**武汉大学计算机学院**

**本科生课程设计报告**

**“医点通”健康监测系统总体设计与实现**

专 业 名 称 ：软件工程

课 程 名 称 ：软件工程概论

指 导 教 师 一：王健 讲师

团 队 成 员 一：李昊（2020302111283）

团 队 成 员 二：钟宝亮（2020302111309）

团 队 成 员 三：尹梓康（2020302111233）

团 队 成 员 四：丁爽（2020302111116）

团 队 成 员 五：杨童舟（2020302111027）

团 队 成 员 六：王靖雄（2020302111399）

二○二二年五月

**郑 重 声 明**

本团队呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

团队成员签名： 日期：2022/5/15

目 录

[1.项目概述 4](#_Toc17018)

[1.1 项目需求 4](#_Toc19777)

[1.2 项目内容 4](#_Toc10533)

[1.3 项目技术 4](#_Toc2881)

[2.软件开发计划 5](#_Toc1092)

[2.1 软件开发方法 5](#_Toc31016)

[2.2 项目管理方法 5](#_Toc232)

[2.3 开发时间表 5](#_Toc25553)

[3.需求分析 5](#_Toc1095)

[3.1 患者 5](#_Toc24642)

[3.1.1登录、注册和管理账号信息 5](#_Toc2415)

[3.1.2 建立身份健康卡 6](#_Toc14078)

[3.1.3 定期测量数据 6](#_Toc20922)

[3.1.4 选择医生 6](#_Toc31550)

[3.1.5 消息和通知 6](#_Toc9658)

[3.2 医生 6](#_Toc13464)

[3.2.1 登录、注册和管理账号信息 6](#_Toc4795)

[3.2.2 管理患者 7](#_Toc30479)

[3.2.3 消息和通知 7](#_Toc12550)

[3.3 管理员 7](#_Toc25785)

[3.3.1 登录、注册和管理个人账号信息 7](#_Toc8586)

[3.3.2 管理其他账号信息 7](#_Toc11595)

[3.3.3 消息和通知 7](#_Toc9630)

[4.概要设计 8](#_Toc845)

[4.1 UML用例图 8](#_Toc2549)

[4.2 UML类图 8](#_Toc31697)

[4.3 UML顺序图 8](#_Toc9716)

[4.4 草图设计 8](#_Toc13522)

[5.第一轮迭代综合报告 11](#_Toc31507)

[5.1 敏捷冲刺每日报告 11](#_Toc16303)

[5.1.1 团队成员具体工作：如下表所示。 11](#_Toc7152)

[5.2.2 发布项目燃尽图：如下图所示。 14](#_Toc30813)

[5.2 Alpha版本测试报告 14](#_Toc309)

[5.2.1 Bug列表：如下表所示 14](#_Toc30531)

[5.2.2 场景测试，包括一下内容。 15](#_Toc25224)

[5.2.3 测试矩阵，如下表所示 16](#_Toc31454)

[5.2.4 出口条件 16](#_Toc7194)

[5.3 Alpha版本发布说明 16](#_Toc20384)

[5.3.1 版本功能 16](#_Toc11665)

[5.3.2 对运行环境的要求 17](#_Toc12196)

[5.3.3 安装方法 18](#_Toc1378)

[5.3.4 描述系统一直的问题和限制 18](#_Toc8498)

[5.3.5说明软件的发布方式以及发布地址 18](#_Toc9173)

[5.4 Alpha阶段小结 18](#_Toc21028)

[6. 第二轮迭代综合报告 20](#_Toc15173)

[6.1 敏捷冲刺每日报告 20](#_Toc8102)

[6.1.1 团队成员具体工作：如下表所示。 20](#_Toc24421)

[6.1.2 发布项目燃尽图：如下图所示。 25](#_Toc13558)

[6.2 Beta版本测试报告 26](#_Toc9093)

[6.2.1 Bug列表：如下表所示 26](#_Toc16644)

[6.2.2 场景测试，包括以下内容。 27](#_Toc18481)

[6.2.3 测试矩阵，如下表所示 28](#_Toc544)

[6.2.4 出口条件 30](#_Toc23201)

[6.3 Beta版本发布说明 30](#_Toc8222)

[6.3.1 版本功能 30](#_Toc31957)

[6.3.2 对运行环境的要求 33](#_Toc20631)

[6.3.3 安装方法 34](#_Toc844)

[6.3.4 描述系统一直的问题和限制 34](#_Toc24110)

[6.3.5 说明软件的发布方式以及发布地址 34](#_Toc15798)

[6.4 Beta阶段小结 34](#_Toc13059)

[7.个人贡献、学习收获和教训总结 36](#_Toc10781)

[7.1 个人贡献 36](#_Toc18363)

[7.2 学习收获与教训总结 36](#_Toc3662)

[7.2.1 组长 李昊 36](#_Toc9155)

[7.2.2 组员 钟宝亮 38](#_Toc4488)

[7.2.3 组员 尹梓康 39](#_Toc16572)

[7.2.4 组员 杨童舟 39](#_Toc196)

[7.2.5 组员 丁爽 40](#_Toc29233)

[7.2.6 组员 王靖雄 41](#_Toc31245)

1.项目概述

1.1 项目需求

如今，越来越多的人习惯于坐在椅子上学习，工作和娱乐，瘫坐的时间越来越长，而运动的时间却越来越少，身体情况也每况愈下；还有的人整天加班加点，为了工作奔波操劳而透支了自己的健康；还有那些独居在家的老人，希望自己可以不用每次跑去医院做检查；在疫情时代下，也有很多人被迫隔离在家，行动受限，也迫切需要线上的设备来关注自己的健康状况。由此，“医点通”健康监测系统应运而生。

1.2 项目内容

“医点通”健康监测系统是一个可以及时监测用户健康情况并提供改善方式的网络平台，用户可以每天上报自己的健康数据，系统会根据情况进行异常监测，用户也可以选择响应的医生，由医生人工判断用户的健康状况，并且根据健康信息的交流记录为患者开病历和处方，供用户参考。

1.3 项目技术

本项目设计为Web项目，采用前后端分离的思想，前后端分别开发，前端所需的技术栈为HTML，CSS,JavaScript和Vue框架，后端所需的技术栈为Spring boot，MyBatis和MySQL数据库，前后端交互使用Swagger进行对接。项目整体结构使用Maven工具构建。

2.软件开发计划

2.1 软件开发方法

本项目采用敏捷开发方式进行开发，计划进行两轮迭代开发。

2.2 项目管理方法

本项目采用Scrum项目管理方法，每一次冲刺周期为一个迭代开发时长，预测并执行每个周期可以完成的功能，每天进行简短会议来对进度进行审评，并及时对进度变化做出调整。

2.3 开发时间表

本项目从学期第五周开始，到十二周结束，共八周。

1. 第五周——第七周：各组员对所需的技术进行学习，每隔一段时间集中进行汇报和探讨。同时对项目的具体需求进行分析，根据学习的知识探讨功能实现的可能性。
2. 第八周：对项目构成进行分析，利用用户故事做出UML类图，用例图和顺序图。
3. 第九周：构建数据库和项目框架，开始正式编写代码，完成第一轮迭代计划——完成用户的登录注册功能。
4. 第十周——第十二周：完成第二轮迭代计划——基本完成项目的所有功能。对完成功能进行测试，并准备汇报。

3.需求分析

本项目采用用户故事的方式方式进行需求分析。需求分析通过患者、医生和社区管理员三个身份进行阐述。

3.1 患者

3.1.1登录、注册和管理账号信息

患者在刚刚打开网页页面时会弹出登录界面，患者可以选择输入现有的账号和密码进行登录，当患者账号和密码正确时进入主界面，登录失败时会显示相关信息。

如果该患者是第一次使用这一网页，则可以点击注册按钮，显示注册界面，在注册界面需要输入账号名和密码，以及确认密码，当密码和确认密码一致并且账号名和密码符合规范时才能注册成功。注册失败会显示相关的错误信息，注册成功后会回到登录页面，进行登录。

进入主界面后会显示个人信息，可以选择管理账号信息，对账号信息进行修改和注销。

3.1.2 建立身份健康卡

患者需要建立身份健康卡进行相关活动，在账户信息里可以切换身份健康卡。点击相应的身份健康卡进行切换。

患者可以添加身份健康卡，进入身份健康卡填写的界面。患者需要填写相应信息（真实姓名，身份证，民族，地址，详细地址，手机号）并提交。

患者也可以选择管理身份健康卡，对身份健康卡进行移除和重新编辑。

一个账户可以添加多个身份健康卡，姓名和身份证相同的身份健康卡可以属于多个账户。但是没有身份健康卡无法进行活动。

3.1.3 定期测量数据

患者在进入主界面时点击“测量数据”按钮，会来到“测量数据”界面，这里可以显示已做和未做的项目，项目项会显示项目名（默认为具体的数据项）和时间，每个项目是在某一时段需要测量的数据集合，未做的按照时间从早到晚排序，已做的则是按照时间从晚到早排序。未做的项目只有在一定时间范围内（比如前后半小时）才可以编辑。可以点击项目名显示对应的项目界面，里面会显示要测量的数据项和测量形式（图片/数字/文字……），用户在填写完数据项后，可以提交确认按钮，完成数据测量。

3.1.4 选择医生

患者可以在医生名单里选择医生，但是不可以选择名额已满的医生。一名患者在一个时刻只能有一个医生，但是患者可以选择切换医生和移除医生。如果患者长时间没有医生，则管理员会介入询问情况帮助其选择合适的医生。选择医生后可以和医生进行文字交流，患者可以请求医生开数据测量和医嘱（包含药物），但也知识通过文字消息的询问。

3.1.5 消息和通知

患者可以和其他账户的角色进行文字的交流，患者也可以选择不接收消息。

患者可以接收通知，当在规定时间没有进行数据测量时，患者会接到系统自动发送的通知。管理员也可以对患者发布通知，当医生为患者添加新的医嘱时，系统也会为患者发送通知。可以点击通知名进入相应的界面查看详细的信息。

患者可以查看医生给患者写的医嘱，可以点击确认表示确认收到医嘱的信息。

3.2 医生

3.2.1 登录、注册和管理账号信息

医生在刚刚打开网页页面时会弹出登录界面，医生可以选择输入现有的账号和密码进行登录，当医生账号和密码正确时进入主界面，登录失败时会显示相关信息。

如果该医生是第一次使用这一网页，则可以点击注册按钮，显示注册界面，在注册界面需要输入账号名和密码，以及确认密码，并且提供真实姓名，身份证号，民族，手机号以及相关的医生执照和证明，当密码和确认密码一致并且账号名和密码符合规范时才能提交注册。提交失败会显示相关的错误信息。提交成功后会交给给管理员进行验证，如果管理员审核成功，则注册成功，否则注册失败，需要重新注册。在注册成功后才可以进行登录。

进入主界面后会显示个人信息，可以选择管理账号信息，对账号信息进行修改和注销。修改账号信息同样需要管理员的审核。

医生没有身份健康卡，账号就相当于身份健康卡。

3.2.2 管理患者

医生可以选择自己同时管理的患者上限，默认为0（即在最开始不接待患者），医生可以查看自己管理的患者的信息，包括部分健康卡信息，已做的和未做的数据测量记录，之前的医嘱。

医生可以为患者下发数据测量项目，为项目起名并规定测量时间和时间范围，为项目添加具体的数据项，每一个数据项包括数据名称，测量形式，并且可以设定合理范围。医生也可以管理患者未做的项目，对其进行修改和删除。

医生也可以为患者写医嘱，可以写下治疗意见，以及为患者开处方药。并写明生效日期（即刻）和生效时间。

3.2.3 消息和通知

医生可以和其他账户的角色进行文字的交流，医生不可以选择不接收消息。

医生可以接收通知，当患者的测量数据超出合理范围时，系统会自动发送警告信息，医生可以点击通知查看异常的测量项目。否则会发送正常的通知，医生也可以查看项目，通过自身的经验和知识查看患者是否有异常。

3.3 管理员

3.3.1 登录、注册和管理个人账号信息

管理员在刚刚打开网页页面时会弹出登录界面，管理员可以选择输入现有的账号和密码进行登录，当管理员账号和密码正确时进入主界面，登录失败时会显示相关信息。

如果该管理员是第一次使用这一网页，则可以点击注册按钮，显示注册界面，在注册界面需要输入账号名和密码，以及确认密码，并且提供真实姓名，身份证号，民族，手机号以及相关的管理员证明，当密码和确认密码一致并且账号名和密码符合规范时才能提交注册。提交失败会显示相关的错误信息。提交成功后会交给给以注册的管理员进行验证，如果管理员审核成功，则注册成功，否则注册失败，需要重新注册。在注册成功后才可以进行登录。

