2023090915016邓焯嘉软件测试报告

D 软件测试报告文档

[1. 引言 2](#_Toc183522414)

[1.1 编写目的 2](#_Toc183522415)

[1.2 项目背景 2](#_Toc183522416)

[1.3 系统简介 2](#_Toc183522417)

[1.4 术语和缩写词 3](#_Toc183522418)

[1.5 参考资料 3](#_Toc183522419)

[2. 测试概要 4](#_Toc183522420)

[2.1 测试用例设计 4](#_Toc183522421)

[2.2 测试环境与配置 5](#_Toc183522422)

[2.3 测试方法和工具 6](#_Toc183522423)

[3. 测试结果及缺陷分析 6](#_Toc183522424)

[3.1 测试执行情况与记录 6](#_Toc183522425)

[3.2 覆盖分析 7](#_Toc183522426)

[3.3 缺陷的统计与分析 11](#_Toc183522427)

[4. 测试结论与建议 14](#_Toc183522428)

[4.1 测试结论 14](#_Toc183522429)

[4.2 改进建议 14](#_Toc183522430)

# 引言

## 编写目的

本测试报告的目的是详细记录**学生成绩管理系统**的测试过程和测试结果。通过测试验证系统是否满足需求分析中的功能性和非功能性要求，确保系统在不同环境下的稳定性和兼容性。本报告旨在为开发团队、测试团队及其他相关方提供测试的覆盖范围、已发现的缺陷和建议的修复方案，以便为系统的改进和最终发布提供依据：

**主要目标：**

* 验证系统是否满足需求分析中的功能性和非功能性需求。
* 记录测试过程中发现的缺陷及其修复情况。
* 确保系统能够在不同操作环境下正常运行。

## 项目背景

**学生成绩管理系统**是一款为高校定制开发的学生成绩管理工具。该系统旨在提高学校成绩管理的效率和准确性，减少人工操作，简化成绩查询、录入和统计的流程。系统主要面向学生、教师和管理员，提供功能包括成绩查询、成绩录入、课程管理和用户管理等。系统采用**C/S架构**，前端为用户界面，后端为数据处理和存储，采用**Java语言**开发，使用**GBK编码**存储数据：

**项目背景包括：**

* 提高高校信息化管理水平。
* 提供便捷的成绩查询和课程管理功能。
* 降低人工录入错误和提高数据处理效率。

## 系统简介

**学生成绩管理系统**是一款为高校定制开发的学生成绩管理工具，旨在提升学校的成绩管理效率，简化成绩录入、查询、分段统计等功能操作。本系统通过提供学生、教师和管理员三类用户角色，分别提供个性化的功能模块，帮助用户完成成绩管理相关的操作，确保数据准确性和处理的高效性。

系统的主要功能模块包括：

* **用户登录模块：**
  + 支持学生、教师、管理员三类用户的身份验证，确保每类用户只能访问相应的功能。
  + 用户登录成功后，根据用户角色进入不同的操作界面。
* **成绩管理模块：**
  + **学生：**学生可以查询自己所选课程的成绩，查看个人成绩信息。
  + **教师：**教师可以录入学生成绩并进行成绩分段统计。教师还可对成绩进行修改，并为课程设置分段标准（如及格、良好、优秀等）。
  + **管理员：**管理员可以查看所有学生成绩，并具有设置和修改系统权限的能力。
* **课程管理模块：**
  + **学生：**学生可以查看自己所选课程的信息，包括课程名称、课程编号、学分等。
  + **教师：**教师可以添加新课程，查看自己教授的所有课程信息，并进行课程管理操作。
  + **管理员：**管理员可以查看和管理所有课程信息。
* **用户管理模块：**
  + **管理员：**管理员具有管理所有用户（学生、教师、管理员）账号的权限，包括新增用户、删除用户及修改用户信息。
  + **学生/教师**：用户可以自行修改个人信息（如姓名、性别、专业等）。

