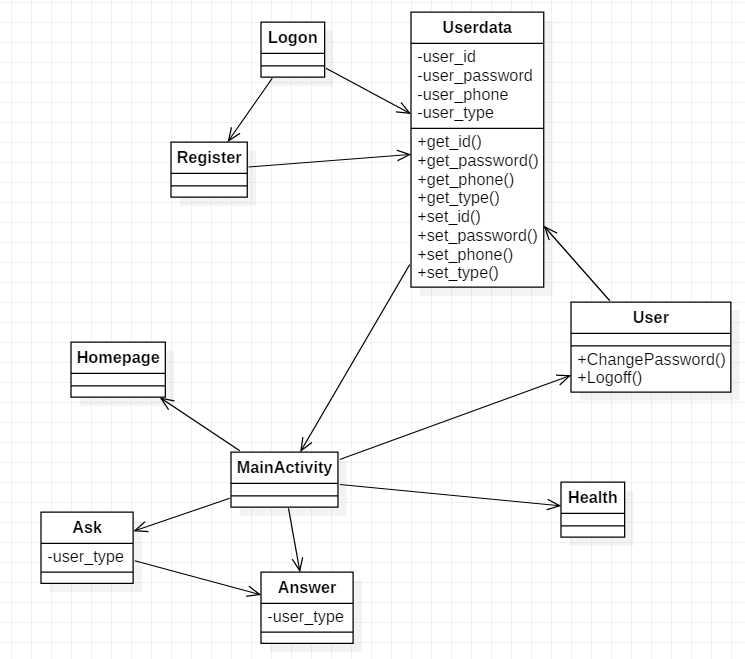


图9

Activity：

图10

Model:

图11

UI

图12

### 4.1.2登录界面跳转到注册界面

Register

Logon

图13

### 4.1.3登录界面跳转到主页面

Homepage

Logon

图14

### 4.1.4 从注册界面跳转到主页面

Homepage

Register

图15

### 4.1.5主页面跳转到看病页面（用户类型为：患者）

Asking

Homepage

图16

### 4.1.6主页面跳转到看病页面（用户类型为：医生）

Answer

Homepage

图17

### 4.1.7主页面跳转到健康知识页面

Health

Homepage

图18

### 4.1.8主页面跳转到用户信息管理页面

User

Homepage

图19

### 4.1.9主页面跳转到主页内容中的 医学新闻推送页面

News1

Homepage

News3

News2

图20

### 4.1.10主页面跳转到主页内容中的 病情分类科普页面

呼吸科（Hxk）

Homepage

眼科（Yk）

……

(共12个科)

图21

### 4.1.11健康知识面跳转到其中的 健康知识详细介绍页面

Health1

Health

……

HealthN

图22

### 4.1.12用户信息管理页面跳转到 用户名、密码、类型更改页面

Userinfo\_Change

User

图23

### 4.2各种类说明

### 4.2.1 Launcher类

Launcher类

|  |  |
| --- | --- |
| 类名 | Launcher |
| 说明 | 是整个应用的起始界面 |
| 方法名 | 功能说明 |
| Button btn1 | 登录按钮 |
| Button btn2 | 注册按钮 |
| Textview txv1 | 记录登录id |
| Textview tx2 | 记录输入的密码 |
| RadioGroup | 记录登录的用户类型 |

表3

### 4.2.2 Userdata类

Userdata类

|  |  |
| --- | --- |
| 类名 | Userdata |
| 说明 | 记录用户信息数据，实现不同activity的信息交互 |
| 方法名 | 功能说明 |
| Initial | 初始化用户信息 |
| Set\_id | 记录用户id |
| Set\_password | 记录用户密码 |
| Set\_type | 记录用户类型（医生或患者） |
| Get\_id | 获取用户id |
| Get\_password | 获取密码以进行身份核实 |
| Get\_type | 获取用户类型以提供不同的功能和界面 |

表4

### 4.2.3 News类

News类

|  |  |
| --- | --- |
| 类名 | News |
| 说明 | 展现最新医学新闻 |
| 方法名 | 功能说明 |
| Textview | 新闻文字信息 |
| ImageView | 新闻图片信息 |

