|  |
| --- |
| 学生信息管理系统  **项目测试分析报告** |

|  |  |  |
| --- | --- | --- |
| 文件状态：[] 草稿 []正式发布 [√]正在修改 | 文件标识 | 学生信息管理系统-项目测试分析报告-2014-7-6 |
| 当前版本 | v1.0.0 |
| 作者 | 陈龙龙 |
| 完成日期 | 2014年7月 日 |

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 内容/备注 | 作者 |
| 2014-07-06 | v1.0.0 | 完成初稿 | 陈龙龙 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1引言

## 1.1 编写目的

测试分析报告即缺陷记录日志，主要记录系统的缺陷、错误和实现模块，同时系统维护人员可根据测试分析报告进行系统的维护，只要报告简明易懂，将大大缩短运维人员的工作时间。

## 1.2 背景

## 1.2.1项目的提出者

上海电力学院计算机科学与技术学院

## 1.2.2 项目的开发者

凤翔、陈龙龙

## 1.3 定义

1. 错误：指系统中缺功能或功能不能用
2. 缺陷：指系统中的功能能实现，但不满足条件
3. 最高优先级/高优先级：一般指错误
4. 中优先级/低优先级：一般指缺陷

## 1.4参考资料

[Node.js v0.10.29 Manual & Documentation](http://nodejs.org/api/)

[The MongoDB 2.6 Manual](http://docs.mongodb.org/manual/)

[Mongoose Schemas v3.8.12](http://mongoosejs.com/docs/guide.html)

[NODE.JS UNIT TESTING BY @朴灵](http://html5ify.com/unittesting/slides/index.html#/)

[Express 4.x API Reference](http://expressjs.com/4x/api.html)

[How to use templates with EJS]( https://code.google.com/p/embeddedjavascript/wiki/Templates)

[Bootstrap 中文文档](http://v3.bootcss.com/)

[LESS 语法](http://www.bootcss.com/p/lesscss/)

[Pro Git](http://git-scm.com/book/zh/)

[初识 mocha in NodeJS](http://cnodejs.org/topic/516526766d38277306c7d277)

[Mocha - the fun, simple, flexible JavaScript test framework](http://visionmedia.github.io/mocha/)

[Wikipedia](http://zh.wikipedia.org/)

[Stackoverflow](http://stackoverflow.com/)

本项目可行性研究报告

本项目开发计划

本项目需求规格分析说明书

本项目概要设计规格说明书

本项目详细设计规格说明书

本项目测试计划

# 2测试计划执行情况

## 2.1 测试项目

## 2.2 测试机构和人员

上海电力学院计算机科学与技术学院学生信息管理系统开发小组

凤翔、陈龙龙

## 2.3 测试结果

# 3软件需求测试结论

# 4评价

## 4.1 软件能力

## 4.2 缺陷和限制

## 4.3 建议

## 4.3 测试结论

# 二、测试概要

一份测试分析报告主要由四部分内容组成，分为为修订记录、测试结果统计列表、测试结果和测试详情。

## 2.1 修订记录

修订记录中记录的是该分析报告修改的内容及时间，是衡量任何一份报告的重要指标。通过修订记录可以很明显的查看出报告前后所发生的变化。

修订记录样板如下所示。

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 内容/备注 | 作者 |
| XXXX-XX-XX | v1.0.0 | 完成XXXX | XXX |
| XXXX-XX-XX | v1.0.1 | 修改XXX/完成XXX | XXX |

## 2.2测试结果统计列表

本列表主要针对某个系统的测试统计，也可针对某个角色的测试统计，测试完成后记录数量，使运维人员或管理者清楚的了解该系统的好坏。

|  |  |
| --- | --- |
| 测试统计结果列表 | |
| 系统名称 | XXXX系统 |
| 测试数量 |  |
| 通过数量 |  |
| 缺陷数量 |  |
| 错误数量 |  |
| 最高优先级数量 |  |
| 高优先级数量 |  |

## 2.3 测试结果

该列表主要显示有错误和缺陷的模块，该列表与2.4节中测试详情的信息有相对应的关系。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| XXXX系统详细测试结果 | | | | |
| 测试用例编号——测试编号 | 测试用例名称 | 种类（缺陷为蓝色，错误为红色，通过为绿色） | 缺陷编号 | 优先级 |
| X——XXX | XXXX测试用例 |  | XX |  |

