Watermelon系统测试计划

一、文档介绍

1、项目名称

Watermelon

2、文档目的

本测试计划描述Watermelon进行系统测试的策略和方法。项目人员可以利用本文档来了解和执行测试活动，并进行管理。本文档定义测试人员的具体职责和活动，并描述测试类型和测试范围。

3、项目背景

“Watermelon”作为一个教育学习性平台，主要面向学生、求职者、算法爱好者等人群，为其提供所需要的算法或程序语言类题目的练习、同行者之间切磋的算法比赛以及对于自己的问题或看法进行发表的讨论区，不同许多其他的同类产品，本平台采用分布式开发将后端与判题机分离，提高判题效率；引入判重技术和hack功能，提高比赛的公平性和题目的正确性。打造一个为学习算法的人提锻炼提高自己算法能力的功能丰富的一站式平台。

4、参考文献

1. Watermelon的需求文档
2. 软件测试与质量教材

二、系统描述

Watermelon的具体功能：

（1）搜索题目。用户在题目列表界面输入题目的相关信息来搜索题目，相关信息可以是题目中的关键字，也可以通过勾选标签搜索特定方面的题目，标签可以勾选多个，标签和关键字可以结合使用。结果以题目列表形式给出，如果没有相关题目显示“没有相关题目”。

（2）提交代码。用户选择自己代码对应语言，将代码粘贴到文本框中提交给判题机进行判题。

（3）查看比赛。用户可以查看所有比赛的信息，信息包括比赛的简介，比赛的开始时间，比赛的结束时间，比赛的持续时间，比赛的难度，比赛的榜单，报名费用等。如果已经报名可以查看比赛的题目。

（4）查看提交记录。用户查看一道题目或者一场比赛的提交记录，提交记录包括提交时间、提交代码的评测时间、提交代码的评测空间、提交代码的结果，并且用户可以查看该提交记录的代码以及提交时所选择的语言。

（5）个人信息管理。用户可以对个人信息进行管理，个人信息包括昵称、做题偏好，最近做题情况，通过题目数等。管理包括个人信息的查看和修改。如果要修改头像，需要上传图片。

（6）查看题目。用户查看一道题目的具体信息，包括题目描述、时间限制、空间限制、输入描述、输出描述、样例输入、样例输出以及提示等。

（7）查看提交代码。用户通过题目编号、用户昵称、语言或者运行结果搜索查看自己或其他用户已提交过的提交记录，包括代码、题目、语言、运行时间、运行结果、所占内存、提交时间等。

（8）查看比赛榜单。用户查看比赛的榜单，可以看到每名参赛选手对应的罚时以及对应的名次，同时，用户可以查看每位比赛选手通过的题目，以及他们是在何时通过的该题目以及在对应题目上因错误而导致的罚时数。

（9）题目管理。对题库中的题目进行分类管理，分为可见和不可见两类，可见题目可以被用户所查看，不可见题目为尚未开始比赛中的题目。对题目的标签进行管理，同一题目可以包含多个标签，筛选出符合题目的标签。

（10）比赛管理。管理员可以添加一场比赛或者删除一场比赛。在添加比赛时，管理员可以为一场比赛制定比赛时间以及为该比赛添加相应的题目。同时，管理员可以修改一场比赛的信息，包括为该比赛添加或删除题目，修改比赛的描述信息以及开始时间与结束时间。

（11）判题系统管理。包括题目重判、测试数据提交。当管理员进行题目重判时，该题目的所有提交都会进行再次评测，新的评测结果将覆盖原有的评测结果。当用户提交测试数据时，提交测试数据的题目将会对所有提交记录进行重判。