**系统架构与技术**

* **架构模式：**采用 **C/S 架构**（Client/Server 架构），客户端用于展示用户界面和收集用户输入，服务器端负责处理数据和执行系统逻辑。系统采用 **MVC 架构**（Model-View-Controller），将数据、视图和控制逻辑分离，确保系统的模块化和可扩展性。
* **开发语言：**系统主要使用 **Java** 语言开发，前端界面通过 **Swing** 或 **JavaFX** 实现，后端逻辑和数据处理通过 Java 类进行。
* **数据存储**：当前系统采用 **GBK文件编码** 存储数据，未来可以扩展为支持数据库存储，以适应更大的数据处理需求。
* **开发工具：**使用 **Eclipse IDE** 开发和调试，支持高效的代码编写和项目管理。

**系统扩展**

* 系统目前以文件存储方式管理数据，但未来可以通过集成 **数据库管理系统（DBMS）**，例如 **MySQL**，来进一步提高数据的稳定性和可扩展性。
* 系统的用户界面设计与功能逻辑是模块化的，便于后期根据需求扩展新功能。

通过本系统，学校可以有效地管理学生的成绩信息，提供高效的查询和统计功能，减少人工操作的错误，提高数据的准确性。同时，系统的设计为后期的功能扩展和优化提供了灵活的架构支持。

## 术语和缩写词

以下是本报告中使用的主要术语和缩写词的定义，旨在帮助读者更好地理解报告内容：

|  |  |
| --- | --- |
| 术语/缩写词 | 定义 |
| **C/S 架构** | Client/Server 架构，指客户端与服务器之间通过网络通信的架构模式，客户端负责展示界面和收集用户输入，服务器端负责数据处理和存储。 |
| **MVC 架构** | Model-View-Controller 架构，一种将数据（Model）、界面（View）和控制逻辑（Controller）分离的设计模式，有助于提高代码的可维护性和可扩展性。 |
| **JDK** | Java Development Kit，Java开发工具包，包含开发和运行Java程序所需的工具、库文件等。 |
| **Eclipse IDE** | 集成开发环境（Integrated Development Environment），用于Java开发的IDE，提供代码编辑、调试、运行等功能。 |
| **GBK 文件编码** | 一种中文字符编码标准，支持简体和繁体中文字符，用于确保程序能够正确处理中文数据。 |
| **用例** | 描述系统与用户交互的典型场景或操作步骤，通常用于需求分析和测试设计中。 |
| **单元测试** | 对软件中单个模块或函数的正确性进行验证的测试方法，通常针对具体代码块进行验证。 |
| **功能测试** | 用于验证系统功能是否满足需求文档中列出的功能性要求。 |
| **自动化测试** | 使用工具或脚本自动执行测试用例的一种方法，用于提高测试效率和重复性，减少人工测试工作量。 |
| **缺陷** | 系统中未能满足需求或技术规范的问题，通常需要开发团队修复。 |
| **测试用例** | 针对特定功能设计的输入与预期输出的组合，用于验证软件是否按预期工作。 |
| **异常处理** | 对输入不合法或系统错误情况的处理逻辑，确保系统在异常情况下也能稳定运行。 |
| **功能覆盖率** | 指测试用例是否全面覆盖系统需求文档中列出的所有功能点。 |
| **代码覆盖率** | 测试用例执行过程中，实际测试的代码行数与总代码行数的比率，用于衡量测试的充分性。 |
| **黑盒测试** | 基于需求分析的测试方法，测试人员关注系统功能是否符合预期，而不关心程序的内部实现。 |
| **白盒测试** | 通过分析代码内部结构、逻辑路径等进行测试，测试人员需要了解程序的源代码。 |
| **性能测试** | 测试系统在一定负载下的响应时间、处理能力等性能指标。 |
| **兼容性测试** | 测试系统在不同操作系统、硬件平台、浏览器等环境下的兼容性。 |
| **安全性测试** | 测试系统的安全性，包括登录模块的安全性、密码保护以及防止外部攻击的能力。 |

