在线看病设计规格说明

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 系统目标和约束 3](#_Toc301301032)

[2 系统设计 4](#_Toc301301033)

[2.1 系统架构概述 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 9](#_Toc301301038)

[2.6 错误日志 9](#_Toc301301039)

[2.7 部署视图 9](#_Toc301301040)

[3 数据库设计 11](#_Toc301301041)

[3.1 逻辑模型 11](#_Toc301301042)

[3.2 物理模型 11](#_Toc301301043)

[4 质量及其他方面 12](#_Toc301301044)

[4.1 可维护性 12](#_Toc301301045)

[4.2 安全性 12](#_Toc301301046)

[4.3 可扩展性 12](#_Toc301301047)

[4.4 可靠性 12](#_Toc301301048)

[4.5 可用性 12](#_Toc301301049)

[4.6 性能设计 13](#_Toc301301050)

[5 附录 13](#_Toc301301051)

[5.1 附件 13](#_Toc301301052)

[5.2 修过记录 13](#_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，结合经过调查的实际需求，项目小组提出了这份软件设计说明书。

此概要设计说明书对在线看病系统的功能分配，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，为软件详细设计奠定了基础，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

本文档包括系统的架构设计、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

部分内容参考了百度关于现代医疗状况的解答。

## 系统目标和约束

系统目标：

对于病人app：项目需要完成网上预订，在线下单，在线看病，支付，搜索医生等功能。

对于医生app：项目完成搜索病人，在线看病，接受订单等功能。

系统的约束：因为时间有限，该系统只实现最基本的功能。

# 系统设计

## 系统架构概述

该系统的架构采用三层架构的模式：视图层+业务逻辑层+数据访问层。采用面向过程的设计方法。

## 对象模型

该系统采用非面向对象的设计方法。

## 接口

docRegister医生注册

patRegister病人注册

logIn登录

searchAll普通搜索

searchDoc搜索医生

buyMed买药

consult咨询

showDetailDoc显示医生详细信息

ShowDetailMine显示自己的详细信息

## 特性实现

1. 医生注册
   1. 医生注册页面
   2. 注册成功，跳转到登录界面





1. 病人注册
   1. 病人注册界面
   2. 注册成功，跳转到登录界面





1. 病人搜索
   1. 病人在搜索界面输入自己想要搜索的信息
   2. 查看搜索结果
2. 预约医生

a)进入选择医生界面

b)选择医生

c)提问

1. 买药
   1. 进入买药界面
   2. 选择购买的药品
   3. 付款
2. 登录
3. 进入登录界面
4. 输入用户名密码，选择身份
5. 若信息输入无误进入首页

## 错误代码

# java.long.NullPointException

## 错误日志

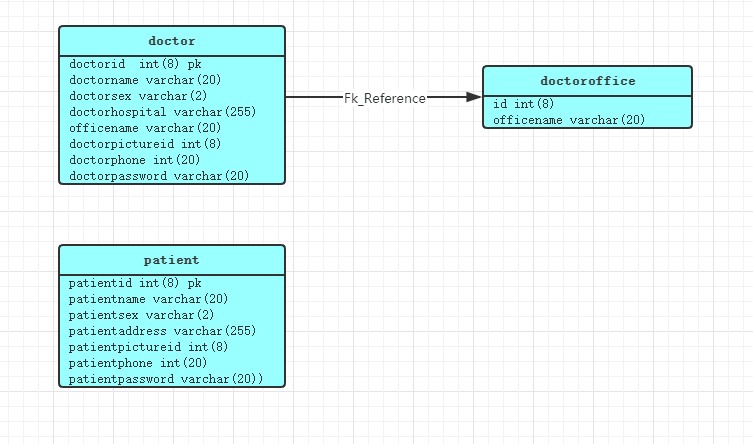
Java.net.binexception:address already in use

## 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. 需要安装mysql数据库的操作系统
   2. 需要Apache2.2作为WEB服务器。
3. 部署：
   1. 配置相应的配置文件，修改IP地址、数据库的用户名和密码。
   2. 将程序包放在Apache安装目录中得htdocs目录下。

# 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. 通过程序注释和对代码进行严格的审查等方式增加代码的可读性和可维护性。
2. 编写规范、完整、一致的程序文档，在软件生命周期的每个阶段都对文档进行检查评审
3. 使用MVC模式的设计，实现分层，降低代码的
4. 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. 数据库存放密码使用md5加密，不存储明文
2. 对用户的输入进行判定，看是否合法
3. 与数据库交互避免使用字符串拼接，防止SQL注入
4. 返回的提示信息要明确，出现异常进行处理，不能返回给用户

## 可扩展性

1. 业务逻辑要分层设计，降低耦合度
2. 实现一些常用的基础工具类，类似功能避免出现大量重复代码
3. 代码规范和命名规范要统一

## 可靠性

1. 数据库按时备份，有日志记录
2. 软件的设计开发阶段，要进过充分讨论和评审，确保架构和编码的严谨。
3. 上线前的测试工作，无论是功能测试还是性能测试，都无比充分进行
4. 软件在硬件上的部署，要全面考虑需要应对的风险

## 可用性

1. 做好充分的前期工作，系统一旦投入使用，尽量减少宕机的次数和时间
2. 开发时要时刻考虑应对故障
3. 考虑到用户可能会作出的所有操作，进行正确的反馈
4. 使用的操作门槛要符合大众的操作水平和能力

## 性能设计

确定功能特性：医生线上预约和看病

确定负载：1万以内

1、超过负载之后提示人数过载限制使用量

2、进行交互的数据统一使用JSON格式

# 附录

## 附件

## 修过记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 1.0 | 刘博文 | 202-4-20 |  |  |