表5

### 4.2.4 km类

km类

|  |  |
| --- | --- |
| 类名 | Km |
| 说明 | 各个医学科目的知识科普，和可供识别判断的性状 |
| 方法名 | 功能说明 |
| Textview | 知识文字信息 |
| ImageView | 知识图片信息 |

表6

### 4.2.9 SeeDrController类

SeeDrController类

|  |  |
| --- | --- |
| 类名 | SeeDrController |
| 说明 | 问诊 |
| 方法名 | 功能说明 |
| SelectPt | 选择病人，并自动与病人建立沟通联系 |
| TextView | 显示对话文本框 |
| EditView | 对话输入文本框 |
| Button | 发送信息按钮 |
| User\_id | 显示用户名称 |
| Time | 显示和记录对话时间 |
| Button2 | 完成就诊后将病人从未处理的列表中删除 |

表11

### 4.3资源列表

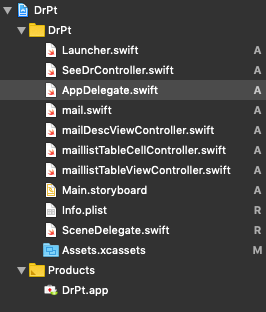


图27

# 五．系统实现

### 5.1登录界面的实现



图28

**注册界面的实现**



图29

**软件主界面**



图30

**点击新闻图片 跳转新闻详细界面**

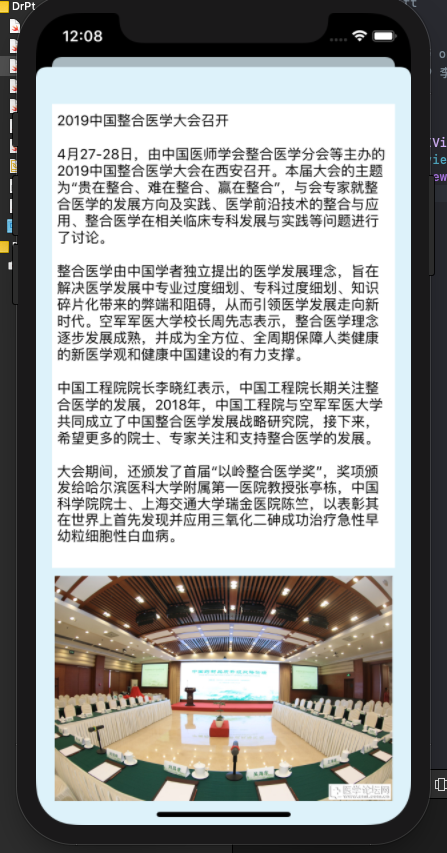


图31