进入主界面后会显示个人信息，可以选择管理账号信息，对账号信息进行修改和注销。

3.3.2 管理其他账号信息

管理员可以查看所有医生和患者的信息。对于医生（注册和修改）和管理员，管理员可以对其身份信息进行审查，确认无误后可以让注册申请通过，否则可以打回注册申请。

管理员可以对患者和医生的个人信息进行增删改查。

管理员可以查看患者的健康状态数据，对于长时间未测量数据的患者管理员可以使用消息和通知告诉患者进行数据测量。

3.3.3 消息和通知

管理员不能和其他人进行消息的交流，但是可以向未上报数据的患者进行消息通知，来提醒他们上报数据。

4.概要设计

4.1 UML用例图

如图4.1所示。

4.2 UML类图

如图4.2所示。

4.3 UML顺序图

如图4.3所示。

4.4 草图设计

实例如图4.4，4.5所示。全部草图可见“项目草图”文件夹。

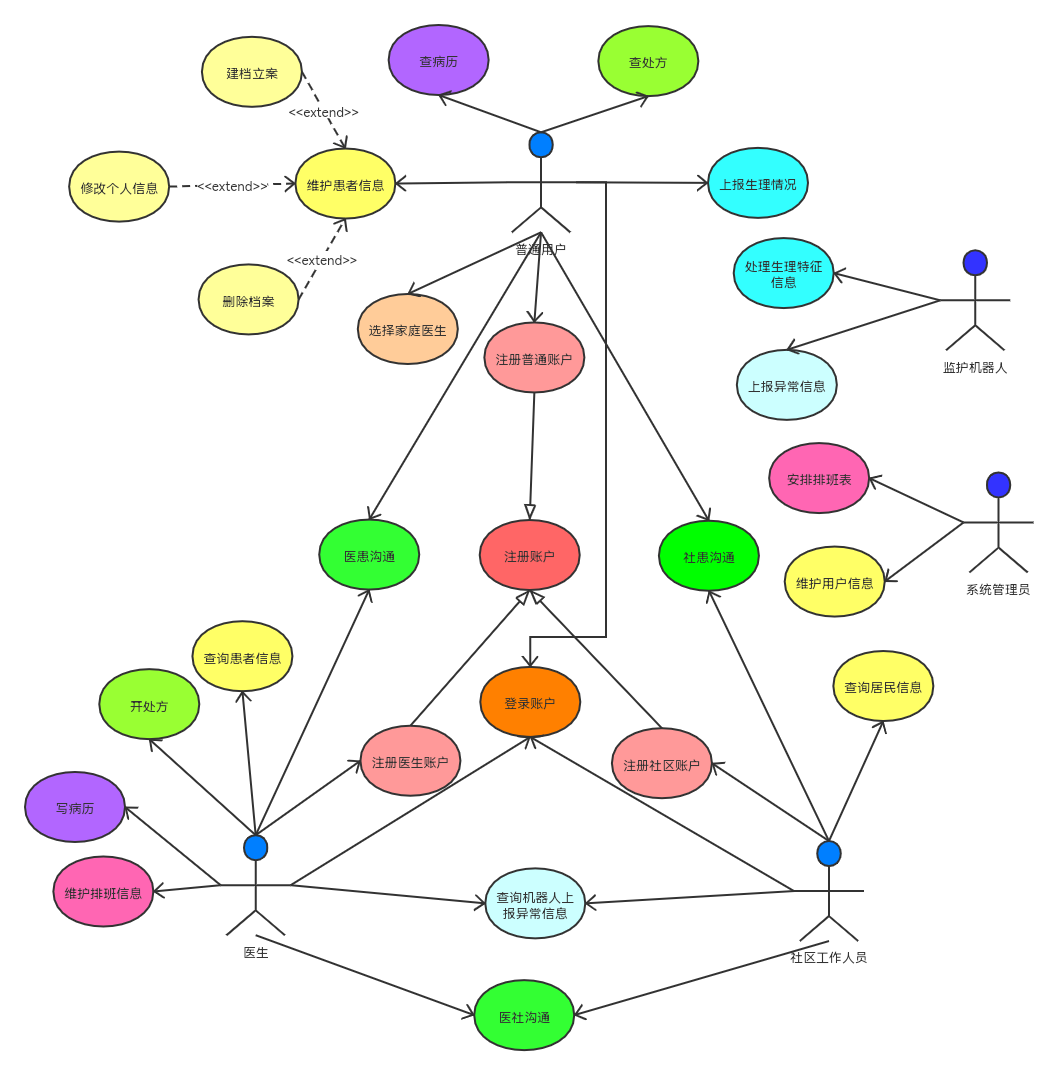


图4.1 用例图

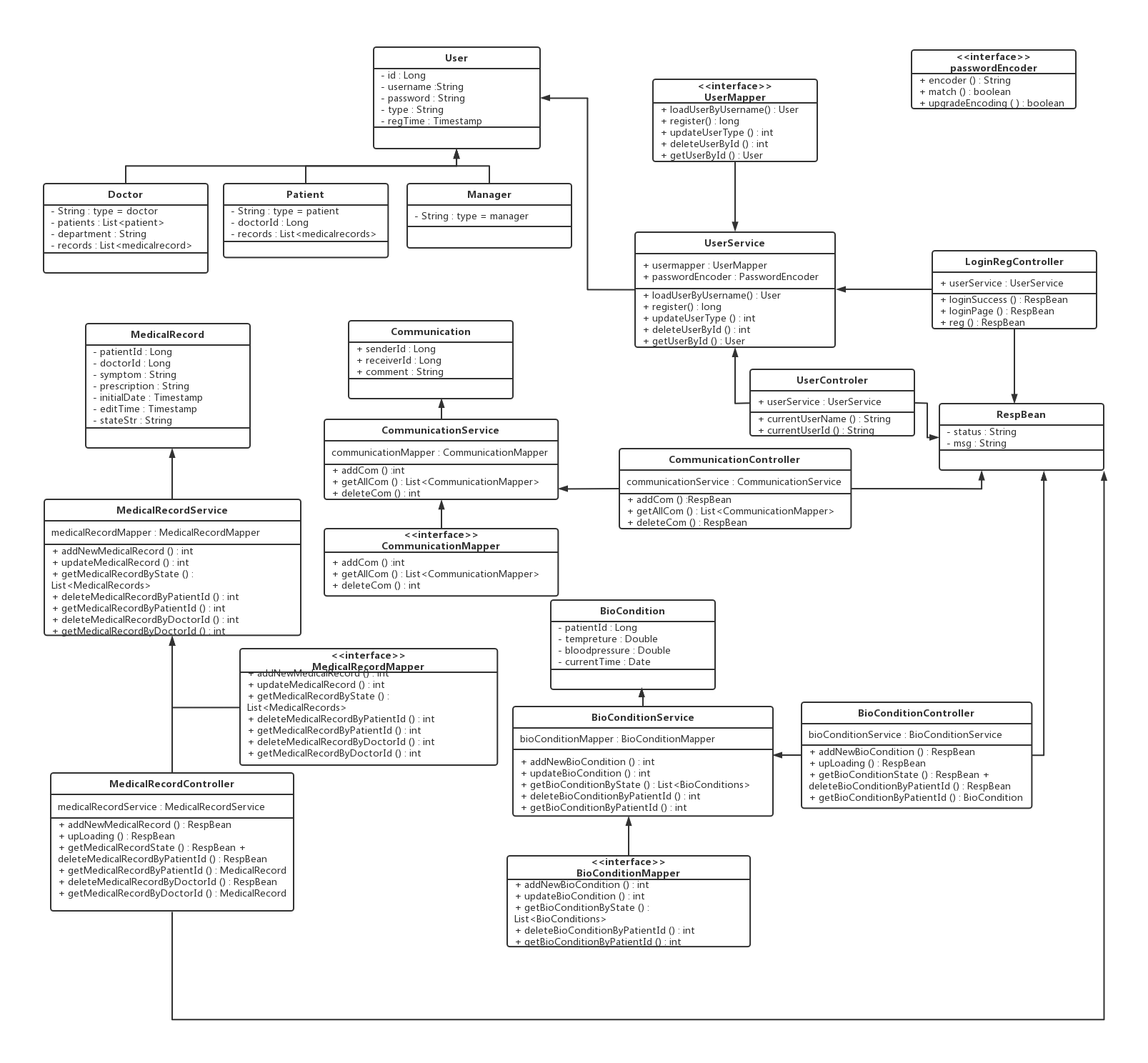


图4.2 类图

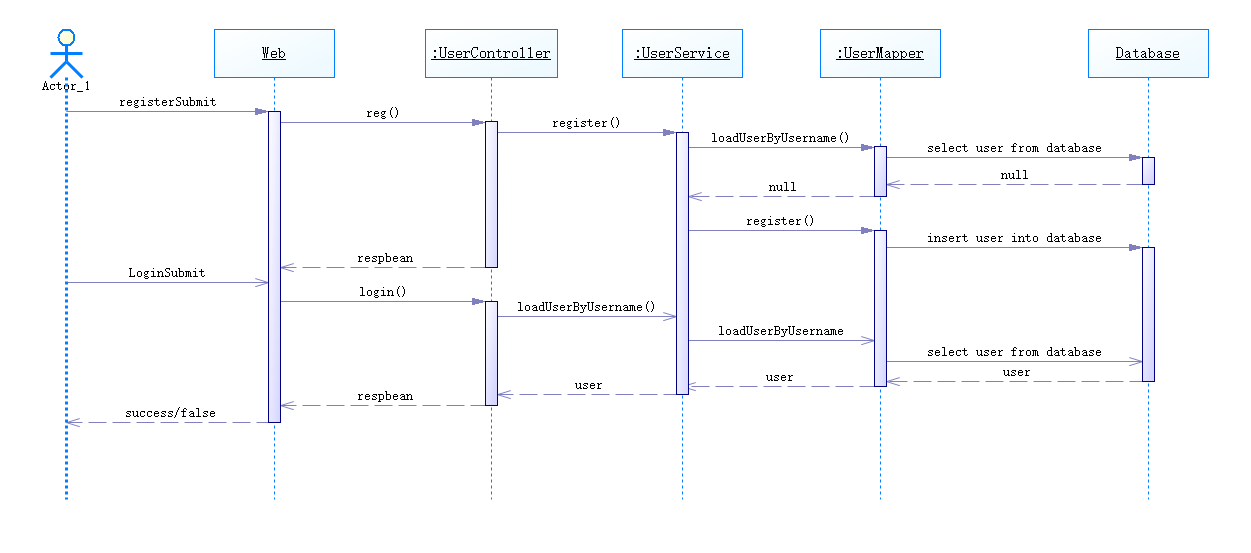


图4.3 顺序图

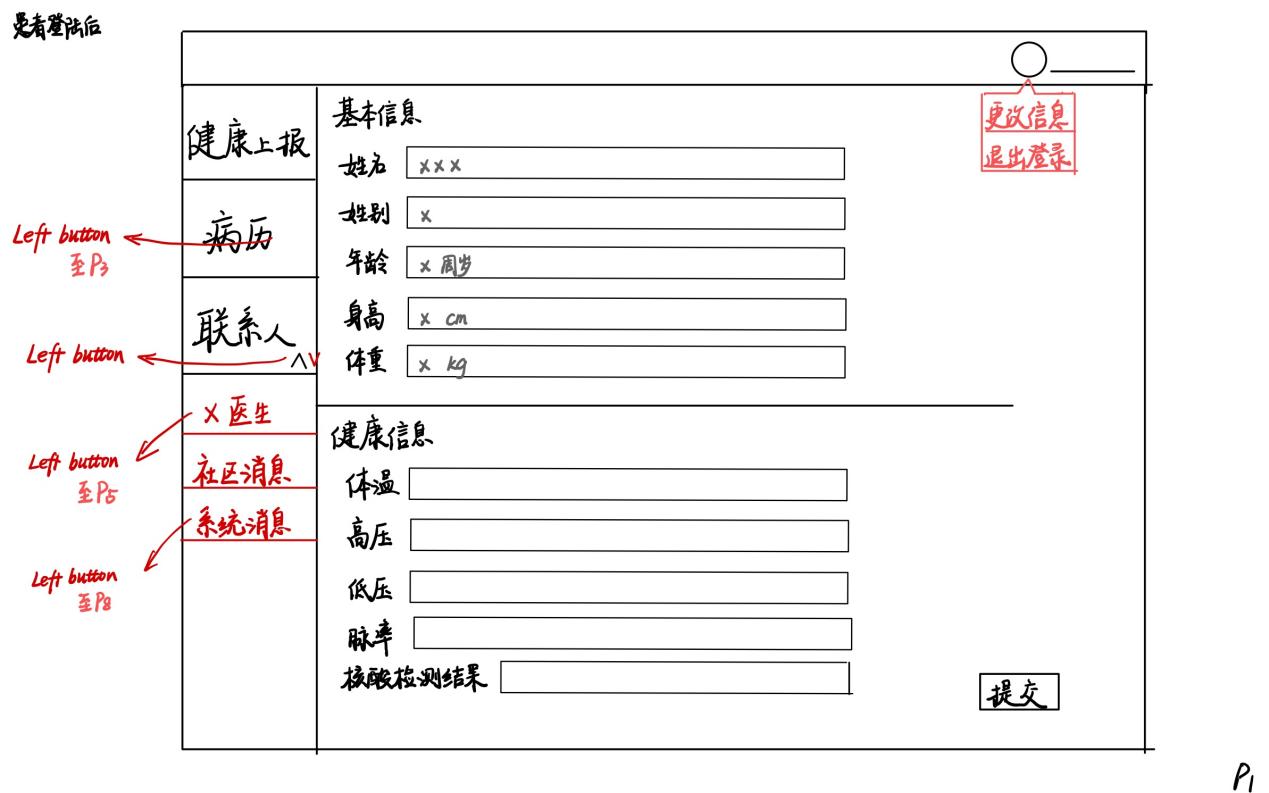


图4.4 草图示例1

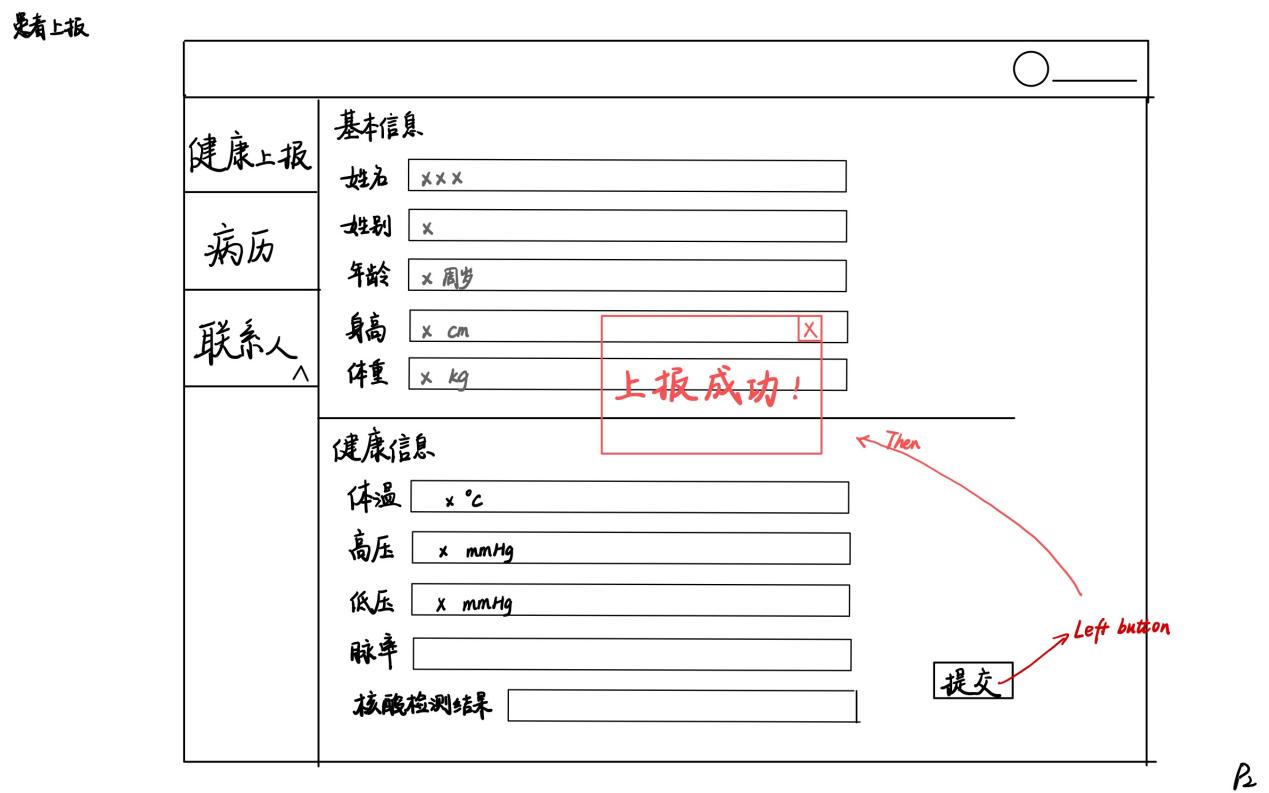


图4.5 草图示例2

5.第一轮迭代综合报告

5.1 敏捷冲刺每日报告

5.1.1 团队成员具体工作：如下表所示。

表5.1.1

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 成员 | 工作 | 困难 |
| 4.11 | 李昊 | 需求的进一步分析和数据库构建分析 | 数据库知识不全面使进展缓慢 |
| 王靖雄 | 尝试搭建数据库，并自主搭建框架 | 无 |
| 丁爽 | 建立对整个项目后端的类图构建，搭建出框架，初步确定系统中的类和功能。 | 对Service和Controller的功能分配因熟悉度不高而难以抉择。 |
| 钟宝亮 | 登录界面设计及结合后端数据的登录逻辑功能实现 | 存在一些漏洞需要完善 |
| 杨童舟 | 根据需求分析确定前端页面功能，并尝试vue框架下的前端页面设计。 | 对vue框架中的路由等功能不熟悉，网页版的社区健康医疗系统可参考的设计不多，有些需求不确定如何呈现。 |
| 尹梓康 | 无（满课） | 无 |
| 4.12 | 李昊 | 远程仓库的构建以及项目框架的基本搭建 | jar包的版本不同导致各种冲突问题 |
| 王靖雄 | 根据讨论意见修改数据库，正式开发 | 需求多次更改造成开发不便 |
| 丁爽 | 学习讨论前后端交互的实现，对bean层实际操作，病历，上报信息，交流等类的初步确立。 | 前后端交互的实现未能解决。 |
| 钟宝亮 | 注册界面设计与基本跳转完善 | 无 |
| 杨童舟 | 利用figma协同开发进行前端的ui设计。 | 无 |
| 尹梓康 | 无（满课） | 无 |
| 4.13 | 李昊 | 解决前一天遗留的冲突问题 | 无 |
| 王靖雄 | 编写User登录注册接口 | 接口存在优化可能 |
| 丁爽 | 学习javaScript，并搭建开发环境。 | 因原编译器无法满足开发的需求，所以需要重新选择编译器。 |
| 钟宝亮 | 注册功能实现与注册数据复用（仅文本部分） | 无 |
| 杨童舟 |  |  |
| 尹梓康 | 完成用户注册页面的框架设计 | 框架技术掌握不熟悉 |
| 4.14 | 李昊 | 完成患者的注册功能 | 无 |
| 王靖雄 | 优化登陆注册接口 | 无 |
| 丁爽 | 病历，交流，上报信息数据库的确立和搭建。 | Java和数据库中相对应的数据类型可能不一致。 |
| 钟宝亮 | 注册图片上传功能研究（暂未实现） | vue版本差异致使难以获得有效的解决办法 |
| 杨童舟 | 研究能否将figma直接转换成vue代码。 | figma转vue代码的插件不一定能将figma项目按照所画的布局进行转化，可能会出现布局混乱 |
| 尹梓康 | 完成医生注册、社区工作者注册的框架设计 | 无 |
| 4.15 | 李昊 | 完成医生和社区管理员的注册已经用户登录功能 | 上传文件方面造成很多困难 |
| 王靖雄 | 编写查看所有患者接口并优化 | 无 |
| 丁爽 | 对xml和数据库相关操作的学习，开始建立mapper层。 | 操作不够熟练。 |
| 钟宝亮 | 普通用户基本信息界面与退出登录跳转实现 | 无 |
| 杨童舟 | 放弃figma，直接画患者身份登录后的操作界面。 | 患者的用户需求较多，如何将信息合理展示，符合人使用的操作习惯。 |
| 尹梓康 | 完善三种注册的框架设计 | 有个别漏洞不能及时处理 |
| 4.16 | 李昊 | 完善后端代码以及熟悉前后端交互操作 | 前端配置出现很大问题 |
| 王靖雄 | 重构数据库，搭建vue环境，测试登陆注册 |  |
| 丁爽 | 建立service层，学习相关操作。 | 之前建立的类图不太适合现在的开发，需要重新分析。 |
| 钟宝亮 | 病例查看界面设计 | 无 |
| 杨童舟 | 画出患者对医生进行的一系列操作，包括选择医生，更换医生，查看医生等。 | 每位医生的患者有人数限制，页面需要在展示医生信息的同时限定医生可选择状态。用户在初次登录时，尚未选择医生，如何进行医生的选择和更换。 |
| 尹梓康 | 为注册页面添加组件，实现整体风格的统一和三种注册切换的功能 | 无 |
