Health设计规格说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc301301027)

[1.1 目标 3](#_Toc301301028)

[1.2 文档范围 3](#_Toc301301029)

[1.3 术语和缩略语 3](#_Toc301301030)

[1.4 参考资料 3](#_Toc301301031)

[1.5 软件目标和约束 3](#_Toc301301032)

[2 软件设计 4](#_Toc301301033)

[2.1 软件架构概述 4](#_Toc301301034)

[2.2 对象模型 4](#_Toc301301035)

[2.3 接口 4](#_Toc301301036)

[2.4 特性实现 4](#_Toc301301037)

[2.5 错误代码 6](#_Toc301301038)

[2.6 错误日志 6](#_Toc301301039)

[2.7 部署视图 6](#_Toc301301040)

[3 数据库设计 6](#_Toc301301041)

[3.1 逻辑模型 6](#_Toc301301042)

[3.2 物理模型 6](#_Toc301301043)

[4 质量及其他方面 7](#_Toc301301044)

[4.1 可维护性 7](#_Toc301301045)

[4.2 安全性 7](#_Toc301301046)

[4.3 可扩展性 7](#_Toc301301047)

[4.4 可靠性 7](#_Toc301301048)

[4.5 可用性 7](#_Toc301301049)

[4.6 性能设计 8](#_Toc301301050)

[5 附录 8](#_Toc301301051)

[5.1 附件 8](#_Toc301301052)

[5.2 修过记录 8](#_Toc301301053)

# 引言

## 目标

在完成软件开发前期的准备工作如项目需求等之后，项目小组提出了这份软件设计说明书。

此设计说明书对Health的功能分配，模块划分，程序的总体结构，输入输出和接口设计，运行设计，数据结构设计及出错设计等方面作了全面的概括性的说明，同时作为系统分析员工作的阶段性总结和程序员进行开发及未来测试的重要文档资料。

## 文档范围

本文档包括软件的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

## 术语和缩略语

## 参考资料

部分内容参考了《范围规划》

## 软件目标和约束

软件目标：项目需要完成用户信息、搜索、找医生、找药品、咨询问答、评论等基本的功能。

软件的约束：设计范围比较广泛，实施非常困难。

# 软件设计

## 软件架构概述

该软件的架构采用三层架构的模式：用户界面层、业务逻辑层和数据访问层。采用面向过程的设计方法。

## 对象模型

该软件采用非面向对象的设计方法。

## 接口

## 特性实现

1. 用户注册
   1. 用户注册页面
   2. 注册成功，可选择登录
2. 医生登录
   1. 医生选择医师用户入口登录Health
   2. 登录成功，可选择进入个人信息界面修改自己的个人信息
3. 医师论坛
   1. 医生登录成功后选择医师论坛模块，进入论坛后医生可以看文章以及对文章评论
   2. 医生可以自己写文章然后发表
4. 普通用户登录
   1. 普通用户选择普通用户入口登录Health
   2. 登录成功，可选择进入个人信息界面修改自己的个人信息
5. 搜索功能
   1. 用户可在首页搜索框搜索医生、药品和疾病，可选择模糊搜索、推荐搜索或者按历史记录搜索
6. 找医生功能
   1. 用户进入首页的找医生模块
   2. 用户可直接搜索医生名字寻找医生
   3. 用户可根据疾病快速匹配医生
   4. 用户可根据科室匹配医生
   5. 可以根据医生的介绍从匹配到的医生中选择医生
7. 咨询医生
   1. 用户选择好医生后可以选择在线交流或者私人预约
   2. 选择在线交流进入交流界面
   3. 用户可以发文字、图片、语音、表情与医生交流
8. 用户寻找药品
   1. 用户进入首页的找药品模块
   2. 用户可直接搜索药品名称或者根据常备药推荐搜索
   3. 搜索之后可得到相关药品的列表，用户可查看药品的详细信息
9. 用户可以根据自己的问题快速提问
   1. 用户在首页选择快速提问模块
   2. 用户进入快速提问界面后，可以上传自己问题的描述，文字或者图片等，可选择是否公开问题
   3. 问题描述完成后可选择回答问题的医生
10. 疾病库功能
    1. 用户在首页选择疾病库模块
    2. 用户可直接搜索疾病，也可以搜索推荐疾病
    3. 用户搜索之后可得到疾病的相关介绍和相关科普文章推荐
11. 小课堂模块
    1. 用户进入首页的小课堂模块
    2. 用户可根据主题分类、热度推荐、内容介绍选择要观看的视频
    3. 用户可在观看后进行评价

## 错误代码

## 错误日志

## 部署视图

1. 硬件环境：
   1. 需要互联网
   2. 至少需要一台服务器
2. 软件：
   1. 需要SQL SERVER或者其他的数据库管理软件
   2. 需要WEB服务器
   3. 需要安装PHP解释器
3. 配置：
4. 部署：

# 数据库设计

## 逻辑模型

## 物理模型

1. 为每个表初始设置300M的存储空间，以10%的大小扩展。
2. 为上传的图片预留空间。

# 质量及其他方面

## 可维护性

1. 通过程序注释等方式增加代码的可读性和可维护性。
2. 将数据访问层分离，做成一个个函数，由其它层调用，以增加代码的可维护性。
3. 数据库有日志记录，系统一旦出现故障有恢复到故障之前的信息和数据的能力

## 安全性

1. 对用户的输入进行验证
2. 对用户的输入的特殊字符进行转义，防止sql注入攻击

## 可扩展性

1. 可以通过增加硬件资源的方式提高系统的响应速度。
2. 可以通过修改代码，扩充软件的功能。

## 可靠性

1. 数据库按时备份，有日志记录

## 可用性

1、做好充分的前期工作，软件一旦投入使用，尽量减少宕机的次数和时间

## 性能设计

1. 负载测试
   1. 负载测试主要是考察软件系统在既定负载下的性能表现
2. 压力测试
   1. 压力测试是为了考察系统在极端条件下的表现，极端条件可以是超负荷的交易量和并发用户数
3. 并发测试
   1. 并发测试是为了验证系统的并发处理能力。一般是和服务器端建立大量的并发连接，通过客户端的响应时间和服务器端的性能监测情况来判断系统是否达到了既定的并发能力指标
4. 稳定性测试
   1. 测试系统在一定负载下运行长时间后是否会发生问题
5. 可恢复性测试
   1. 测试系统能否快速地从错误状态中恢复到正常状态
6. 基准测试
   1. 当软件系统中增加一个新的模块的时候，需要做基准测试，以判断新模块对整个软件系统的性能影响

# 附录

## 附件

## 修过记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 |  |  |  |  |