**点击各科目知识介绍，跳转详细介绍界面**

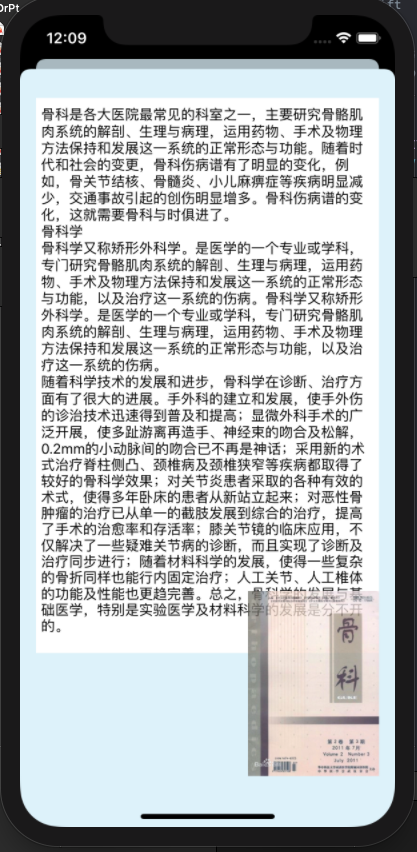


图32

**看病界面**

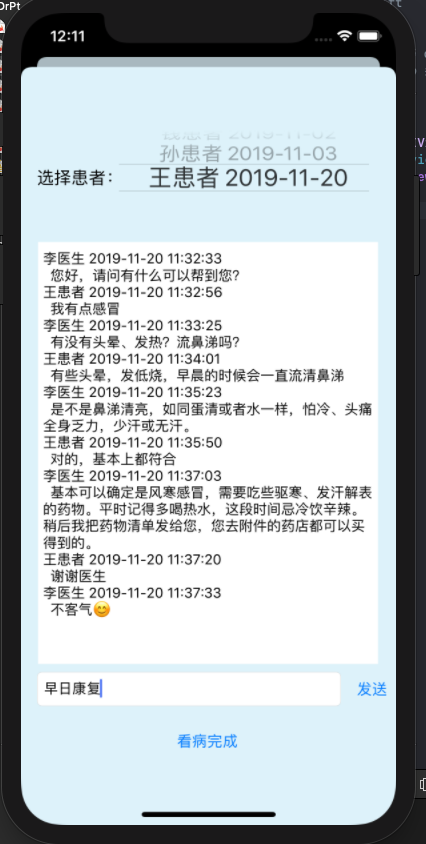


图33

**健康知识推送界面**

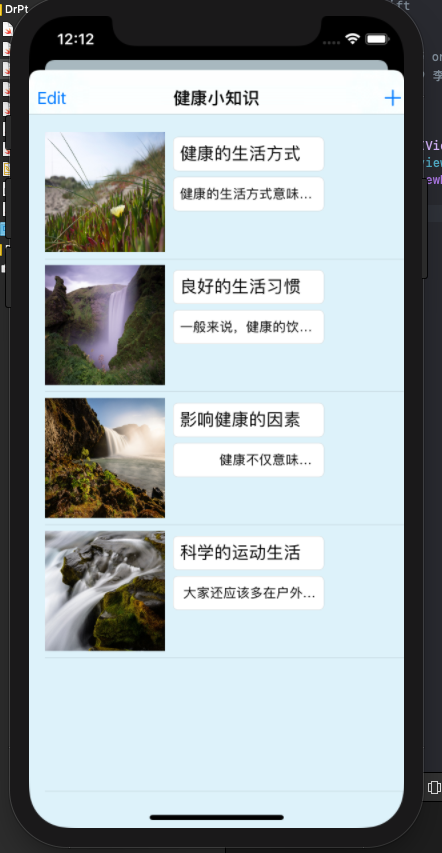


图34

**推送详情界面（发表、编辑健康知识界面）**

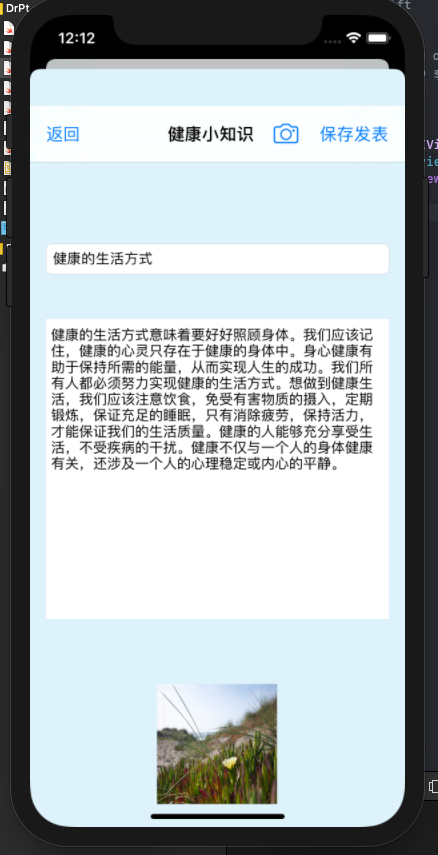


图35

**删除健康知识界面**



图35

**账户信息管理界面**



图36

# 六．系统测试说明

### 6.1测试项目及其说明

### 6.1.1测试方案

分段测试、界面测试、功能测试、性能测试等

### 6.1.2测试项目

测试项目

|  |  |  |  |
| --- | --- | --- | --- |
| 测试名称 | 内容 | 目的 | 进度 |
| 分段测试 | 没结束一个开发阶段后，或修改了后，对项目进行测试 | 早期排除bug，减少最终测试阶段的工作量 | 第13周开始，一周一次测试 |
| 用户界面测试 | 观察界面的成效是否达到预期标准 | 使界面简介易用，能被广大用户接受 | 15周开始测试 |
| 功能测试 | 运行程序，将所有功能通过按钮等实现一遍 | 检查是否已实现预期的功能要求 | 16周开始测试 |
| 性能测试 | 多次不断测试，并记录每个界面和阶段的运行速度 | 找出效率最低的地方，适当进行改进，提升性能 | 17周开始测试 |