| 4.17 | 李昊 | 实现用户的更改和删除操作 | 无 |
| 王靖雄 | 编写显示所有医生各种信息的接口 | 无 |
| 丁爽 | 根据讨论的内容，添加了新的属性，修改之前的代码和数据库。 | 修改个别地方有遗漏不能及时发现。 |
| 钟宝亮 | 健康上报功能界面设计与医生列表查看设计 | 无 |
| 杨童舟 | 画出医生登录后的页面，包括查看患者，开病历，查看健康上报信息，患者交流等。 | 无 |
| 尹梓康 | 为注册页面添加一些方法，并解决输入框有时无法输入的bug | 无 |

5.2.2 发布项目燃尽图：如下图所示。

图5.1.1

5.2 Alpha版本测试报告

5.2.1 Bug列表：如下表所示

表5.2.1

|  |  |  |
| --- | --- | --- |
| 缺陷ID | 主题 | 测试人 |
| 1 | 消息添加方法返回异常但数据库中正常更新 解决：将@Controller注解改为@RestController注解 | 王靖雄 |
| 2 | 输入框嵌套太深导致无法输入内容 解决：使用刷新方法forceUpdate()，按照最新数据进行渲染 | 尹梓康 |
| 3 | 添加表格后，页面出现了超出的内容 解决：1. 将图标属性改成默认样式；2. 重设表格项宽度 | 尹梓康 |
| 4 | 使用Swagger进行数据测试时无反应 解决：在配置中将数据库改成本地的数据库 | 李昊 |
| 5 | 后端代码下载后无法正常运行 解决：相应项目部分添加至根目录 | 钟宝亮 |
| 6 | 常规element plus代码无法使用 解决：资源包应该导在前端部分专门的文件夹里 | 钟宝亮 |

5.2.2 场景测试，包括一下内容。

* 典型用户，如下表所示。

表5.2.2

|  |  |
| --- | --- |
| 姓名 | 张三 |
| 性别 | 男 |
| 年龄 | 25 |
| 职业 | 程序员 |
| 使用目的 | 监测身体健康状况 |
|  |  |
| 姓名 | 李四 |
| 性别 | 女 |
| 年龄 | 31 |
| 职业 | 医生 |
| 使用目的 | 处理数据异常的患者 |
|  |  |
| 姓名 | 王五 |
| 性别 | 男 |
| 年龄 | 27 |
| 职业 | 社区管理员 |
| 使用目的 | 管理该社区的医生和患者 |

* 典型场景

张三是一名程序员，最近上面指派了重要的任务，要求在很短的时间内完成，他和他的小组日夜奋战，任务没有完成，但是张三觉得自己的身体出现了一些问题，在看过医生之后知道了熬夜的风险，但是任务依然要按时完成。于是张三打开健康监测系统网站，通过每天上报数据来监测自己的健康状况。他首先进行了注册，并且登录进入了主页。在登陆之后，他修改了自己的个人信息，并且上传了他的个人头像。然后他进行了监测数据的上报，上报之后发现数据填写有问题，于是又对上报的数据进行了修改。修改成功后，系统提示自己的情况有异常，于是他希望选择一名医生为他解决。最终他在选择列表里选择了人数还未满的李四医生，选择后它通过发消息向医生告知了自己的异常，没过一会张三看到了李四医生为他开的病历，它通过病历吃药调养，并且每日坚持上报数据，终于回归了正常值。

李四是一名医生，负责为选择她的患者提供治疗方案，她首先注册了医生的账户，上传了资格证明，在管理员对她的资格证明验证后，她成功登录进入了主页，不久看到张三选择自己作为他的医生，并且看到他上报的数据有异常，在观察了他上报的数据并且与张三沟通后，为张三开了一份病历。

王五是一名社区管理员，负责管理该社区的患者和医生。一日，在登录账户进入主页之后，他先验证了李四医生的注册请求，之后发现了用户张三今日还没有上报数据，于是提醒张三上报数据，在张三上报数据后发现张三上报的数据异常，因此向张三选择的医生李四提醒她关注张三的异常。

5.2.3 测试矩阵，如下表所示

表5.2.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检查点** | **测试功能** | **完成情况** |
| 登录 | 静态功能 |  | 登录界面布局 |  | 完成 |
|  | 注册按钮 | 点击进入注册页面 | 完成 |
|  | 登录按钮 | 点击登录 | 完成 |
| 动态功能 | 正确的账号和密码 | 确认功能，结果显示 | 进入相应类型用户的主页 | 完成 |
| 错误的账号或密码 | 确认功能，结果显示 | 提示错误信息 | 完成 |
| 注册 | 静态功能 |  | 注册界面布局 |  | 完成 |
|  | 性别单选框 | 点击切换性别 | 完成 |
|  | 确认注册按钮 | 点击进入登录页面 | 完成 |
| 动态功能 | 密码和确认密码不一致 | 确认功能，结果显示 | 提示错误信息 | 完成 |
| 身份证号重复 | 确认功能，结果显示 | 提示错误信息 | 完成 |
| 输入全部正确 | 确认功能，结果显示 | 注册成功，返回登录界面 | 完成 |
| 对非空数据判断 | 确认功能，结果显示 | 提示错误信息 | 未完成 |
| 医生和社区上传资格证明 | 确认功能，结果显示 | 上传照片 | 未完成 |
| 主页 | 静态功能 |  | 导航栏按钮分布 | 可点击 | 完成 |
|  | 主界面信息布局 |  | 完成 |
|  | 右上角账号名 | 移动鼠标出现选项 | 完成 |
| 动态功能 | 个人信息更改 | 确认功能，结果显示 | 成功更改信息 | 完成 |
| 退出登录 | 确认功能，结果显示 | 返回登录界面 | 完成 |