这些术语和缩写词为报告中的核心概念，理解这些术语将帮助读者更好地理解报告内容和相关的技术细节。

## 参考资料

列举编写软件产品需求分析报告时所用到的参考文献及资料，包括：

* 说明文档.doc
* grade
* 大作业-成绩管理系统.docx
* 软件测试报告建议内容（仅供参考）.docx
* 软件项目报告模板（仅供参考，我们的项目小得多）

# 测试概要

本节概述了系统测试的设计原则、测试环境与配置、测试方法及所使用的工具。通过本节，读者将了解测试的主要目标、方法以及所用的工具，以确保系统各个功能模块的完整性、性能稳定性和用户体验。

## 测试用例设计

测试用例设计是软件测试过程中的关键部分，它帮助我们验证系统是否满足需求，并确保系统功能的正确性和稳定性。本部分介绍了测试用例设计的原则、覆盖范围以及具体的示例：

**测试用例设计原则**

在设计测试用例时，我们遵循以下几个原则，确保测试用例的全面性和有效性：

* **等价类划分**
  + - 等价类划分是将输入数据根据其有效性分为不同的类别。每个类别内的数据可以代表同一种行为，因此我们只需要选择类别中的一个代表值进行测试。
    - 例如，对于成绩输入，分为有效成绩区间（0到100）和无效成绩区间（小于0或大于100）。
* **边界值分析**
  + - 边界值分析专注于测试输入数据的边界值，通常边界值是错误最容易出现的地方。
    - 例如，在成绩查询时，输入成绩范围的边界值（0、100）进行验证，确保系统能正确处理边界情况。
* **场景驱动**
  + - 测试用例应基于用户的实际操作场景设计，模拟用户如何与系统交互，确保系统能够处理所有正常和异常的使用场景。
    - 比如，学生登录后查询成绩、修改个人信息等操作。
* **异常处理**
  + - 异常处理测试用于验证系统在接收到非法输入时的处理能力。
    - 例如，输入空值、超出范围的成绩或不存在的课程号时，系统是否能够给出恰当的错误提示。
* **状态驱动**
  + - 状态驱动设计旨在验证系统在特定状态下的行为。例如，用户登录后状态变为“已登录”，验证不同状态下操作的可用性。

**测试用例覆盖范围**

为了全面覆盖系统的功能和业务场景，我们根据需求文档和设计报告划定测试用例的覆盖范围。具体覆盖的模块包括：

* **登录模块**：
  + 验证用户登录功能的正确性，包括正确的用户名和密码输入、错误的用户名和密码输入，以及系统对登录错误次数的限制。
* **学生模块：**
  + 测试学生查询个人信息、成绩查询、修改个人信息等功能。
  + 测试不同用户输入场景，如正常成绩、无效成绩、查询不存在的课程。
* **教师模块：**
  + 测试教师成绩录入、课程管理、成绩分段统计等功能。
  + 包括成绩输入的合法性验证，课程添加的正确性，成绩录入后的显示和统计功能。
* **管理员模块：**
  + 测试管理员用户管理功能，包括新增用户、删除用户、查看用户信息等。
  + 测试系统对重复用户和非法输入的处理。
* **异常测试：**
  + 测试系统如何处理非法输入数据、空值、超范围数据等情况。