（12）讨论区管理。分为帖子管理和查看帖子。用户对自己的帖子进行管理，包括创建帖子、删除帖子。用户通过相关信息搜索帖子，信息包括帖子标题的关键字、帖子创建者。

三、测试目标

1、测试目标

功能方面：

（1）需求规格说明书中所有的基本功能都正确实现

（2）保证系统中不存在需求文档没有要求的功能。

性能方面：

（1）系统处理一般业务操作时，响应时间不超过1秒，对于判题时间根据题目测试数据量决定。系统资源CPU和内存的利用率不超过90％；业务成功率100％。

（2）系统能够支持200人同时在线操作。

安全性方面：

（1）用户只能使用和访问已被授权的功能和数据信息。

（2）能正确的识别出恶意非法代码，防止对系统或数据库产生破坏。

易用性方面：

（1）要求系统界面直观、明了，布局合理了，可以让非计算机专业人员经过简单的培训即可掌握。

可靠性方面：

（1）全天24小时可用。

（2）保证比赛过程顺利，题目测试数据正确。

2、测试范围

参与测试的功能模块包括：

• 用户管理。

• 搜索题目。

• 提交代码。

• 查看比赛。

• 查看提交记录。

• 查看个人信息。

• 查看题目。

• 查看提交代码。

• 查看榜单。

• 题目管理。

• 比赛管理。

• 判题系统管理。

不参与测试的模块包括：

四、测试策略

1、测试方法

系统测试过程包括：功能测试、性能测试、安全性测试、容量测试。功能测试随着项目开发不断测试，分为单元测试和集成测试，单元测试是小组成员在编写完代码单独测试当前模块的功能；集成测试是在各模块集成之后对各模块的功能进行测试。功能测试采用手动测试。在功能测试后，分别对系统进行性能测试、安全性测试、容量测试。性能与容量测试采用自动化测试，安全性测试采用手动与自动化测试。测试具体的目标、方法和标准由下表列出：

（1）功能测试

|  |  |
| --- | --- |
| 测试目标 | ·各模块的功能单独实现  ·各模块集成后所有功能正确实现  ·模块与模块之间的接口正确 |
| 方法 | ·设计测试用例，手动检测结果是否正确  ·进行用户体验测试，用户判断功能是否正确实现 |
| 完成标准 | ·98%的测试用例通过  ·所有缺陷全部解决 |
| 需考虑的特殊事项 | ·在执行测试用例时，除此模块外也要判断其他模块是否发生错误 |

（2）性能测试

|  |  |
| --- | --- |
| 测试目标 | ·正常负载情况下，在除判题外的任意事务应在1s内得到响应，每次代码提交等待判题时间应小于5s，判题时间应小于该题时间限制\*10，每题判题结果应在提交后1分钟内返回；  ·高负载情况下，任意事务不得超过正常情况下3倍时间； |
| 方法 | ·编写脚本，通过curl等命令进行自动化测试事务响应时间；  ·通过脚本频繁发送请求，模拟高负载情况进行性能测试； |
| 完成标准 | ·在网络正常的情况下，系统响应时间应达成95%测试目标； |
| 需考虑的特殊事项 | ·性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测；  ·性能测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。 |

（3）安全性测试

|  |  |
| --- | --- |
| 测试目标 | ·用户和管理员只能使用已被授权的功能，访问已被授权的数据信息  ·系统可以正确识别用户提交的代码是否为恶意代码 |
| 方法 | ·手动提交恶意非法代码，判断系统能否识别  ·编写测试脚本，用自动化测试工具测试，各类型用户是否越过权限 |
| 完成标准 | ·98%的测试用例通过  ·所有缺陷全部解决 |
| 需考虑的特殊事项 | ·无 |

（4）容量测试

|  |  |
| --- | --- |
| 测试目标 | ·系统应支持不少于200个用户同时进行操作  ·判题队列小于300时，判题机应正常工作 |
| 方法 | ·结合性能测试，并使用自动交题脚本短时间内批量交题； |
| 完成标准 | ·达成95% 测试目标 |
| 需考虑的特殊事项 | ·批量提交应包含不同时间限制的题目 |

2、测试数据

测试数据由小组成员根据用户故事及用例文本描述进行设计。

3、测试通过 / 失败标准

测试通过的标准如下：

• 所有功能测试的测试用例都被执行并通过；

• 所有发现的缺陷都被修复并且通过回归测试；

• 性能指标达标；

• 安全性需求满足。

测试失败的标准如下：

• 缺陷密度大于10个/ KLOC；

• 发现有重大结构设计问题，修改结构会导致 20％以上的功能发生变化，进一步测试已经无意义；

• 关键功能出现缺陷，修复该缺陷会导致 20％以上的功能发生变化，进一步测试已经无意义。

五、测试组织

为了合理并且顺利的进行项目测试，将小组成员分为三个角色分别为：测试经理、测试设计员、测试员，一起合作完成功能测试、性能测试、安全性测试、容量测试等方面的工作。

|  |  |  |
| --- | --- | --- |
| 姓名 | 角色 | 职责 |
| 孙增奎 | 测试经理 | 跟踪项目的进度，协调项目组成员之间的合作；  负责软件测试团队管理工作；  项目提交测试后，项目经理需了解测试结果，根据测试的bug的严重程度来重新更改开发计划。 |
| 徐维彦 | 测试设计员 | 带领测试团队，设计、执行、优化测试过程,丰富测试手段，测试框架和测试策略；  与其他测试人员、项目管理人员沟通和协作，推动整个项目的顺利进行；  维护测试流程，统计和分析测试结果，提高测试效率和质量。 |
| 张愉飞 | 测试员 | 准确地定位并跟踪问题，推动问题及时合理地解决；  完成对产品的集成测试与系统测试，实施对产品的源代码缺陷检测、性能测试、安全性测试。 |