5.2.4 出口条件

界面完整，功能都可以运行，多次测试没有问题

5.3 Alpha版本发布说明

5.3.1 版本功能

* 登录功能，可以完成已注册用户登录的账号检查和提示功能。



图5.3.1

* 注册功能：实现新用户账户的注册，使用身份证号进行唯一标识。

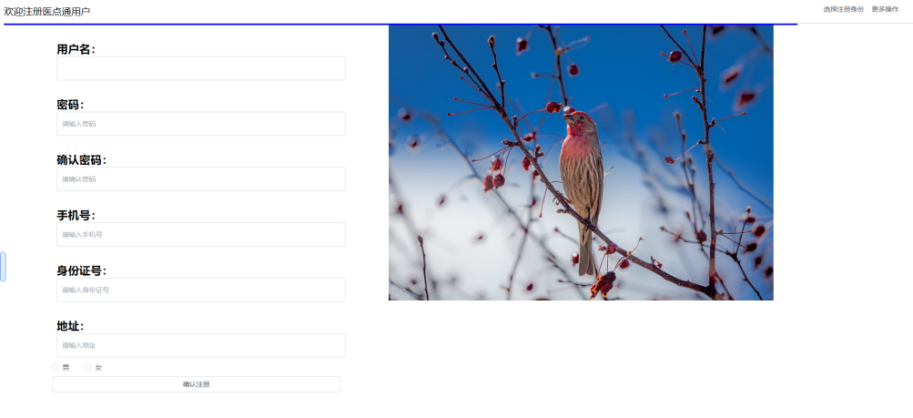


图5.3.2

修改个人信息：对个人信息进行修改

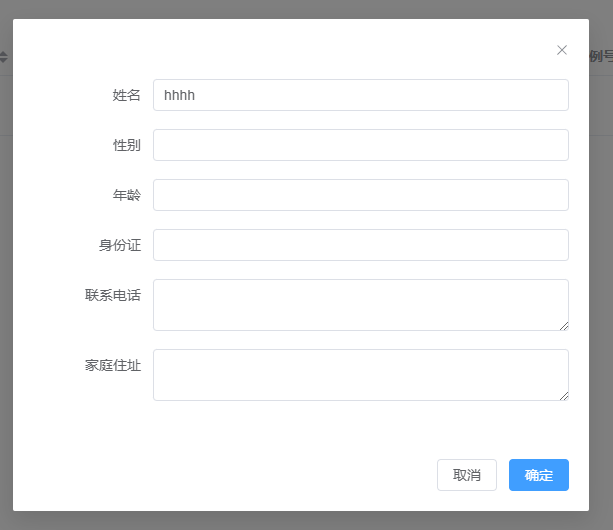


图5.3.3

5.3.2 对运行环境的要求

健康监测系统以网页形式存在，下表针对测试过的浏览器进行说明。

表5.3.1

|  |  |
| --- | --- |
| 火狐FireFox | 使用正常 |
| 谷歌Chrome | 使用正常 |
| Edge | 使用正常 |
| IE浏览器 | 使用正常 |

5.3.3 安装方法

以网页形式存在，无须安装，只需访问网站的网址就可以访问该健康监测网站。

5.3.4 描述系统一直的问题和限制

Alpha版本仅仅完成了各个用户的登录和注册功能，其他正式的功能还没有开发完成，现有的一些细节也需要完善。数据库储存在本地数据库并未发布到远程服务器上，还只能供开发人员调试使用。数据库中还未存储足够数量的数据。

5.3.5说明软件的发布方式以及发布地址

Alpha版本还未发布到远程服务器上，还只能供开发人员本地调试使用。

5.4 Alpha阶段小结

5.4.1 团队的源码仓库地址

<http://125.220.157.107/WorldOneTwo/healthmonitoringsystem.git>

5.4.2 Alpha过程回顾

* 团队项目的目标，预期的典型客户，预期的功能描述，预期的用户数量？

我们团队项目的目标是为身体情况不健康、亚健康的人进行身体情况监测的平台，让他们可以通过每日的数据上报了解自己身体的现状和变化情况，对潜在的疾病防患于未然。

预期的典型用户是对所有自身健康情况有担忧的患者，负责治疗的医生和负责人员管理的社区管理员。

预期功能有：

（1）用户可完成注册登录，个人信息编辑

1. 用户可以进行健康数据的上报和修改。可以查看历史记录
2. 用户可以选择医生，查看医生为他写的病历
3. 用户可以和其他人交流，可以获得系统消息的通知。
4. 医生可以查看选择自己的患者信息，为其开处方
5. 社区管理员可以对别人进行资格审查，对特定的用户进行处理

由于该系统主打健康监测功能，对广大群众都有需求，但缺乏宣传推广渠道，用户在数量可能会有所限制。

* 团队的产品如何满足了用户的需求？

团队的产品满足了用户对于自身健康状况的关心，并且随时随地可以测量数据，比去医院更为方便快捷。

* 用户在Alpha阶段已完成了那些目标？

搭建基本框架，完成了登录注册功能的基本内容，按阶段完成所需的具体文档。

* 团队成员是如何分工协作的？有什么经验教训？

团队对每一阶段的工作任务都按照美味成员的工作能力和强项进行了合理分配，并注重交流合作，每天进行一次线上或线下会议，分享每个人完成的工作，遇到的问题和独到的看法。但是在工作中仍然存在沟通失误的情况，会存在前端界面设计问题和前后但交互冲突问题。在需求分析上也存在问题，没有在前期对需求进行详细的确认，导致成员对于需求的理解不一致，后期在协调需求统一方面浪费了大量时间。于此同时，组员对于开发软件和技术仍然不够熟悉，因此出现的无谓错误也延误了开发进度，导致后期时间紧任务中，给每一位团队成员都造成了不少压力。

* 团队是如何进行项目管理的？

合理分配项目任务，在GitLab上建立团队项目，及时上传改进部分并整合。

* 团队如何平和 时间/质量/资源 争取如期完成任务的？

在项目开发前期，我们对团队成员的技术问题进行了讨论和总结，对某些成员困难的方面给予支持，并确定各个成员擅长的方面，以此对工作内容进行合理的分配。每天进行会议，了解成员的工作情况，对不合理的任务进度做出对应调整。在产品质量上，我们计划首先完成基础功能的实现，在此基础上增加一些功能特色，并在界面上进行美化，达到视觉的赏心悦目。

5.4.3 团队项目的实际进展（见燃尽图），发布的功能（见发布文档）。说明在项目管理中，软禁图是如何真实反应项目的状态的？或者燃尽图美化了状态？

前期由于对框架的不熟悉，需求的不统一以及队员之间的默契度不足，导致项目进展缓慢，后期主键熟悉，能够大体上完成基本功能，但仍有些具体功能未能实现，后期将继续完善。

5.4.4团队成员在Alpha阶段的角色和具体贡献

如下表所示。

表5.4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 角色 | 团队贡献分 | 可验证的贡献 |
| 李昊 | PM&Dev&Test | 20 | 基本框架搭建 |
| 王靖雄 | Dev&Test | 20 | 后台功能代码 |
| 丁爽 | Dev&Test | 20 | 数据库实现 |
| 钟宝亮 | Dev&Test | 20 | 界面UI和功能代码 |
| 杨童舟 | Dev&Test | 20 | 界面设计和界面UI |
| 尹梓康 | Dev&Test | 20 | 界面UI和功能代码 |

5.4.5 Beta阶段需要改进那些方面？请给出下一阶段的展望。

Beta阶段会完成项目预计的功能任务，并进行更详细的测试以及优化处理，修复bug，将Alpha阶段未完成的功能进行完善。

1. 第二轮迭代综合报告

6.1 敏捷冲刺每日报告

6.1.1 团队成员具体工作：如下表所示。

表6.1.1

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **成员** | **工作** | **困难** |
| 4.18 | 李昊 | 完善用户登录和注册界面 | 注册登录的验证逻辑复杂 |
| 王靖雄 | 编写查看所有患者接口并优化 |  |
| 丁爽 | 建立新的病历药品数据库，并完成bean层代码的书写。 | 要对之前的各个层的代码进行修改，也要提醒其他成员数据库的修改。 |
| 钟宝亮 | 联系人前端呈现思路理清 | 存储全局变量有问题 |
| 杨童舟 | 完成草图的全部剩余部分，完善页面跳转顺序等逻辑以及部分页面布局调整 |  |
| 尹梓康 | 完成医生的侧边栏和头部 | 无 |
| 4.19 | 李昊 | 实现三种用户的更近和删除 | 无 |
| 王靖雄 | 重构数据库，搭建vue环境，测试登陆注册 | 配置前端出现问题 |
| 丁爽 | 修改检查之前的代码，因为类的属性有变化，有的地方的方法和参数都要修改。 | 无 |
| 钟宝亮 | 采用store存储全局变量基本原理搞清 | 无 |
| 杨童舟 | 尝试完成社区工作者登录后的页面，能够显示社区中的患者列表。 | 如何在页面上显示全部患者的信息，并在每行右侧增加操作按钮。 |
| 尹梓康 | 完成医生总览病患页面的大体框架 | 对表格的用法还不太熟悉 |
| 4.2O | 李昊 | 添加健康数据功能的实现 | 上报时间的存储有问题 |
| 王靖雄 | 编写显示所有医生各种的接口 | 可能存在缺漏的接口 |
| 丁爽 | 建立新的药品对象的mapper层，和添加药品功能。 | 无 |
| 钟宝亮 | 医生列表呈现 | 无 |
| 杨童舟 | 尝试患者列表中的信息显示，以及患者列表的操作栏中的按钮显示 | 患者列表所用的患者信息数组数据如何保存和显示 |
| 尹梓康 | 完成医生查看某病患的病历页面的大体框架 | 无 |
| 4.21 | 李昊 | 解决健康数据中上报时间的存储问题 | 无 |
| 王靖雄 | 编写消息数据库 | 无 |
| 丁爽 | 病历上开药方功能出现了bug，进行修改和解决。 | 开病历时病历还未生成，没有病历号，则药品添加时无法匹配。 |
| 钟宝亮 | 医生详情页面实现 | 无 |
| 杨童舟 | 完成社区工作者查看今日异常患者的页面 | 无 |
| 尹梓康 | 在查看病历页面添加了查看健康上报功能 | 无 |
| 4.22 | 李昊 | 完成管理员的三个查询用户操作 | SQL语句逻辑实现有难度 |
| 王靖雄 | 编写两个用户之间发送消息的接口 | 无 |
| 丁爽 | 再次修改数据库，新建一个数据库表病历-药品，来解决病历和药品的匹配关系，解决了向病历添加药品的功能和bug。 | 又要大修代码。 |
| 钟宝亮 | 医生有关能容优化及聊天界面设计 | 消息的接收与实时同步 |
| 杨童舟 | 完成社区工作者查看今日未上报健康状况的患者界面 | 社区工作者对尚未上报健康状况的患者有一键提醒功能，需要社区工作者和患者之间的交流功能 |
| 尹梓康 | 完成处方页面的大体框架 | 对按钮的用法还不太熟悉 |
| 4.23 | 李昊 | 完成判断医生选择情况的人数 | 无 |
| 王靖雄 | 测试两个用户之间发送消息的接口 | 无 |
| 丁爽 | 完成用户查看缩略病历功能。 | 无 |
| 钟宝亮 | 聊天记录逻辑与核心代码呈现 | 无 |
| 杨童舟 | 完成社区工作者信息审核界面 | 无 |
| 尹梓康 | 在医生头部添加了编辑信息和退出登录功能 | 无 |
| 4.24 | 李昊 | 完成一次响应传递多个不同类型对象 | 自己探索花费不少时间 |
| 王靖雄 | 修改消息数据库编写的问题 | 无 |
| 丁爽 | 完成用户查看详细病历功能。 | 无 |
| 钟宝亮 | 医生选择功能完善与聊天界面过度 | 无 |
| 杨童舟 | 完成患者查看，今日异常查看，今日未上报查看和信息审核四个界面之间的跳转 | 完成患者查看，今日异常查看，今日未上报查看和信息审核四个界面之间的跳转 困难：四个页面通过侧边栏按钮连接，仅仅改变中间的显示区域，不改变Header和SideBar |
| 尹梓康 | 尝试调整医生各个页面的显示逻辑 | 医生模块在初期设计时没有在侧边栏里添加功能，所以跳转键须放在具体的每一个医生功能模块里 |
| 4.25 | 李昊 | 提交并修改项目文档 | 无 |
| 王靖雄 | 优化发送消息接口 | 需要和前端沟通解决消息显示问题 |
| 丁爽 | 完成医生查看病历功能。 | 无 |
| 钟宝亮 | 聊天记录实时刷新实现（轮询） | 无 |
| 杨童舟 | 尝试用界面显示的true和false进行页面间的跳转显示 | 界面的组织结构发生变化，需要重构 |
| 尹梓康 | 优化医生各个页面的显示逻辑，以适应整体风格 | 无 |
| 4.26 | 李昊 | 修改后端接收请求格式 | 对后端接收方式不熟悉 |
| 王靖雄 | 为完成前后端正常交互修改原来写的注解 | 无 |
| 丁爽 | 在病历和药品进行修改后的添加病历功能。 | 无 |
| 钟宝亮 | 聊天记录刷新bug处理与功能优化 | 无 |
| 杨童舟 | 完善四个界面之间的跳转，解决四个界面选择后侧边栏按钮状态的问题。 | 无 |
| 尹梓康 | 调整页面初始化方法，为对接做准备 | 无 |
| 4.27 | 李昊 | 实现登录后返回个人数据的功能 | 无 |
| 王靖雄 | 用swagger测试所编写接口，检验错误 | 无 |
| 丁爽 | 在病历和药品进行修改后的添加病历功能。 | 无 |
| 钟宝亮 | 个人信息的修改功能实现 | 无 |
| 杨童舟 | 完成社区工作者查看患者详情界面，显示患者的详细信息以及患者对应的医生的信息 | 无 |
| 尹梓康 | 完善各个页面的框架和方法，以及相互跳转的逻辑 | 无 |
| 4.28 | 李昊 | 实现健康数据其他方面的功能 | 无 |
| 王靖雄 | 修改部分错误接口的代码 | 无 |
| 丁爽 | 在swagger ui上调试接口。 | 有很多奇怪的bug。 |
| 钟宝亮 | 个人信息修改后与其他页面信息同步功能完善 | 无 |
| 杨童舟 | 完成社区工作者查看信息审核详情的界面 | 无 |
| 尹梓康 | 为患者病历和健康上报添加了状态栏 | 无 |
| 4.29 | 李昊 | 无 | 无 |
| 王靖雄 | 修改部分错误接口的代码 | 无 |
| 丁爽 | 与前端进行交互，完成药品的的相关操作。 | 前端对数据库不太熟悉，有时候修改表或者操作不熟悉，最后决定实操我来帮助一部分。 |
| 钟宝亮 | 前端整体代码调整与结构优化 | 无 |
| 杨童舟 | 完成患者信息浏览界面到患者详情查看界面的跳转 | 跳转仅在界面中间完成，且需要在界面之间传递信息 |
| 尹梓康 | 为修改信息添加了清空功能 | 无 |
| 4.30 | 李昊 | 完善以前的代码结构 | 代码繁多 |
| 王靖雄 | 系统测试接口 | 发现一些bug |
| 丁爽 | 完成删除病历功能。 | 无 |
| 钟宝亮 | 病例概览功能实现（取出所有病例概览数据） | 无 |
| 杨童舟 | 将一个主界面和其分支放在同一个vue组件中，通过v-if进行显示 | 需要改变目前的组织架构，将vue组件重写 |
| 尹梓康 | 为各个表格添加了隐式的id属性，便于前后端交互 | 无 |
| 5.1 | 李昊 | 无 | 无 |
| 王靖雄 | 无 | 无 |
| 丁爽 | 与前端进行交互，完成用户和医生查看自己历史病历和自己病人的病历的功能。 | 需要让代码更适合前端同学去调用，就需要修改。 |
| 钟宝亮 | 病例详细信息查看功能实现 | 无 |
| 杨童舟 | 完成社区工作者信息审核和信息详情之间的界面跳转 | 无 |
| 尹梓康 | 学习前后端交互相关内容 | 无 |
| 5.2 | 李昊 | 病历的添加与修改 | 数据库中病历与处方的关系复杂 |
| 王靖雄 | 无 | 无 |
| 丁爽 | 完成医生开病历这一功能的前后端交互。 | 交互时参数的声明不知道为什么不对，最后改了下实现方式。 |
| 钟宝亮 | 病例删除功能实现 | 无 |
| 杨童舟 | 建立数据库，并将MySQL连接到IDEA，尝试患者信息浏览的数据显示 | 接口使用不熟悉，数组内容传递可能存在问题 |
| 尹梓康 | 前后端对接，完成呈现病患总览、呈现健康上报和医生修改自身信息的方法 | 对前后端交互较为生疏 |
| 5.3 | 李昊 | 根据修改后的数据库继续开发 | 前期代码作废，需要重新编写 |
| 王靖雄 | 协助完成前后端交互 | 前后端交互存在一些困难 |
| 丁爽 | 在电脑上运行整个web项目，调试看有没有bug。 | npm之前没下好，出了点问题。 |
| 钟宝亮 | 医生列表选择更改功能测试 | 无 |
| 杨童舟 | 完成患者详情查看中患者以及患者对应的医生的数据显示 | 患者详情查看时需要两个view\_model之间传递信息，确保点击患者A一栏的“查看详情”按钮显示的是患者A的详细信息。需要将患者的信息与医生对应，可能需要从后端另外调用接口 |
| 尹梓康 | 用formatter解决前后端数据类型不匹配的问题，并完善各个页面的细节部分 | 难以迅速处理各个琐碎的细节 |
| 5.4 | 李昊 | 辅助前端进行前后端交互 | 前后端格式不对应 |
| 王靖雄 | 协助完成前后端交互 | 无 |
| 丁爽 | 与小组讨论项目汇报的细节和形式，决定做一个ppt。 | 做ppt时对前端成员的工作不太熟悉不好写清楚。 |
| 钟宝亮 | 健康上报功能测试 | 无 |
| 杨童舟 | 完成社区工作者查看今日异常以及今日未上报的数据显示 | 无 |
| 尹梓康 | 前后端对接，完成呈现病历方法 | 前后端交互不当，忽视了传回数据未用data封装，花费了大量时间Debug |
| 5.5 | 李昊 | 辅助前端进行前后端交互 | 无 |
| 王靖雄 | 协助完成前后端交互 | 无 |
| 丁爽 | 完成小组汇报的ppt。 | 无 |
| 钟宝亮 | 病例相关功能测试 | 无 |
| 杨童舟 | 完成社区工作者查看信息审核和信息详情的数据显示，并实现审核功能 | 无 |
| 尹梓康 | 前后端对接，完成呈现处方、添加病历、添加处方、删除病历方法 | 无 |
| 5.6 | 李昊 | 辅助前端进行前后端交互，测试已完成的功能 | 无 |
| 王靖雄 | 协助完成前后端交互，写个人心得 | 无 |
| 丁爽 | 对部分功能进行测试 | 无 |
| 钟宝亮 | 实时聊天与刷新功能测试 | 无 |
| 杨童舟 | 完成社区工作者对今日未上报患者进行一键提醒功能，完善社区工作者登录后的页面跳转功能 | 无 |
| 尹梓康 | 添加与患者交流功能，补充完成删除处方、修改诊断方法，修改注册页面图片，为密码框添加显示或隐藏密码的新功能，添加了刷新按钮 | 无 |
| 5.7 | 李昊 | 对整体功能测试，进行演示 | 无 |
| 王靖雄 | 无 | 无 |
| 丁爽 | 无 | 无 |
| 钟宝亮 | 无 | 无 |
| 杨童舟 | 无 | 无 |
| 尹梓康 | 无 | 无 |