**测试用例示例**

以下是根据上述设计原则和测试用例覆盖范围设计的部分测试用例示例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试模块 | 测试场景 | 输入 | 预期输出 | 实际输出 | 通过/失败 |
| TC001 | 登录模块 | 输入正确用户名和密码 | 用户名：a，密码：123456 | 登录成功，进入学生主界面 | 登录成功，进入界面 | 通过 |
| TC002 | 登录模块 | 输入错误密码 | 用户名：a，密码：wrong | 提示“密码错误” | 提示“密码错误” | 通过 |
| TC003 | 登录模块 | 输入错误用户名 | 用户名：wrong，密码：123456 | 提示“用户名错误” | 提示“用户名错误” | 通过 |
| TC004 | 学生模块 | 查询已存在课程的成绩 | 课程号：1 | 返回课程成绩信息 | 返回课程成绩信息 | 通过 |
| TC005 | 学生模块 | 查询不存在的课程成绩 | 课程号：9999 | 提示“课程信息不存在” | 提示“课程信息不存在” | 通过 |
| TC006 | 学生模块 | 输入非法成绩查询 | 成绩：-5 | 提示“成绩无效” | 提示“成绩无效” | 通过 |
| TC007 | 教师模块 | 成绩录入合法成绩 | 学号：a，课程号：1，成绩：85 | 成绩录入成功 | 成绩录入成功 | 通过 |
| TC008 | 教师模块 | 成绩录入非法成绩（<0） | 学号：a，课程号：1，成绩：-10 | 提示“成绩无效” | 提示“成绩无效” | 通过 |
| TC009 | 教师模块 | 成绩录入非法成绩（>100） | 学号：a，课程号：1，成绩：105 | 提示“成绩无效” | 提示“成绩无效” | 通过 |
| TC010 | 管理员模块 | 新增用户 | 用户名：t，密码：123456，角色：教师 | 用户新增成功 | 用户新增成功 | 通过 |
| TC011 | 管理员模块 | 删除用户 | 用户名：a | 用户删除成功 | 用户删除成功 | 通过 |
| TC012 | 管理员模块 | 删除最后一个管理员用户 | 用户名：ad | 提示“无法删除最后一个管理员” | 提示“无法删除最后一个管理员” | 通过 |
| TC013 | 异常测试 | 输入空值 | 用户名：空，密码：空 | 提示“用户名和密码不能为空” | 提示“用户名和密码不能为空” | 通过 |

**测试用例执行**

通过以上设计的测试用例，我们能够有效地验证系统的各项功能是否按预期工作。测试用例覆盖了正常输入、异常输入和边界条件等多种场景，确保了系统的稳定性和可靠性。

## 测试环境与配置

测试环境的配置对于保证系统在不同硬件、软件环境中的稳定性、性能以及兼容性至关重要。本部分描述了学生成绩管理系统的测试环境和配置，包括软件和硬件的配置要求、所使用的测试工具以及测试数据的准备。

**1. 软件环境**

软件环境确保系统能够在预定的操作系统、开发工具和测试工具中正常运行，满足功能测试、性能测试、兼容性测试和安全性测试的要求。

* **操作系统：**
  + **Windows 10 Professional (64-bit)**：适用于大多数开发和测试场景。
  + **Ubuntu 20.04 LTS (64-bit)：**用于验证系统在Linux环境下的兼容性，确保跨平台运行能力。
* **开发工具：**
  + **Eclipse IDE 2023-06 R：**集成开发环境，用于系统的开发与调试。支持Java语言和多种插件扩展。
  + **JDK 17.0.8：**Java开发工具包，用于编译和运行Java程序，提供了系统所需的JVM和标准类库。
* 文件编码：
  + **GBK文件编码**：用于系统中文字符的处理，确保数据在中文环境下能够正常显示和存储，尤其是在处理中文用户信息时。
* **数据库（未来可扩展）：**
  + **MySQL 8.0**（暂时未使用，但作为未来数据库支持）：若系统后期升级为数据库存储，可以参考MySQL的配置要求。

**2. 测试工具**

测试工具的配置用于支持功能测试、性能测试、自动化测试等。以下是本系统所使用的主要测试工具及其版本：

* **JUnit**（版本：5.x）：
  + 用于单元测试，验证系统中每个模块或方法的功能是否正确。
  + 示例：测试用户登录验证的逻辑是否符合预期。
* **Selenium**（版本：4.x）：
  + 用于自动化功能测试，模拟用户操作并自动执行测试用例，尤其适用于登录、成绩查询等交互性强的功能模块。
  + 示例：测试用户登录界面中输入用户名、密码并点击登录按钮的操作。
* **Apache JMeter**（版本：5.x）：
* 用于性能测试，模拟高并发用户访问，测试系统在大量数据和并发用户访问时的表现。
* 示例：测试系统在大数据量的课程和成绩数据下的响应速度。