表13

### 6.2测试结果

### 6.2.1测试结果及其相应的说明

1、分段测试

许多早期bug都在分段测试时被发现并解决，后期工作量减少，有更多的时间用于程序的优化

2、用户界面测试：

用户界面运行良好，易于上手和操作，适用群体广泛。

3、功能流程测试

功能基本实现

|  |  |
| --- | --- |
| 功能 | 实现情况 |
| 登录功能，是否能够识别登录的账号、密码和账户类型 | 账号信息能被识别并成功登录 |
| 注册功能，注册的账户，其信息是否得到记录保存 | 在个人信息管理处可以看到自己注册时输入的信息，说明信息得到了保存 |
| 各个界面的跳转是否顺利 | 指定界面的跳转没有发生错误 |
| 新闻图片滚动展示功能和点击查看详情功能 | 图片能流畅的自动滚动，但点击图片查看详情时可能出现失灵，更改所设计的点击触摸时间的判断标准，增加了点击灵敏度，但同时削弱了手动滑动图片的功能。 |
| 健康知识推送功能 | 能够正常打开推送 |
| 看病功能 | 病人能够正确选定所要看科和医生，医生能够正确接受病人的请求，两者之间的沟通联系能通过文字交流正常进行 |
| 用户信息管理功能 | 用户能成功更改自己的账户信息 |
| 注销登陆功能 | 用户点击注销后，回到登录界面 |

表14

4、性能测试：

性能良好，程序运行速度较快

**6.3评价**

**优点和不足**

优点：

1. 用户体验良好，流程符合逻辑
2. UI简洁美观
3. 运行性能良好

不足：

1、用户UserData的密码保护机制不完善，虽然private password，但可能通过恶意代码使用public get\_password()获取密码。

1. 信息更新机制不完善，目前只能通过手动升级版本，更新推送等信息。

3、信息交互功能，还在开发中，未设置真正能用于交互的接口。

**未来展望**

通过更加完善程序的相关接口，实现真正的交互。同时医患帮具有大量的潜在用户资源可以发掘。

**测试结论**

本程序可以通过测试

# 七．用户手册

本用户手册为 医患帮 项目的用户手册，目的在于指导用户如何使用本系统。预期参考人员包括用户、测试人员、开发人员、项目经理和需要阅读本报告的项目验收者。

### 7.1系统说明

说明：

a. 医患帮 是一个在线的看病和问诊的平台。主要功能是实现医生和患者的实时交互，意在减少看病麻烦困难的情况，并且为不方便出门去医院的患者提供了良好的看病平台和方式，同时也为医生带来了更多的病情诊断经历和额外的收入。

b. 任务提出者：由项目开发团队共同提出

c. 开发者：李海涛

d. 用户：前期主要先面向医院，给医院提供在线问诊的新途径。之后可以建立详细的医者资格审核方案，推广于所有群众。

e. 服务器：PC单机

运行环境： iOS

运行设备： iPhone、mac、iWatch、iPad

### 7.2用途

### 7.2.1功能

功能

|  |  |
| --- | --- |
| 功能 | 解释 |
| 医学新闻 | 提供最新医疗信息 |
| 医学科目常识 | 提供各科常识，便于患者就诊 |
| 在线看病 | 提供一个可选择和沟通的平台给医生和患者进行交流，实现在线看病 |
| 健康知识推送 | 推送一些健康知识和保健知识 |
| 用户信息管理 | 用户可以更改自己的信息 |