6.1.2 发布项目燃尽图：如下图所示。

图6.1.1

6.2 Beta版本测试报告

6.2.1 Bug列表：如下表所示

表6.2.1

|  |  |  |
| --- | --- | --- |
| **缺陷ID** | **主题** | **测试人** |
| 1 | 利用侧边栏按钮进行页面跳转时，用按钮点击事件连接到新路由无法保留页面中的其他信息。 解决：为每个页面添加view\_model标签，用标签控制显示的页面 | 杨童舟 |
| 2 | 患者信息显示和患者信息详情显示在同一个界面上。 解决：将两个界面都添加标签，先调用函数显浏览界面，在点击按钮后显示第二个界面 | 杨童舟 |
| 3 | 患者信息和患者对应的医生信息因为响应时间问题，无法同时显示。 解决：后端添加同时查看患者和医生的函数，调用接口。 | 杨童舟 |
| 4 | 身份信息审核通过后，信息审核列表中的待审核信息没有消失。 解决：调用reload函数，重新加载页面 | 杨童舟 |
| 5 | 个人信息修改后其他页面信息没有及时修改。 解决：采用store全局变量解决 | 钟宝亮 |
| 6 | 聊天信息输入后无法获取。 解决：设置输入框ref变量 | 钟宝亮 |
| 7 | 聊天记录无法实时刷新。 解决：采用轮询隔一定时间刷新信息显示 | 钟宝亮 |
| 8 | 图片上传无法解析为mutipart请求 解决：数据在前端被默认为json格式 | 钟宝亮 |
| 9 | 查看健康上报信息表格不显示信息，呈现信息方法在为呈现信息所需id赋值之前被调用 解决：调节调用该方法的时机 | 尹梓康 |
| 10 | 呈现病历不显示信息，传回数据未用data封装，前端调用时仍使用不存在的data数组 解决：去掉data，直接用传回数据赋值 | 尹梓康 |
| 11 | 消息页面不显示，消息页面被放在了其他页面内 解决：将消息相关代码独立出来 | 尹梓康 |
| 12 | 用swagger ui调试出现问题，接口无法正常使用。 解决：数据库的配置因为用git拉取的缘故，变成其他成员的，每次更新后检查修改。 | 丁爽 |
| 13 | 药品无法正常添加到病历，因为添加病历时，病历还未生成，药品的病历号为不存在。 解决：将药品的病历号属性删除，新建一个表，病历-药品匹配表来解决。 | 丁爽 |
| 14 | 病历无法正常添加。 解决：病历的service层也用到了drugmapper，没有添加@Autowired，添加上即可。 | 丁爽 |
| 15 | 用requestbody不知道为何一直报错，最后决定还是一个一个添加参数。 | 丁爽 |
| 16 | 后端发送数据时并非以RespBean形式发送导致使用困难 解决：以RespBean形式传数据 | 王靖雄 |
| 17 | 前后端交互失效 解决：修改注解 | 王靖雄 |

6.2.2 场景测试，包括以下内容。

* 典型用户，如下表所示。

表6.2.2

|  |  |
| --- | --- |
| 姓名 | 张三 |
| 性别 | 男 |
| 年龄 | 25 |
| 职业 | 程序员 |
| 使用目的 | 监测身体健康状况 |
|  |  |
| 姓名 | 李四 |
| 性别 | 女 |
| 年龄 | 31 |
| 职业 | 医生 |
| 使用目的 | 处理数据异常的患者 |
|  |  |
| 姓名 | 王五 |
| 性别 | 男 |
| 年龄 | 27 |
| 职业 | 社区管理员 |
| 使用目的 | 管理该社区的医生和患者 |

* 典型场景

张三是一名程序员，最近上面指派了重要的任务，要求在很短的时间内完成，他和他的小组日夜奋战，任务没有完成，但是张三觉得自己的身体出现了一些问题，在看过医生之后知道了熬夜的风险，但是任务依然要按时完成。于是张三打开健康监测系统网站，通过每天上报数据来监测自己的健康状况。他首先进行了注册，并且登录进入了主页。在登陆之后，他修改了自己的个人信息，并且上传了他的个人头像。然后他进行了监测数据的上报，上报之后发现数据填写有问题，于是又对上报的数据进行了修改。修改成功后，系统提示自己的情况有异常，于是他希望选择一名医生为他解决。最终他在选择列表里选择了人数还未满的李四医生，选择后它通过发消息向医生告知了自己的异常，没过一会张三看到了李四医生为他开的病历，它通过病历吃药调养，并且每日坚持上报数据，终于回归了正常值。

李四是一名医生，负责为选择她的患者提供治疗方案，她首先注册了医生的账户，上传了资格证明，在管理员对她的资格证明验证后，她成功登录进入了主页，不久看到张三选择自己作为他的医生，并且看到他上报的数据有异常，在观察了他上报的数据并且与张三沟通后，为张三开了一份病历。

王五是一名社区管理员，负责管理该社区的患者和医生。一日，在登录账户进入主页之后，他先验证了李四医生的注册请求，之后发现了用户张三今日还没有上报数据，于是提醒张三上报数据，在张三上报数据后发现张三上报的数据异常，因此向张三选择的医生李四提醒她关注张三的异常。

6.2.3 测试矩阵，如下表所示

表6.2.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **测试功能** | **测试项** | **输入/操作** | **检查点** | **测试功能** | **完成情况** |
| 登录 | 静态功能 |  | 登录界面布局 |  | 完成 |
|  | 注册按钮 | 点击进入注册页面 | 完成 |
|  | 登录按钮 | 点击登录 | 完成 |
| 动态功能 | 正确的账号和密码 | 确认功能，结果显示 | 进入相应类型用户的主页 | 完成 |
| 错误的账号或密码 | 确认功能，结果显示 | 提示错误信息 | 完成 |
| 注册 | 静态功能 |  | 注册界面布局 |  | 完成 |
|  | 性别单选框 | 点击切换性别 | 完成 |
|  | 确认注册按钮 | 点击进入登录页面 | 完成 |
| 动态功能 | 密码和确认密码不一致 | 确认功能，结果显示 | 提示错误信息 | 完成 |
| 身份证号重复 | 确认功能，结果显示 | 提示错误信息 | 完成 |
| 输入全部正确 | 确认功能，结果显示 | 注册成功，返回登录界面 | 完成 |
| 对非空数据判断 | 确认功能，结果显示 | 提示错误信息 | 未完成 |
| 医生和社区上传资格证明 | 确认功能，结果显示 | 上传照片 | 未完成 |
| 主页 | 静态功能 |  | 导航栏按钮分布 | 可点击 | 完成 |
|  | 主界面信息布局 |  | 完成 |
|  | 右上角账号名 | 移动鼠标出现选项 | 完成 |
| 动态功能 | 个人信息更改 | 确认功能，结果显示 | 成功更改信息 | 完成 |
| 退出登录 | 确认功能，结果显示 | 返回登录界面 | 完成 |
| 患者主页 | 静态功能 |  | 患者主页界面布局 |  | 完成 |
|  | 导航栏按钮布局 | 切换不同功能 | 完成 |
|  | 医生消息按钮 | 点击显示与医生的聊天记录 | 完成 |
|  | 系统消息按钮 | 点击显示系统消息记录 | 完成 |
|  | 健康上报按钮 | 点击上报今日的健康数据 | 完成 |
|  | 清空按钮 | 清空填写的健康数据 | 完成 |
|  | 所有医生按钮 | 点击查看所有医生信息 | 完成 |
|  | 我的医生按钮 | 点击查看所选医生信息 | 完成 |
| 动态功能 | 查看消息记录 | 确认功能，结果显示 | 相应消息记录的显示 | 完成 |
| 发送消息 | 确认功能，结果显示 | 消息的发送和正确显示 | 完成 |
| 定时刷新消息记录 | 确认功能，结果显示 | 定时调用消息显示函数 | 完成 |
| 健康上报错误形式输入 | 确认功能，结果显示 | 显示错误信息 | 完成 |
| 选择医生时，医生选择人数+1 | 确认功能，结果显示 | 到达上限时显示已满 | 完成 |
| 删除病历 | 确认功能，结果显示 | 删除所有病历相关数据 | 完成 |
| 健康上报数据出现异常 | 确认功能，结果显示 | 显示异常通知 | 完成 |
| 医生主页 | 静态功能 |  | 医生主页界面布局 |  | 完成 |
|  | 发送消息按钮 | 点击进入和患者消息界面 | 完成 |
|  | 查看详情界面 | 点击查看患者详细信息 | 完成 |
|  | 开病历按钮 | 点击为患者添加病历 | 完成 |
|  | 开处方按钮 | 点击为患者添加处方 | 完成 |
|  | 删除病历按钮 | 点击为患者删除病历 | 完成 |
|  | 删除处方按钮 | 点击为患者删除处方 | 完成 |
| 动态功能 | 进入主页后有患者选择自己 | 确认功能，结果显示 | 刷新显示该患者信息 | 完成 |
| 患者有异常数据 | 确认功能，结果显示 | 异常记录标记为红色 | 完成 |
| 管理员主页 | 静态功能 |  | 管理员主页界面布局 |  | 完成 |
|  | 左侧导航栏布局 | 切换不同功能 | 完成 |
|  | 患者按钮 | 点击显示所有的患者和对应医生信息 | 完成 |
|  | 今日异常按钮 | 点击显示今日异常的患者信息 | 完成 |
|  | 今日未上报按钮 | 点击显示今日未上报患者信息 | 完成 |
|  | 信息审核按钮 | 点击显示注册且未审核的医生信息 | 完成 |
|  | 一键提醒按钮 | 点击发送系统消息 | 完成 |
| 动态功能 | 信息审核点击同意 | 确认功能，结果显示 | 注册的医生可以登录 | 完成 |