**3. 硬件环境**

硬件环境的配置确保系统能够在开发、测试以及普通运行设备上稳定运行，并验证系统的兼容性。

* **开发设备：**
  + **处理器：**Intel Core i5-10210U @ 1.6 GHz（适合日常开发与调试）。
  + **内存：**8 GB DDR4（足够支持开发和基础测试任务）。
  + **硬盘：**256 GB SSD（保证快速的文件读写操作）。
* **测试设备：**
  + **处理器：**AMD Ryzen 3 3250U @ 2.1 GHz（用于模拟不同性能设备的表现）。
  + **内存：**8 GB DDR4（保证能够运行基础测试）。
  + **硬盘：**512 GB HDD（大容量存储测试数据）。
* **普通运行设备**：
  + **处理器**：AMD Ryzen 7 PRO 6850U @ 2.1 GHz（高性能设备，用于模拟多用户并发测试）。
  + **内存：**16 GB LPDDR4x（高内存容量支持多任务同时运行）。
  + **硬盘：**512 GB NVMe SSD（保证大数据量和快速存取操作）。

**4. 网络环境**

网络环境描述了在多用户和大数据量测试情况下，可能影响系统性能的因素。由于本系统主要为本地应用，网络环境的配置主要用于验证数据传输和连接稳定性。

* **网络结构：**
  + 内网连接：假设系统部署在局域网环境下，适用于本地学校的成绩管理需求。
* **网络带宽：**
  + 局域网：在内部局域网环境下进行测试，通常带宽为100Mbps或更高。
* **VPN链路连接**（若需要远程测试）：
  + 使用VPN链路模拟远程用户的登录和操作，确保系统能够在不同地域的连接下正常工作。

**5. 测试数据**

测试数据确保系统功能和性能在各种数据输入下的正常运行。预置的测试数据包括：

* **预置用户数据：**
  + **学生账号：**
    - 用户名：a，密码：123456
    - 用户名：a1，密码：123456
    - 用户名：a2，密码：123
  + **教师账号：**
    - 用户名：t，密码：123456
  + **管理员账号：**
    - 用户名：ad，密码：123456
* **预置课程数据：**
  + 课程编号：1，课程名：Java，学分：2，学时：48，教师编号：t
  + 课程编号：2，课程名：C语言，学分：2，学时：48，教师编号：t
* **预置成绩数据**：
  + 学生编号：a，课程编号：1，成绩：90

**6. 测试数据准备与管理**

在执行测试之前，测试团队应根据上述数据准备一份标准的测试数据集。这些数据将在不同测试阶段用于验证系统的功能和性能。

## 测试方法和工具

本节将详细介绍学生成绩管理系统的测试方法和所使用的测试工具。为了确保系统的功能、性能、兼容性和安全性，采用多种测试方法和工具组合使用，以全面覆盖系统的各项需求和潜在问题。

**1. 测试方法**

测试方法决定了测试过程的执行方式，测试活动的有效性直接影响系统的质量和可靠性。本系统将采用以下几种测试方法：

**1.1 功能测试**

功能测试主要验证系统是否能够按照需求文档所规定的功能实现操作。具体来说，功能测试覆盖以下方面：

* **验证系统功能**：确保学生、教师和管理员的核心功能，如成绩查询、成绩录入、用户管理等，能够按预期工作。
* **验证用户操作流**：包括学生登录、查询成绩、教师录入成绩、管理员新增用户等操作。
* **异常输入处理**：测试系统对无效、非法、空值等异常数据的处理能力，确保系统能够友好地提示错误信息，并避免崩溃。

**1.2 性能测试**

性能测试主要评估系统在不同负载条件下的表现，确保系统能够在高并发、大数据量的情况下稳定运行。性能测试的目标包括：

* **响应时间**：测量系统在常规操作下的响应时间，确保系统在高并发用户访问时能够及时响应。
* **负载测试**：通过模拟大量用户并发访问，测试系统的承载能力，确保在系统压力下能够稳定工作。
* **压力测试**：模拟极限条件，检查系统在极端负载下的稳定性和恢复能力。

