|  |
| --- |
| 阿尔法小队 |
| 测试计划 |
| 测试组 |

|  |
| --- |
|  |

文档修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 变更内容 | 变更时间 | 变更人 |
| V1.0 | 创建 | 2017.10.23 | 胡忠宇 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 简介 3](#_Toc496802523)

[1.1 文档目的 3](#_Toc496802524)

[1.2 项目目标 3](#_Toc496802525)

[1.3 测试目标 3](#_Toc496802526)

[1.4 参考文档 3](#_Toc496802527)

[1.5 小组成员 3](#_Toc496802528)

[2 测试环境 4](#_Toc496802529)

[2.1 硬件环境 4](#_Toc496802530)

[2.2 软件环境 4](#_Toc496802531)

[3 测试过程 5](#_Toc496802532)

[3.1 主要使用测试方法 5](#_Toc496802533)

[3.2 使用测试工具 5](#_Toc496802534)

[4 测试计划 5](#_Toc496802535)

[4.1 单元测试 5](#_Toc496802536)

[4.2 集成测试 5](#_Toc496802537)

[4.3 回归测试 6](#_Toc496802538)

[5 测试日程 6](#_Toc496802539)

[6 缺陷处理 6](#_Toc496802540)

[6.1 记录日志，提交缺陷 6](#_Toc496802541)

[6.2 修复完成，复测验证 6](#_Toc496802542)

[7 风险分析 7](#_Toc496802543)

[8 系统发布 7](#_Toc496802544)

# 简介

## 文档目的

文档的目的是为了避免错误的发生，确保应用程序能够正常高效的运行， 能够发现至今未发现的错误；保证在以后的编写代码中不再有类似错误的存在。

## 项目目标

为了提高新生入学各项事宜的处理效率，在方便新生的同时，减少老师们的动作量。

待开发的软件系统的名称为：新生入学管理系统

项目的任务提出者：阿尔法队

项目开发者为：北京信息科技大学 阿尔法队

本系统的用户：健翔桥新生、班主任及辅导员

针对教学目的，本系统对实际的入学管理系统做了大量的简化。

## 测试目标

通过规范的、较完善的场景用例测试，确保基础业务流程的正常使用；通过其他方法测试系统的边界，保障系统对于权限具有一定的限制；通过模拟高并发保证系统的稳定性。

## 参考文档

《需求规格说明书v2.0》

《需求-模块表》

《模块说明》

## 小组成员

|  |  |  |
| --- | --- | --- |
| 姓名 | 职务 | 联系方式 |
| 胡忠宇 | 测试组长 | Qq |
| 西森 | 测试成员 | Qq |
| 崔照阳 | 测试成员 | Qq |
| 刘士源 | 测试成员、文档评审员 | Qq |

# 测试环境

## 硬件环境

服务器端子系统的运行要求：

系统软件： Window XP/7/8/10

数据库管理系统：SQL Server2000

硬件要求：Pentium III 450以上, 256M RAM, 40G HD

客户端子系统的运行要求：

系统软件： Window XP/7/8/10

硬件要求：Pentium III 450以上, 256M RAM, 10G HD

## 软件环境

开发工具：eclipse

版本控制工具：git

用例记录：excel

Bug记录、提交：excel

# 测试过程

## 主要使用测试方法

（1）等价类划分方法；

（2）边界值分析方法；

（3）错误推测方法；

（4）因果图方法；

（5）判定表驱动分析方法；

（6）正交实验设计方法；

（7）功能图分析方法。

## 使用测试工具

Findbugs、Weblink、Loadrunner等。

## 测试文档

测试计划、测试用例、缺陷记录等。

# 测试计划

## 单元测试

开发人员在自身功能开发完成之后，根据自己的代码编写相关的单元测试，确保功能在开发之初可以正常使用。

## 集成测试

测试组参考《需求规格说明书v2.0》等文档，撰写主要业务场景的测试用例。在开发的功能可以集成之后，在测试环境上集成开发组代码，并开始以用例为驱动的第一轮黑盒测试。在测试过程中继续完善测试用例，尝试编写自动化测试脚本，方便后续测试。

## 回归测试

为确定bug修复过程中与之前的工程没发生冲突，在所有bug修复完成、系统发布之前应该完成回归测试，通过前期整理的测试用例进行系统、完善的测试。

# 测试日程

|  |  |  |  |
| --- | --- | --- | --- |
| 事务 | 开始时间 | 结束时间 | 参与人员 |
| 单元测试 |  |  | 开发组 |
| 测试用例编写 |  |  |  |
| 集成测试 |  |  |  |
| 回归测试 |  |  |  |

# 缺陷处理

## 记录日志，提交缺陷

测试人员在发现系统bug时，应记录相关测试环境、测试数据，便于与开发人员交流时进行复现。向开发人员提交bug时，应提供相应错误日志、错误报告，并以邮件的形式提交开发组长，抄送测试组长。

## 修复完成，复测验证

开发人员在缺陷修复之后，更新测试环境，进行bug的复测验证，验证情况以邮件的形式提交开发组长，抄送测试组长。

# 风险分析

|  |  |  |
| --- | --- | --- |
| 序号 | 存在风险 | 应对方案 |
| 1 | 需求不具体 | 编写用例时及时与开发人员沟通，后期一律以测试用例为准。 |
| 2 | 开发进度缓慢 | 及时跟进开发进度，调整测试计划。 |
| 3 | 需求变更 | 及时沟通，及时修改相关用例，并作说明。 |
| 4 | 测试环境与开发环境差异大 | 安排开发人员协同配置测试环境。 |
| 5 | 熟悉系统时间长 | 安排测试组成员学习产品手册，并与开发多交流。 |

# 系统发布

发布之前应该完成一下目标：

需求均已覆盖；

所有测试用例至少执行过一次，并可执行；

所有测试具有相关文档记录，将明确的边界记录与产品手册；