6.2.4 出口条件

界面完整，功能都可以运行，多次测试没有问题。

6.3 Beta版本发布说明

6.3.1 版本功能

* 登录功能，可以完成已注册用户登录的账号检查和提示功能。



图6.3.1

* 注册功能：实现新用户账户的注册，使用身份证号进行唯一标识。

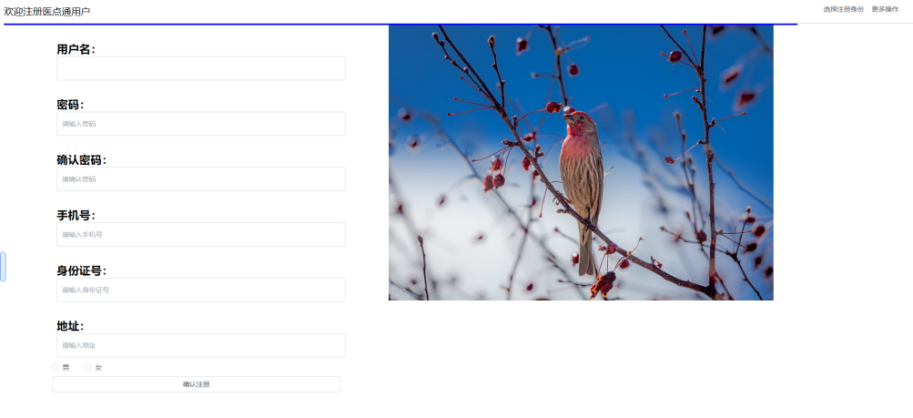


图6.3.2

* 修改个人信息：对个人信息进行修改

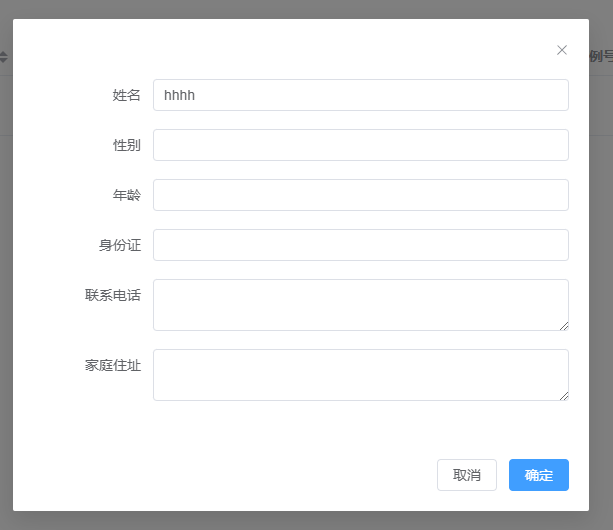


图6.3.3

* 消息交流功能：医生和患者之间可以通过互发消息进行交流

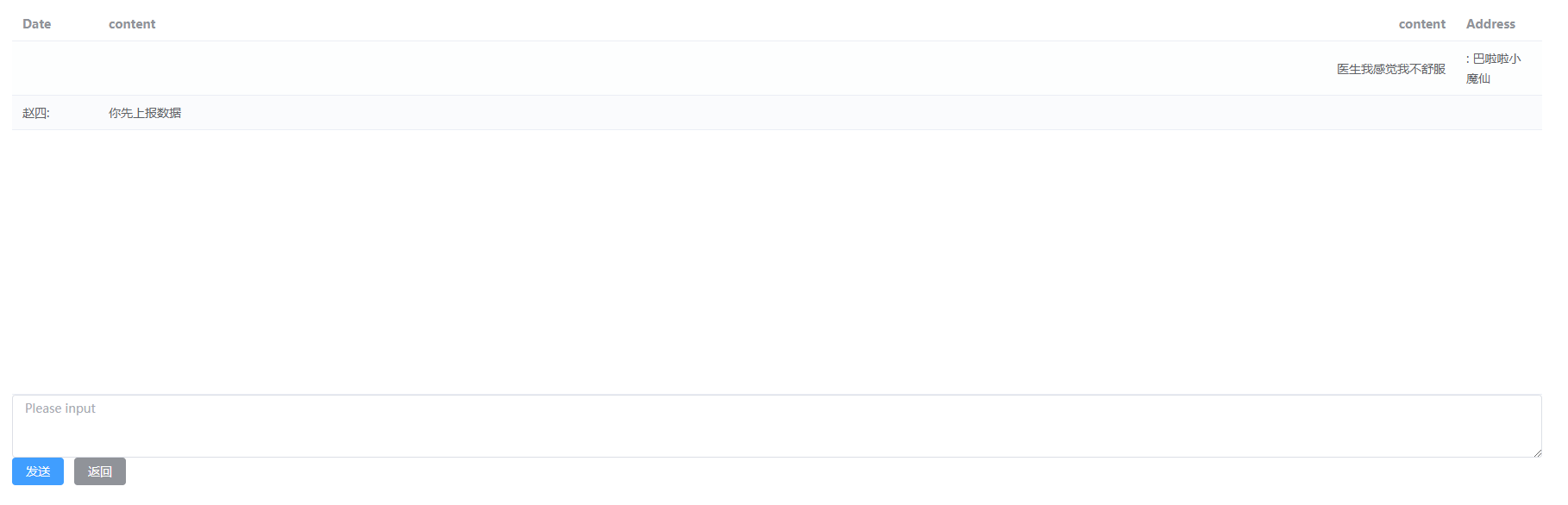


图6.3.4

* 健康上报数据功能：每天上报数据，如未上报会发送提示。

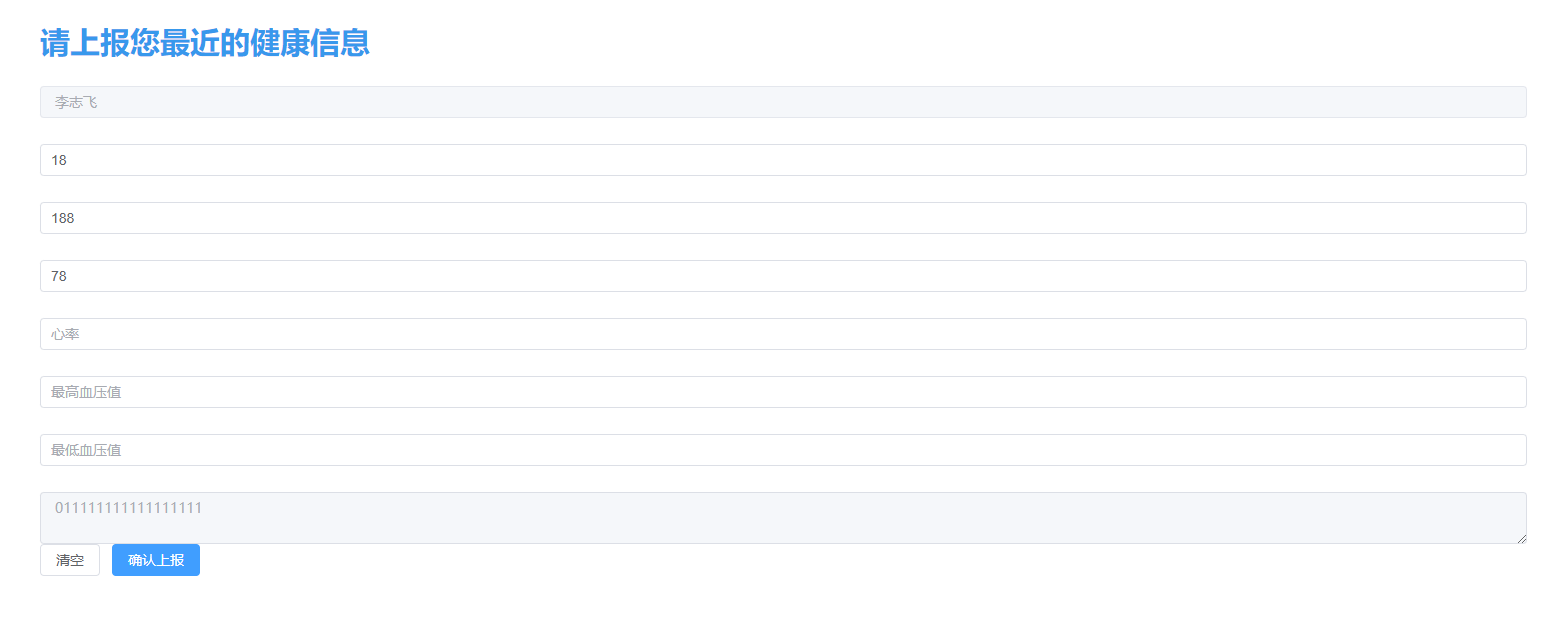


图6.3.5

* 医生选择：患者在某一时刻只能最多选择一位医生。



图6.3.6



图6.3.7

* 病历管理功能：医生可以查看选择自己的患者的个人信息，病历记录情况和个人健康数据状况，并对病历进行管理（添加，删除，修改，添加处方）。患者也可以查看到自己的病历，通过病历内容进行相应的调整和治疗。