## 2.4 测试详情

一般只记录存在错误或缺陷的用例，同时通过表格中的执行步骤能让运维人员清楚的了解错误发生在哪儿，出现了什么样的错误，以此能使运维速度大大增加。

|  |  |  |
| --- | --- | --- |
| 缺陷名称 | XXXX缺陷 | |
| 缺陷编号 |  | |
| 优先级 |  | |
| 测试用例名称 | XXXX测试用例 | |
| 测试用例编号——测试编号 | X——XXX | |
| 角色 |  | |
| 输入/动作 |  | |
| 期望的输出/相应 |  | |
| 实际情况 |  | |
| 应出现的界面 |  | |
| 执行过程 | 登录身份 | 执行界面 |
|  |  |

上表中详细列出了缺陷的信息，根据“缺陷名称”可大致了解是系统中出现了何种错误；根据“缺陷编号”可知缺陷序列如何；根据“优先级”可知道该缺陷所属的类型是缺陷还是错误；根据“测试用例名称”可知该缺陷发生在哪个测试用例中；根据“测试用例编号——测试编号”可知在该用例中的第几个步骤发生了错误；根据“角色”可知是在何种登录角色发生缺陷；根据“输入/动作”可知发生错误的前一个步骤是什么；根据“期望的输出/相应”可知希望出现的界面；根据“实际情况”可知输入动作结束后出现了怎样的界面；根据“应出现的界面”可知测试人员理想中的界面如何；根据“执行过程”可知登录的信息和执行步骤，使运维人员清楚的了解错误是如何发生的。

# 三、测试结果及发现

下面我们以用户信息维护模块为例，使大家能更深入的了解什么是测试分析报告。首先，我们从第七章中的2.1节可知，用户信息维护模块共有5个测试用例，而且每个用例都需进行测试，才能保证系统的完整性。随后根据第七章中的2.3节测试用例进行测试，每一个步骤都需测试，避免产生遗漏的项目，因2.3节中只实现了一个用例的测试用例，故其余四个用例仍需根据该用例原理编写测试用例，并进行测试。下面我将从测试的角度为大家主要阐述测试详情部分。

如下为一个测试用例的缺陷记录，请大家仔细阅读。

|  |  |  |
| --- | --- | --- |
| 缺陷名称 | 修改用户信息时，界面出现乱码情况 | |
| 缺陷编号 | 1 | |
| 优先级 | 高 | |
| 测试用例名称 | 修改用户信息测试用例 | |
| 测试用例编号——测试编号 | 4——XXX（编号由测试内容中的编号决定） | |
| 角色 | 管理员 | |
| 输入/动作 | 修改信息完毕，点击确定按钮 | |
| 期望的输出/相应 | 出现修改后的信息 | |
| 实际情况 | 界面中信息出现乱码 | |
| 应出现的界面 | 修改的信息应出现在界面中，不应出现乱码 | |
| 执行过程 | 登录身份 | 执行界面 |
| 用户名：1001  密码：12345 | 1. 管理员登录系统 2. 选择【用户信息】中的【用户列表】按钮，出现用户列表界面   小Q截图-20130708010344   1. 单击修改按钮，进入编辑状态   小Q截图-20130708010723  小Q截图-20130708010920   1. 修改后，单击确认按钮，返回用户列表界面，界面中显示乱码信息   小Q截图-20130708011318 |

上表中详细列出了缺陷的信息，根据“缺陷名称”可大致了解是系统中的修改功能出现乱码问题；根据“缺陷编号”可知该缺陷是本模块中测试阶段出现的第一个缺陷；根据“优先级”可知道该缺陷所属的类型是错误；根据“测试用例名称”可知该缺陷发生在修改用户信息测试用例中；根据“测试用例编号——测试编号”可知在该用例中的第几个步骤发生了错误；根据“角色”可知是管理员在进行修改操作时发生了错误；根据“输入/动作”可知发生错误的前一个步骤为单击确定按钮；根据“期望的输出/相应”可知希望出现的界面为修改成功页面；根据“实际情况”可知输入动作结束后出现了乱码的界面；根据“应出现的界面”可知测试人员理想中的界面是不要乱码；根据“执行过程”可知测试人员是通过怎样的步骤发现了乱码问题。

# 四、评价

一份好的测试分析报告可为运维人员提供很大的便利，根据表格的形式可清楚的了解一个系统在什么地方发生了怎样的错误，故测试分析报告十分重要。本系统的设计与实现并没有十全十美，还仍存在大大小小的问题，而测试与维护需在一段时间内不断的反复的进行。本次因时间有限，因此只有部分测试用例及测试分析，相信在今后，会以此为基础不断进行改进，最终使产品能使用。