六、测试进度

本次测试计划自2020年3月26日开始，到2020年6月2日结束，由孙增奎、徐维彦、张愉飞参与，具体的测试进度安排如下表所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 任务 | 责任人 | 计划开始时间 | 计划工时/h | 实际开始时间 | 实际工时/h |
| 制定测试计划 | 孙增奎 | 2020.3.26 | 10 | 2020.3.26 | 6 |
| 评审测试计划 | 徐维彦 | 2020.3.28 | 10 |  |  |
| 设计功能性测试用例 | 张愉飞 | 2020.3.29 | 12 |  |  |
| 设计性能测试用例 | 徐维彦 | 2020.3.30 | 14 |  |  |
| 设计安全性测试用例 | 张愉飞 | 2020.3.31 | 15 |  |  |
| 设计测试数据 | 张愉飞 | 2020.4.1 | 12 |  |  |
| 评审测试用例和测试数据 | 孙增奎 | 2020..4.3 | 4 |  |  |
| 搭建测试环境 | 徐维彦 | 2020.4.5 | 14 |  |  |
| 执行冒烟测试 | 徐维彦 | 2020.4.15 | 16 |  |  |
| 执行功能（易用性）测试 | 孙增奎 | 2020.4.17 | 13 |  |  |
| 执行性能测试 | 张愉飞 | 2020..5.10 | 10 |  |  |
| 执行安全性测试 | 孙增奎 | 2020.5.20 | 11 |  |  |
| 进行回归测试 | 徐维彦 | 2020.5.25 | 7 |  |  |
| 跟踪监督、度量分析 | 张愉飞 | 2020.5.28 | 8 |  |  |
| 编写测试分析与总结报告 | 孙增奎 | 2020.5.30 | 12 |  |  |
| 提交测试工作产品 | 徐维彦 | 2020.6.2 | 1 |  |  |

七、项目风险分析说明表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序 号 | 风险说明 | 风险严重性 | 风险可能性 | 规避措施 | 应急计划 |
| 1 | 测试人员无法到位 | 高 | 中 | 定期催促测试人员进行资源协调； | 推迟进度  小组其他成员参与测试 |
| 2 | 测试人员技能不符合要求 | 高 | 低 | 通过博客学习相关技能 向其他组成员学习相关技能 | 加强学习 |
| 3 | 测试工具无法到位 | 中 | 中 | 尽量采用开源、免费、简单易用的开发工具 | 推迟进度  使用替代工具 |
| 4 | 项目开发延期 | 高 | 高 | 在项目进行过程中严格控制进度及时通知测试人员，协商解决 | 修改测试计划； 测试人员加班 |
| 5 | 项目中缺陷密度高 | 高 | 低 | 开发人员进行规范的单元测试与集成测试； | 修改测试计划； 协助开发人员修复缺陷 |
| 6 | 测试准备进度延误 | 高 | 低 | 测试人员严格按照计划执行 | 测试人员加班；  调整测试计划 |
| 7 | 发现难以修复缺陷 | 高 | 中 | 及早了解需求、设计与实现的模块，及早预测可能出现重大缺陷的模块 | 测试人员配合开发人员 进行解决 |
| 8 | 测试脚本bug | 高 | 高 | 编写脚本时，仔细分析逻辑并写注释  运行脚本前，进行脚本检查 | 项目回滚，修复脚本 |
| 9 | 测试版本落后于开发版本 | 中 | 低 | 对代码仓库设置提醒功能 | 及时更新测试版本 |
| 10 | 服务器崩溃 | 低 | 高 | 定期对数据进行备份 | 启用备份 |

八、测试环境

测试环境如表8-1所示。整个测试在Internet环境下进行，其中web应用服务器和数据库服务器可以由同一台计算机充当。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 主机用途 | 数量 | CPU | 内 存 | 硬 盘 | 安装软件 |
| web服务器 | 1 | 3.4 GHz | 4 GB | 至少 5 GB 剩余空间 | Ubuntu 16.04 JDK1.6 以上版本 Tomcat6.0 以上版本 |
| 数据库服务器 | 1 | 3.4 GHz | 4 GB | 至少 5 GB 剩余空间 | Ubuntu 16.04 mySql5.5 |
| 客户端 | 3 | 1.6 GHz | 2 GB | 至少 5 GB 剩余空间 | Ubuntu 16.04 chrome、firefox、safari Jmeter Badboy webScarab |

表8-1

九、辅助管理工具

使用AppScan工具进行自动化测试，并对发现的缺陷进行跟踪。

十、测试结束时提交的工作产品

本次系统测试结束时，应提交以下工作产品：

• 测试计划；

• 测试用例；

• 测试脚本；

• 缺陷报告；

• 测试总结报告。