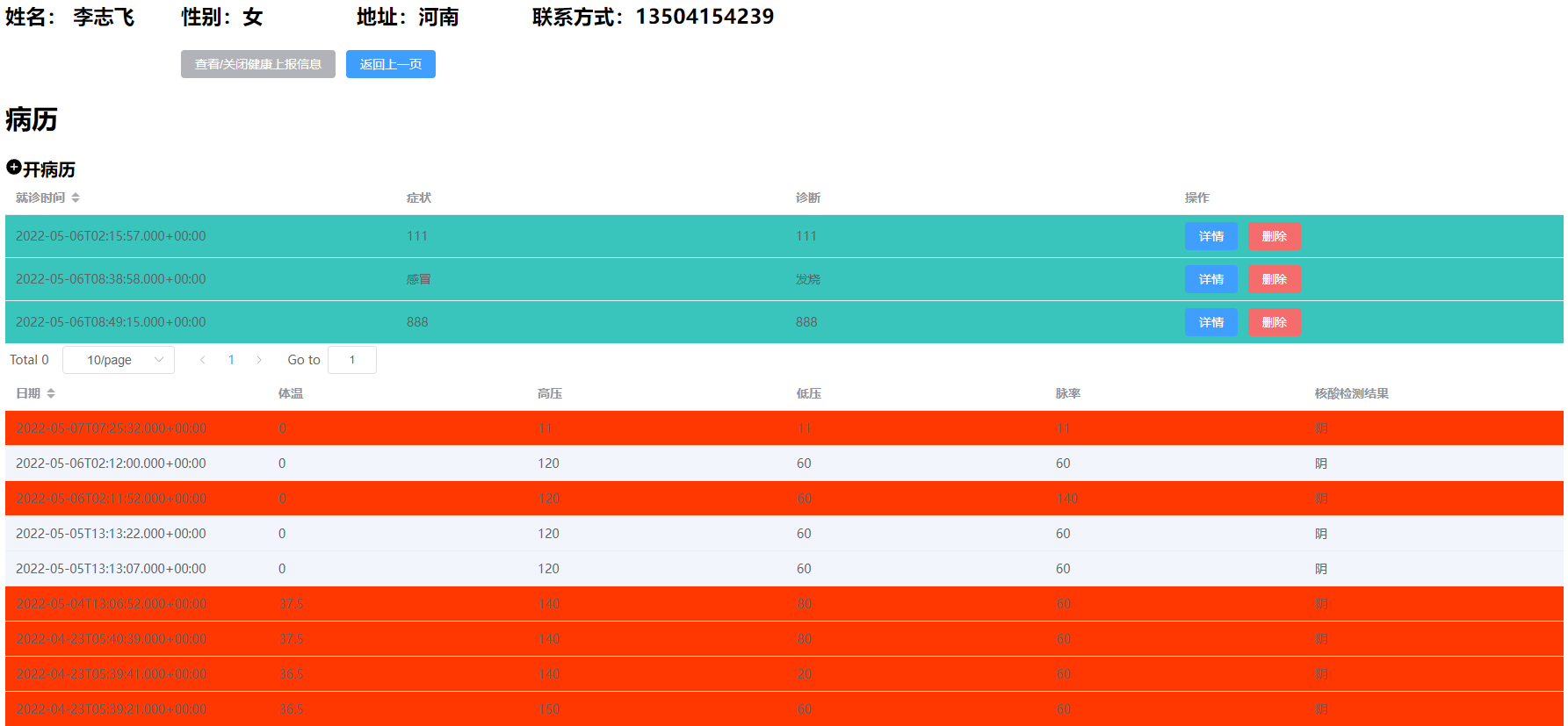


图6.3.8

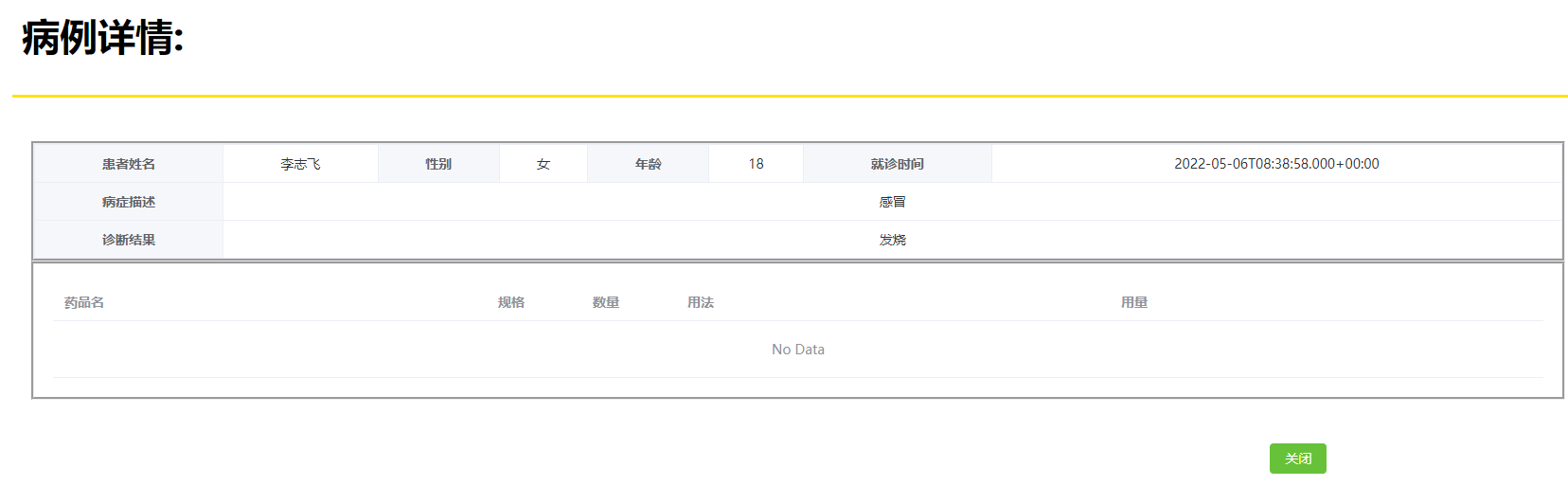


图6.3.9

* 信息审核功能：刚刚注册的医生账号需要进行管理员的信息审核，管理员会验证信息的真实性后才可以通过。未通过审核的医生账号不可以成功登录。



图6.3.10

* 患者管理功能：管理员可以对所有的患者进行管理，并且可以查看到不同信息的患者信息，如上报数据异常的患者和为上报数据的患者，对这些患者会有相应的提示。

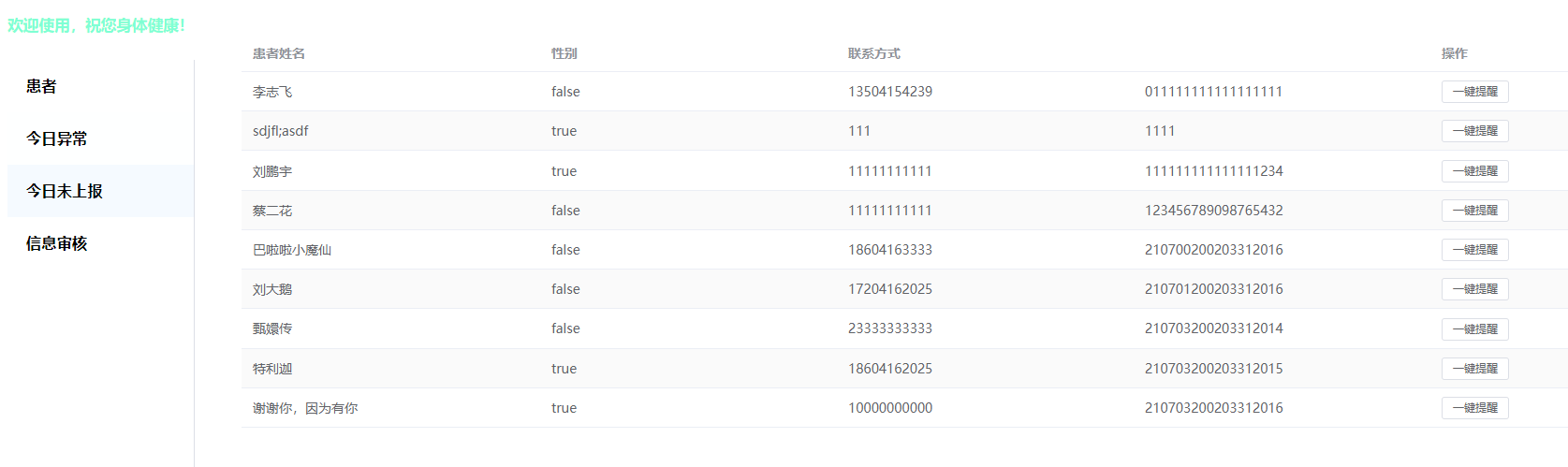


图6.3.11

6.3.2 对运行环境的要求

健康监测系统以网页形式存在，下表针对测试过的浏览器进行说明。

表6.3.1

|  |  |
| --- | --- |
| 火狐FireFox | 使用正常 |
| 谷歌Chrome | 使用正常 |
| Edge | 使用正常 |
| IE浏览器 | 使用正常 |

6.3.3 安装方法

以网页形式存在，无须安装，只需访问网站的网址就可以访问该健康监测网站。

6.3.4 描述系统一直的问题和限制

Beta版本基本完成了所需的功能和相应的测试环节，实现了患者、医生、管理员三者有机的交流，但是依然有很多功能值得去优化和改进。数据库中存储的数据有所增加，但是相较于正式的发布仍有很大差距。未来如有机会，将会继续完善和改进相关功能，为用户提供更出色的体验。

6.3.5 说明软件的发布方式以及发布地址

Beta版本还未发布到远程服务器上，还只能供开发人员本地调试使用。

6.4 Beta阶段小结

6.4.1 团队的源码仓库地址

<http://125.220.157.107/WorldOneTwo/healthmonitoringsystem.git>

6.4.2 Beta过程回顾

* 团队项目的目标，预期的典型客户，预期的功能描述，预期的用户数量？

我们团队项目的目标是为身体情况不健康、亚健康的人进行身体情况监测的平台，让他们可以通过每日的数据上报了解自己身体的现状和变化情况，对潜在的疾病防患于未然。

预期的典型用户是对所有自身健康情况有担忧的患者，负责治疗的医生和负责人员管理的社区管理员。

预期功能有：

（1）用户可完成注册登录，个人信息编辑

1. 用户可以进行健康数据的上报和修改。可以查看历史记录
2. 用户可以选择医生，查看医生为他写的病历
3. 用户可以和其他人交流，可以获得系统消息的通知。
4. 医生可以查看选择自己的患者信息，为其开处方
5. 社区管理员可以对别人进行资格审查，对特定的用户进行处理

由于该系统主打健康监测功能，对广大群众都有需求，但缺乏宣传推广渠道，用户在数量可能会有所限制。

* 团队的产品如何满足了用户的需求？

团队的产品满足了用户对于自身健康状况的关心，并且随时随地可以测量数据，比去医院更为方便快捷。

* 用户在Beta阶段已完成了那些目标？

已经完成和测试了初步所需的功能，按阶段完成所需的具体文档。

* 团队成员是如何分工协作的？有什么经验教训？

团队对每一阶段的工作任务都按照美味成员的工作能力和强项进行了合理分配，并注重交流合作，每天进行一次线上或线下会议，分享每个人完成的工作，遇到的问题和独到的看法。但是在工作中仍然存在沟通失误的情况，会存在前端界面设计问题和前后但交互冲突问题。在需求分析上也存在问题，没有在前期对需求进行详细的确认，导致成员对于需求的理解不一致，后期在协调需求统一方面浪费了大量时间。于此同时，组员对于开发软件和技术仍然不够熟悉，因此出现的无谓错误也延误了开发进度，导致后期时间紧任务中，给每一位团队成员都造成了不少压力。

* 团队是如何进行项目管理的？

合理分配项目任务，在GitLab上建立团队项目，及时上传改进部分并整合。

* 团队如何平和 时间/质量/资源 争取如期完成任务的？

在项目开发前期，我们对团队成员的技术问题进行了讨论和总结，对某些成员困难的方面给予支持，并确定各个成员擅长的方面，以此对工作内容进行合理的分配。每天进行会议，了解成员的工作情况，对不合理的任务进度做出对应调整。在产品质量上，我们计划首先完成基础功能的实现，在此基础上增加一些功能特色，并在界面上进行美化，达到视觉的赏心悦目。

6.4.3 团队项目的实际进展（见燃尽图），发布的功能（见发布文档）。说明在项目管理中，软禁图是如何真实反应项目的状态的？或者燃尽图美化了状态？

由于已经经历了Alpha阶段的开发，组员们对于开发的技术和经验都逐步加深，使得Beta阶段前期的开发非常迅速，进度略快于预期进度。之后由于前后端交互方面的种种问题和测试方面的困难使得中后期得进度有所放缓，但是依然按时完成了所需得开发。

6.4.4 团队成员在Alpha阶段的角色和具体贡献

如下表所示。

表6.4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 名字 | 角色 | 团队贡献分 | 可验证的贡献 |
| 李昊 | PM&Dev&Test | 20 | 后台功能实现 |
| 王靖雄 | Dev&Test | 20 | 后台功能代码 |
| 丁爽 | Dev&Test | 20 | 后台功能实现 |
| 钟宝亮 | Dev&Test | 20 | 患者界面功能开发 |
| 杨童舟 | Dev&Test | 20 | 管理员界面功能开发 |
| 尹梓康 | Dev&Test | 20 | 医生界面功能开发 |

7.个人贡献、学习收获和教训总结

7.1 个人贡献

如下表所示。

表7.1

|  |  |  |  |
| --- | --- | --- | --- |
| 姓名 | 角色 | 贡献分 | 可验证贡献 |
| 李昊 | PM&Dev&Test | 20 | 项目基本框架的设计与搭建 健康上报、登录注册、医生患者管理员等模块中的部分功能实现（代码总量：800行） |
| 王靖雄 | Dev&Test | 20 | 完成消息收发后端数据库的更新工作,完成显示当前用户所有病例的后端部分,完成显示系统内所有医生的后端部分,完成用户选择医生的后端部分,完成医生获取所有用户基本信息的后端部分,实现部分登录注册的后端功能（代码总量：463行） |
| 丁爽 | Dev&Test | 20 | 后端病历模块和药品模块的功能实现。 （代码总量：464行） |
| 尹梓康 | Dev&Test | 20 | 医生页及其功能实现，注册页及其功能实现（代码总量：1400行） |
| 钟宝亮 | Dev&Test | 20 | axios前后端交互接口功能设计与搭建 注册框架、登录、实时聊天、个人信息修改等可复用功能（有效代码约450行） 普通用户所有功能前端页面搭建（vue+js有效代码约650行） |
| 杨童舟 | Dev&Test | 20 | 所有页面的布局跳转草图设计 社区工作者前端相关界面功能的实现及数据交互 代码总量：400行 |

7.2 学习收获与教训总结

7.2.1 组长 李昊

**（1）学习收获**

我在这次的项目中主要负责整体项目的构建和后端功能的开发，在经过了几周的开发之后，我对于我所掌握的部分有了全新的理解。

* 使用Maven工具进行开发，可以用来方便地导入Java类依赖的jar，并且协调jar之间的依赖和冲突，在构建项目，管理依赖和单元测试等方面有着显著的作用。Maven规定了统一的目录结构，所有的java代码都放在src/main/java中，所有的测试代码都放在src/test/java中，而其他格式的文件则统一放在相应的resources中，使得管理代码和文件十分方便，也便于程序跨平台的使用。Maven统一维护jar包，maven使用仓库管理依赖的jar包，当项目需要使用相应的依赖时，只需要从中央仓库中寻找相应的版本并下载到本地仓库中，并在pom.xml文件中给出需要使用的jar包名称和版本号。简化了依赖的导入和冲突。
* 使用MyBatis框架进行和数据库的交互，Mybatis是一个优秀的基于java的持久层框架，它内部封装了jdbc，能够让开发者只需要关注sql语句本身，而不需要花费精力去处理加载驱动/创建连接/创建statement等繁杂的过程。MyBatis可以通过xml文件的方式将java代码与SQL语句分离，定义相应的Mapper文件，在拥有唯一标识的标签中书写SQL语句，并且在相同的目录中定义与Mapper文件同名的接口，接口名与xml文件一一对应，接口中的每一个方法与每一个标签一一对应，这样便可以通过调用Mapper接口中的方法调用SQL语句，完成对数据库的操作。
* 使用Springboot框架简化开发，Springboot建立在Spring框架的优秀基础上，全方面的简化了Java开发难度，通过依赖注入等方式降低了耦合度，并且摒弃了Spring配置困难的缺点，默认配置了很多框架的使用方式，降低了依赖冲突等问题。将类的对象作为一个bean对象存储在Spring容器中，在其它类中使用注解的方式进行bean对象注入，相当于完成对象的new操作，使得两个类在代码中的联系降低，利于维护代码。
* 使用MVC三层架构，将应用程序划分为模式、试图和控制器三个部分。其中模型部分又包含了业务处理层和数据持久层（数据业务）。视图部分负责数据的输出形式，即页面部分。控制层部分负责协调模型和试图，根据用户请求的来选择调用哪个模型来处理业务，以及最终由哪个视图为用户做出响应。MVC模式的这三个部分的职责十分明确，而且相互分离，因此每个部分都可以独立地改变而不影响其他部分，从而大大提高了应用的灵活性和重用性，使得运行和维护程序更加方便。
* 使用Swagger工具进行前后端交互的对接，由此也了解了Web项目的请求和响应，以及传递数据的格式。为此也浅显地了解了一些前端的axios技术，为前后端项目的交接打下基础。