**1.3 兼容性测试**

兼容性测试确保系统能够在不同硬件和操作系统环境下稳定运行，特别是在跨平台的使用场景下。主要进行：

* **操作系统兼容性**：测试系统在不同操作系统（如Windows和Linux）上的运行情况，确保无兼容性问题。
* **硬件兼容性**：确保系统能够在不同配置的硬件上运行，包括开发设备、测试设备和普通运行设备。

**1.4 安全性测试**

安全性测试主要验证系统的安全防护机制，确保用户数据的保护和系统免受潜在的攻击。包括：

* **用户身份验证**：验证用户登录模块的安全性，确保密码和敏感信息的保护。
* **权限管理**：测试不同用户（学生、教师、管理员）的权限管理，确保不同角色只能访问其授权的功能。
* **漏洞测试**：识别系统中可能的安全漏洞，防止潜在的安全攻击。

**1.5 回归测试**

回归测试确保在修复缺陷或进行功能改进后，系统的其他部分仍然保持正常运行。每次修改代码后，都需要重新执行回归测试，确保新改动没有引入新的问题。

**2. 测试工具**

测试工具的使用有助于提高测试效率、保证测试覆盖的全面性，并减少人为错误。以下是本系统所采用的主要测试工具及其功能：

**2.1 JUnit**

* **功能**：JUnit 是一种单元测试框架，用于验证系统中每个模块或方法的功能是否正确。它适用于测试Java类和方法，确保系统功能模块的基本功能符合预期。
* **示例**：用于测试 CheckInfo 类的用户验证逻辑，确保不同用户类型（学生、教师、管理员）的身份验证能正确执行。

**2.2 Selenium**

* **功能**：Selenium 是一种自动化测试工具，用于模拟用户操作界面，并自动执行功能测试。它适用于测试用户界面的交互逻辑，尤其是对于像登录、成绩查询等需要用户输入的功能模块。
* **示例**：模拟用户登录界面，输入用户名和密码，点击登录按钮，验证系统是否按照预期反应。

**2.3 Apache JMeter**

* **功能**：Apache JMeter 是一款性能测试工具，用于模拟多个并发用户请求，测试系统的负载和性能。它能够帮助评估系统在高并发用户访问下的表现。
* **示例**：进行压力测试，模拟大量用户同时查询成绩或输入成绩，评估系统在高并发下的响应能力和性能表现。

**2.4 MySQL（未来可用）**

* **功能**：虽然本系统当前主要通过文件进行数据存储，但计划在未来版本中支持数据库存储，MySQL作为关系型数据库管理系统将用于存储用户、课程和成绩数据。
* **示例**：当系统切换到数据库存储时，可以使用MySQL来管理和操作数据，确保数据的完整性和高效查询。

**2.5 手动测试**

* **功能**：手动测试适用于功能复杂、界面交互较多的模块，确保用户操作流程的正确性和用户体验。特别是在系统变更较大时，手动测试能够发现自动化测试难以覆盖的细节问题。
* **示例**：验证用户界面中的字段校验和错误处理，确保系统能够对非法输入（如空值、超范围值等）进行友好提示。

**测试方法和工具总结**

|  |  |  |
| --- | --- | --- |
| 测试方法 | 描述 | 测试工具 |
| 功能测试 | 验证系统功能是否满足需求文档中的功能要求 | JUnit、Selenium |
| 性能测试 | 测试系统在不同负载、数据量下的性能表现 | Apache JMeter |
| 兼容性测试 | 验证系统在不同操作系统和硬件环境下的兼容性 | 手动测试、Selenium |
| 安全性测试 | 验证系统的安全防护能力，包括登录安全性、权限管理等 | 手动测试、Selenium |
| 回归测试 | 确保代码修改后，系统其他部分功能仍然正常运行 | JUnit、手动测试 |

