牙科医院管理系统

系统测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2022/6/17 | 1.0 | 对牙科医院管理系统进行系统测试 | 李昱翰 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 4

4.1 需求覆盖率及缺陷分布 4

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

5.2 非功能性缺陷 6

6. 测试结论与建议 6

系统测试报告

# 简介

## 目的

本测试报告主要是对本课程本小组开发的牙科医院管理系统进行系统测试。

## 范围

本测试报告的主要范围是本牙科医院管理系统实现的主要功能。如对登录、挂号、并发挂号、信息查询(患者、医生、管理员)等功能进行的单元测试等。

## 定义、首字母缩写词和缩略语

定义1：系统测试：软件集成及集成测试完成后，对整个软件系统进行的一系列测试，称为系统测试

定义2：单元测试：又称为模块测试，是针对软件结构中独立的基本单元（如函数、子过程、类）进行的测试。

## 参考资料

暂无

## 概述

本测试报告主要由需求覆盖率分析、缺陷具体内容以及分布、测试结论及分析等几部分组成。

# 测试概要

测试时间：2022/06/17-2022/06-18,测试人员：本小组成员，测试内容：对本课程完成的功能性以及非功能性需求做出测试

# 测试环境

测试所使用的硬件配置为：使用Intel core-i7 CPU的联想THINKPAD笔记本电脑

软件配置：自动化测试可使用JUnit进行测试，使用JMeter进行并行测试

# 测试结果及分析

## 需求覆盖率及缺陷分布

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **登录** | 3 | 100% | 0 | 0% | 分别测试正常登录、用户名或密码错误、输入信息不完全三种情况 |
| **患者挂号** | 3 | 100% | 0 | 0% | 第一种测试为单机挂号测试，第二种为并发挂号，第三种为剩余诊号数量小于挂号人数 |
| **管理员发放诊号** | 1 | 100% | 0 | 0% | 主要测试了管理员能否正常挂号 |
| **管理员修改医生信息** | 2 | 100% | 0 | 0% | 主要测试正确输入修改信息以及未输入正确信息两种 |
| **医生查看诊号以及排班信息** | 1 | 100% | 0 | 0% | 主要测试医生在有具体排班的日期能否正确查看自己的排版以及挂号信息 |
| **医生修改诊疗流程** | 1 | 100% | 0 | 0% | 主要测试了医生能否对已经挂了自己号的患者当前流程进行修改 |
| **患者查看个人信息** | 1 | 100% | 0 | 0% | 主要测试患者能否正确查看自己的已挂号信息以及自己每个号对应的诊疗阶段 |
| **患者进行提问以及系统的答疑** | 1 | 100% | 0 | 0% | 主要是对答疑页面功能进行测试 |
| **功能项小计** | 10 | 100% | 0 | 0% |  |
| **非功**  **能项** | **性能** | 2 | 100% | 1 | 50% | 主要是对并发挂号时的系统性能进行测试 |
| **可靠性** | 0 | 0% | 0 | 0% |  |
| **界面易用性** | 8 | 100% | 2 | 25% | 页面基本实现易用性，但是并没有实现对于键盘操作的支持，也没有实现更多的辅助功能，只是实现了辅助朗读功能 |
| **界面兼容性** | 2 | 100% | 0 | 0% | 在两个浏览器中分别操作，检测功能是否能在不同浏览器中运行 |
| **安全性** | 0 | 0% | 0 | 0% | 没有对于系统的安全性做出测试 |
| **非功能项小计** | 11 | 75% | 3 | 25% |  |
|  | **总计** | 21 | 85% | 2 | 15% |  |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 0 | 0 | 1 | 1 | 2 |
| **占缺陷百分比** | 0% | 0% | 50% | 50% | 100% |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | 0 | 细微 |  |  | 在被次测试进行的功能测试中，未发现缺陷 |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

**表4 非功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **测试类型** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | 1 | 一般 | 易用性测试 | 未支持键盘操作 | 系统只支持了鼠标操作，而为支持可能存在的通过键盘来控制页面的情况。 |  |
| 2 | 2 | 严重 | 性能 | 需求覆盖不全面 | 在本次系统测试中，只测试了挂号部分的性能，但没有测试其余功能的性能，如发放诊号，患者提问的回复时间等 |  |
| 3 | 3 | 一般 | 安全性、部署性能等 | 非功能性测试测试方面不全面 | 在本次系统测试中，由于系统本质上都是组内开发组内运行，且并发都是使用软件进行模拟，故对于部署之后的性能无法测试，同时由于成员不会软件攻击的知识，无法测试安全性，只能简单的对应户进行鉴权，因而没有实现全覆盖而产生缺陷。 |  |

# 测试结论与建议

1. 遇到的问题：系统虽然基本实现了所要求的基本功能，但是由于组员无法进行安全性以及部署测试，使得系统真正发布后的性能无法得到保证。
2. 解决方式：继续学习相关的软件测试知识，并在小学期的课程大作业中进行运用。
3. 本次测试结论：基本实现了基本功能，基本满足了系统所要求的功能性需求，但是在非功能性需求方面还有较大缺陷。