1. **教训总结**

* 选择参照物学习：在刚刚准备开发时，Web项目对于我们来说是一个还未接触的领域，因此大量的学习肯定必不可少，但是后端需要了解的知识点多且繁杂，只有这些时间我很难将他们融会贯通。所以我尝试改变策略，不再只是跟着网上的教程学习，而是结合着老师给我们的项目示例。没学到一个知识，我就会对照着给定项目中哪里运用到了这样的知识，在整个项目中到底起着什么作用，有多么重要；给定的项目和我们要开发的项目有什么相同点，有什么不同点……在这样的方式下，我学习的速度比之前快了很多，并且对要开发的项目构造，在心中也渐渐有了雏形。
* 边开发边学习：即使如此，学习的知识仍然有限，还不足以熟练地编写代码。因此在开发过程中也出现了很多问题和困难，但是在开发中解决问题也是开发中重要的一环，在开发中寻找解决方案，也让我对知识的理解进一步加深。
* 沟通是团队的核心：作为一个小组的组长，在这次开发中我深刻明白了组长所要担当的责任，那就是协调与沟通。团队开发肯定是要比个人开发快的，但是如何发挥出每个人的优势，让其达到大于1的效果，是一个核心的问题。组员之间积极的沟通一个良方，很多一个人解决起来非常困难的问题，在经过沟通之后往往迎刃而解。而作为组长就应该积极调动大家的积极性，带头和大家沟通，项目的开发才能更加顺利的进行。

7.2.2 组员 钟宝亮

**（1）知识与能力层面**

第一次尝试搭建前端框架，同时负责了前后端交互的核心逻辑功能书写具体而言：

·vue框架基本知识，同时也是第一次使用框架，对项目常用的框架有了基本的感知

·axios 前后端交互工具使用，也是第一次实现以web应用为载体的前后端交互功能，对于请求的封装有了一定的了解，同时也了解了前后端交互的基本逻辑，为以后开发类似的web应用打下基础

·elementplus 封装组件使用，传统的前端书写需要程序员编写大量的样式代码，流程相对繁琐，采用elementplus之类的封装好的前端组件（诸如：下拉框、对话框）可以获得更好的组件视觉效果，也节省了大量工程时间。

·以聊天记录功能实现为代表的前端数据请求，同时也通过后端同学了解了后端大致请求处理机制，厘清了前后端交互请求的发送与处理大体流程，并通过轮询实现信息实时刷新、通过store全局变量存储实现全局信息统一

·web开发过程中的debug方法与错误信息分析的能力

**（2）工程构架整体思维层面**

·草图尤其重要：前端的设计需要一定的基本流程，切忌开始就进行网页界面代码书写，前端的草图构架尤其重要，通过草图能大致明确前端需要设计怎样的界面，后端需要怎样的建表方式与信息存储

·注重模块复用：前端的许多功能模块是可以进行复用的，这在提高工程效率的同时也实现了一些重要复用功能模块的风格统一

·职责分工明晰：不是每一个人都着重于去写代码，有的可能着重于去研究核心功能实现逻辑，有的则需要重复劳动去书写，有的则负责前端绘图

·前端非完全独立：功能模块要及时和后端人员沟通，尤其在页面做好后刚开始做功能时需要通过后端负责人了解大致后端表单与接口设计进度（最好线下），这样效率会高不少，线上会议再多没人做也是白搭

·学会放下

**（3）个人感受**

“慢慢走会比较快”，这句话说的一点也不假，刚接触前端及web类型应用前后端交互的时候需要很多前期知识的铺垫，急于速成开发无异于挖肉补疮，这样学不到东西而且报错一堆，但是通过视频对未知（框架）有大致认知是可以的，就是别急着写，这主要也降低了（我这种低水平程序员）对未知的恐惧，敢于更大胆去接触框架。在项目推进过程中，会存在有的同学功能实现跟不上的情况，这时候着重于功能的人在前端单独把功能逻辑讲一下、原理解释清楚是至关重要的，“集体的”效率可以提高不少。

最后，这是一次难忘的前端开发经历，感谢每一位小组成员的努力，同时特别感谢挑大梁的组长、耐心解答的助教及老师，打板愉快！

7.2.3 组员 尹梓康

**学习收获**

在这次难忘的开发经历中，我学习了使用vue框架并结合html、js、css来进行前端页面的开发，也从中学习到了前后端交互的相关内容。整个开发的过程可以说是摸着石头过河，因为很多东西实际上都是边学边做，等到实现中才发现原先制定的计划的不合理之处，从而不得不对计划进行修改。但在这个较为艰难的过程中，我也收获了不少东西。在技术层面上，这些优秀的开发工具和模式显然能极大地提高开发的效率和速度，学习这些东西是对自己的充实和提高，对未来的学习和工作也有很大的帮助。在团队协作层面上，多次和谐有效的团队协作让我明白了众人拾柴火焰高的道理，团队协作既能相互协调、提高整体的工作效率，又能让每一个人感受到团队的力量与温情，从而继续以高涨的热情工作。除此之外，我还明白了逻辑的重要性。好的逻辑能极大程度地减少重复性的工作，而逻辑较差的代码则需要添加不少额外的部分来进行修补——这一点也让我更加确信了设计模式的巨大作用。只要不停下脚步，道路就会不断延伸，相信我们今后前进时，步子会迈得更加从容。

**教训总结**

（1）由于初期设计时医生页面的侧边栏没有具体的内容，导致在实现功能 时医生页面的切换相对而言很繁琐。这启示我一定要重视初期的设计，在最开始 设计的框架中就应该考虑具体的功能和实现，否则就是在为自己挖坑。

（2）实现呈现病历方法时，因为完全忽视了传回数据未用data封装而继续使用不存在的data数组进行赋值，所以花费了大量时间Debug，这让我意识到一定要善用swagger的每一个功能，不能浅尝辄止，同时也应该更加细心一点，绝不可过多地想当然。

（3）在修改病历和处方时为了即时显示到前端而自行添加了只用于前端更新的方法，但实际上同步到数据库后在调用一次呈现数据的代码即可，这些只用于前端更新的方法就成了没有任何作用的垃圾代码了，所以说写代码时也应该适度考虑一下某些功能到底有没有必要添加，有时一味地写代码其实只是在浪费时间。

7.2.4 组员 杨童舟

（1）在做前端页面时，一开始一心想用图形化进行设计，因为觉得全部用代码写太麻烦而且也不方便布局。于是用了figma还有build.io之类的很多工具，耗费了不少时间，但最终做出来的页面并不是很好看而且并没有方便多少。新的工具要去探索和尝试，但是基本的方法还是需要掌握，当开发时间较紧的时候要减少在无谓的事情上浪费时间。

（2）这次项目前端的开发让我对vue框架有了很多的了解，在此之前对网页的前端一直只停留在html, css, javascript，对整体的框架一直缺乏认识，所以做的前端其实没有什么与后端交互的功能，这次开发让我知道了如何在vue框架下对网页前端进行开发。

（3）通过此次的项目开发，我深切感受到了源码阅读的重要性。进行源码阅读的同时在网络上查找资料，这样可以让我快速地把握代码的整体框架和各个部分的作用，方便了自己写代码时代码块的安排。

（4）在这次的项目开发中，自己的时间安排其实并不是很合理，整体呈现出前松后紧的状态，对自己的能力估计不是很准确，导致有的计划一天完成的任务，需要耗费两天甚至三天的时间。

（5）在项目开发过程中，我们小组开会的频率几乎都维持在一到两天一次，每次开会的时间其实都不长，但是我们每次能够及时交流进度，并且对接下来的开发计划进行规划，督促完成各人的任务。

（6）在项目开发的过程中，我们小组进行了多次集体开发，还会分前后端分别进行集体开发。这种开发方式从很大程度上提高了我的工作效率。大家在敲代码的过程中遇到问题相互讨论可以很快地解决问题，同时这种随时交流的工作方式也可以让大家及时交换意见。

（7）在此次的项目开发中，最大的收获应该时学会了有问题不要自己闷着头解决，多和小组中的成员沟通交流，常常会得到更好的解决方法。就比如在进行页面上部分区域的跳转时，自己的想法实验了很多种都无法实现，网上找到的资料又多与自己的实际情况不同，最后问了小组的成员得到了很好的解决。

7.2.5 组员 丁爽

（1）在刚开始做后端的时候没有完全了解完开发的环境要求，只想急于求成做出一点东西，最后发现环境不适合，编译器不适配等问题最后还是要从头再来。万事开头难，不能因为难就随便开始，不然你后面会发现开的早其实没有用，环境和工具不行就是不行，你还要从头再来。

（2）面对问题我们要做到的实解决和绕开，最初的功能设计中，涉及到较复杂的功能实现时。一开始的想法是通过修改功能简化，比如药品和病历的匹配，我没有多建一个匹配表，因为感觉麻烦，最后功能实现的时候发现不行，还是加上了。很多问题都是这样，好难啊做不到，删了吧，但其实在和小组成员协商过程中，我们最后还是保留了原来的想法，并克服了技术上的困难。

（3）这次的项目后端开发让我对java框架有了更多的了解，也终于明白为什么有的老师说其实项目开发很简单，因为他就像填空题，他给了你框架和实现的方法，你去实现和写功能就可以，而且bean,mapper,service,controller的层次真的很清晰，什么功能该放哪里都很清楚，你自己后续修改调试包括给别人看都很方便。

（4）通过此次的项目开发，我深切感受到了多学习一些优秀项目的益处，我最感谢的就是老师布置的那个源码阅读的作业，对我的开发真的很有用，很清晰的告诉我项目到底该怎么做，他也给我很多开发时的启发，也让我在功能开发不下去的时候解决我的疑惑。真的很感谢老师源码阅读的这个作业的布置，并且要求我们进行分析，真的很有用。

（5）软件开发的过程基本需要自主学习。而且很多工具的介绍都是英文版的，但是看多了我也不再排斥官方文档的阅读，很多方法很多库其实编辑器已经给你做好了，但是需要你自己发现，能够从英文的帮助文档中找到自己需要的信息，是一个很大的突破。

（6）这次实训的我深刻的感受就是，优秀的人会促使你的进步，和我一起开发后端的成员的代码水平是比我高的，他做的好，我看见了就会学习到，而且他的完成效率也促使着我不拖拉按时完成任务，比如当时他画完用例我画类图，虽然当时要求是一个模块就好，但是他把整个系统功能都画了出来，我受其影响，研究了一下午，把整个系统的类图全画了出来，当时真的很有成就感，其实你画出来了，你要做什么也就很清晰了，所以说要和优秀的人一起学习。

（7）在这次的项目开发中，自己的时间安排其实并不是很合理，整体呈现出前松后紧的状态，对自己的能力估计不是很准确，导致有的计划一天完成的任务，需要耗费两天甚至三天的时间。

（8）这次开发我深刻感受到了交流的重要性，其实我们的开发开始是比较靠后的，有的组在4月之前就开始做，但是我们每天都开一次小会，两三天线下开会一起开发，真的有感受到效率，项目每天肉眼可见的形成，开发的效率非常高，虽然做的多，但又没觉得特别辛苦，因为有问题可以一起解决，需要交流的地方也可以及时沟通，所以也没有很辛苦。

（9）这次开发我感受到了前后端开发的优越性，之前实训我们是按界面分的各做各的，我现在懂了为什么要分前后端，你可以更好的掌握一个方面，并去进行开发，分工也会更加清晰，有问题之后你可以很清楚的知道问题在哪，而不是晕头转向。

（10）此次软件开发是我第一次接除git，它真的非常优越，以前我们一起小组合作，最头疼的就是每个人电脑上东西都不一样，合的超级麻烦，你用git就会很方便，每一次更新都会显示，别人代码写好了你拉取一下就好了，很适合多人开发，也挺适合自己写东西进行更新，我觉得我以后离不开git了。

7.2.6 组员 王靖雄

**学习收获**

经过本次项目实训，我接触了过去可能不大敢上手的SpringBoot，在实际后端开发过程中，我深刻体会到SpringBoot简化编码、简化配置、简化部署、简化监控的强大优势。从最初的选题、需求分析到实际开发，不断测试迭代，到此刻总结本次开发，我投入到每一个环节中，感受项目开发全过程，体会软件工程思想。通过本次小组开发，我不断学习后端开发相关的知识，并在不断试错纠错的过程中取得进步，在小组每晚开会总结中不断成长。感恩组长组员、感恩此次项目开发经历，相信我们在一起，能做到最好！

**教训总结**

SpringBoot中有一个非常重要的，也是过去很少接触的内容叫注解，本次开发中由于不了解，在网上找的教程五花八门，在注解上我走了弯路，甚至在项目结束后，我还是不是很理解有的地方注解为什么要这么写，只是依葫芦画瓢去完成。

对于任何新知识，一定要深入理解之后再上手，不要失去全局观。在此次开发时，SpringBoot对我来说是完全的新知识，我一开始上手时就觉得根据老师给的博客源码照着慢慢写就能写明白，但是这样效率很低。今后接触新的开发技术一定要先把它大致的思路弄懂，再去慢慢学习，不能急于求成。

代码调试、模块测试工作很复杂，虽然不追求一步到位，但尽量还是能在一开始写的时候就保持清晰的思路能把它写得像模像样。此外，在DeBug的时候一定要有耐心，不能只依靠最显眼的报错来调试，要慢慢去看每一个后台输出，这样反而能更快的找到错误原因。要学会挑选论坛博客，不能网上一搜直接就开始改，仔细看过思考过以后再来修改自己的能减少许多麻烦。