表15

### 7.2.2性能

### 7.2.2.1精度

发送的文字发送能精确显示在对方聊天界面中。推文能够准确更新推送。

### 7.2.2.2时间特性

用户成功登陆到加载界面时间小于1秒

聊天信息发送时间与对方接收时间差小于2秒

### 7.2.2.3灵活性

用户可以在所有主流安卓手机端使用本软件

### 7.2.3安全保密

应用不会透露用户自己填写以外的个人信息

### 7.3使用过程

### 7.3.1安装与初始化

暂供内部测试，之后可能放入App Store

### 7.3.2注册&登陆

按照输入提示，填写信息后完成注册，或者填写已注册过的账号的id和密码

### 7.3.3首页

登陆后默认进入首页，可查看医疗新闻和科普知识

### 7.3.4看病

点击看病按钮进入看病界面

医生可从已收到的请求信息中，选择来进行就诊。患者可以选择所看的科目和医生，开始看病。

### 7.3.5健康知识推文

点击健康知识按钮，可以看到相关推文。

### 7.3.6用户信息管理

点击用户按钮，可以更改账户相关信息

## 八．关键代码

**用NScoding存储健康小知识的表格**

**mail.swift**

//

// mail.swift

/// TableView

//

// Created by 李海涛 on 2019/11/6.

// Copyright © 2019年 李海涛. All rights reserved.

//

**import** os.log

//此为swift文件，非cocotouchfile

**import** UIKit

//NSCoding用于持久化数据，是一种协议protocol

**class** mail: NSObject, NSCoding {

**var** name: String?

**var** mail: String?

**var** photo: UIImage?

//MARK: **Archiving Paths**

**static** **let** DocumentsDirectory = FileManager().urls(for: .documentDirectory, in: .userDomainMask).first!

**static** **let** ArchiveURL = DocumentsDirectory.appendingPathComponent("maillList")

**init**(inputName: String?, inputMail: String?, inputPhoto: UIImage?){

**self**.name = inputName

**self**.mail = inputMail

**self**.photo = inputPhoto

}

// MARK:**NSCoding**

**func** encode(with aCoder: NSCoder) {

aCoder.encode(name, forKey: PropertyKey.name)

aCoder.encode(mail, forKey: PropertyKey.mail)

aCoder.encode(photo, forKey: PropertyKey.photo)

}

//The required modifier means this initializer must be implemented on every subclass, if the subclass defines its own initializers.

//The convenience modifier means that this is a secondary initializer, and that it must call a designated initializer from the same class.

//The question mark (?) means that this is a failable initializer that might return nil.

**required** **convenience** **init**?(coder aDecoder: NSCoder) {

// The name is required. If we cannot decode a name string, the initializer should fail.

**guard** **let** name = aDecoder.decodeObject(forKey: PropertyKey.name) **as**? String **else** {

os\_log("Unable to decode the name for a Mail object.", log: OSLog.default, type: .debug)

**return** **nil**

}

**guard** **let** mail = aDecoder.decodeObject(forKey: PropertyKey.mail) **as**? String **else** {

os\_log("Unable to decode the name for a Mail object.", log: OSLog.default, type: .debug)

**return** **nil**

}

// Because photo is an optional property of Meal, just use conditional cast.

**let** photo = aDecoder.decodeObject(forKey: PropertyKey.photo) **as**? UIImage

// Must call designated initializer.

**self**.init(inputName: name, inputMail: mail, inputPhoto: photo)

}

**struct** PropertyKey {

**static** **let** name = "nameKey"

**static** **let** mail = "mailKey"

**static** **let** photo = "photoKey"

}

}

## 九．附录

### 9.1小组分工

小组分工

|  |  |
| --- | --- |
| 小组分工 | |
| 李海涛 | 组长  小组分工  功能设计  编码  测试  编写文档 |

表16

### 9.2版本记录

版本记录

|  |  |
| --- | --- |
| 版本1.0 | 能实现各个功能的单机版 |

表17

### 9.3设计心得

设计心得

|  |
| --- |
| 李海涛：  通过这次app的设计开发，我学到了很多。尤其在iOS开发的activity的理解和UI界面的学习上有了很大的提升。同时也学会了用科学的方法客观分析一个项目的需求、所要实现的功能，以及对UI界面的要求等等。这次项目的课题——医患帮，我个人认为是一个很符合当今时代互联网发展信息化的主题的，同时我也觉得这种“互联网+看病”的新模式是对传统模式的一种创新的应用，具有很广阔的前进。虽然由于技术能力问题，现阶段的项目成果一般，但我会继续试着在这方面去学习。 |
|  |

表18

### 9.4参考文献

1.刘铭：《iOS 10 快速开发》 北京，机械工业出版社，2017

2.管蕾 :《iOS 10 开发指南》 北京，人民邮电出版社 2017