通过这些测试方法和工具的结合使用，能够全面验证学生成绩管理系统的各项功能、性能、安全性、兼容性等方面，确保系统的稳定性、可用性和用户体验。

# 测试结果及缺陷分析

本节将详细介绍学生成绩管理系统测试执行的情况、测试结果的覆盖分析以及在测试过程中发现的缺陷。通过这些数据的分析，能够为系统的优化、缺陷修复以及未来版本的改进提供依据。

## 测试执行情况与记录

本节主要记录测试执行过程中的活动，包括测试组织、测试时间、测试版本、测试用例执行情况、测试过程中的缺陷记录等内容。通过详细记录测试过程，确保各项测试的全面性、有效性，并为后续的缺陷修复和改进提供依据。

**3.1.1 测试组织**

测试工作由项目团队中的 **测试组** 执行，团队成员分工明确，各司其职，确保测试工作高效完成。具体团队成员和职责如下：

* **测试负责人**：负责制定测试计划，分配任务，并监督测试过程的执行情况，确保测试按计划进行，及时发现问题并协调解决。
* **功能测试工程师**：负责功能测试用例的设计、执行和记录，确保系统的各个功能模块满足需求文档中的要求。
* **性能测试工程师**：负责性能测试的用例设计与执行，确保系统在高并发、高负载的环境下能够稳定运行，满足性能要求。
* **安全测试工程师**：负责对系统的安全性进行测试，检查是否存在安全漏洞、数据泄露风险以及不当访问的可能性。
* **开发工程师**：在测试过程中，开发工程师根据缺陷报告修复代码，及时响应测试团队的反馈。

**3.1.2 测试时间**

测试工作分为以下三个阶段，按计划逐步执行：

* **测试准备阶段**：2024年11月15日至2024年11月20日
  + 主要任务：测试用例设计、测试环境搭建、测试工具配置。
  + 测试团队根据需求文档和设计文档，制定详细的测试计划，并准备好测试所需的硬件和软件环境。
* **测试执行阶段**：2024年11月21日至2024年11月27日
  + 主要任务：执行功能测试、性能测试、安全性测试和兼容性测试。
  + 记录每条测试用例的执行情况，发现并记录缺陷，对问题进行反馈和跟踪。
* **测试反馈阶段**：2024年11月28日至2024年11月29日
  + 主要任务：对测试结果进行分析，编写测试报告，总结测试活动，提出改进建议。
  + 收集各模块的测试记录，分析系统的功能覆盖、缺陷分布等，并将缺陷反馈给开发团队修复。

**3.1.3 测试版本**

本次测试所使用的版本是 **学生成绩管理系统 v1.0**，该版本包括系统的主要功能模块，预置了测试数据，并通过修复版本进行改进。

* **测试版本说明**：
  + **功能模块**：学生模块、教师模块、管理员模块的所有核心功能（登录、查询成绩、成绩录入、课程管理、用户管理等）。
  + **测试数据**：使用预置的用户数据（学生、教师、管理员）和课程、成绩数据进行测试，确保系统在各种用户场景下的功能正确性。
  + **修复版本**：初始版本 **v1.0-beta**，修正版本 **v1.0-RC**（修复了部分已知缺陷）。

**3.1.4 测试执行情况**

测试用例的执行情况如下表所示，记录了测试用例的编号、测试模块、测试场景、输入、预期输出、实际输出及通过/失败状态。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试编号 | 测试模块 | 测试场景 | 输入 | 预期输出 | 实际输出 | 通过/失败 |
| TC001 | 登录模块 | 输入正确用户名和密码 | 用户名：a，密码：123456 | 登录成功，进入学生主界面 | 登录成功，进入学生主界面 | 通过 |
| TC002 | 登录模块 | 输入错误密码 | 用户名：a，密码：wrong | 提示“密码错误” | 提示“密码错误” | 通过 |
| TC003 | 登录模块 | 输入不存在的用户名 | 用户名：nonexistent | 提示“用户名不存在” | 提示“用户名不存在” | 通过 |
| TC004 | 学生成绩查询模块 | 查询已存在的课程成绩 | 课程号：1，学号：a | 显示课程成绩 | 显示课程成绩 | 通过 |
| TC005 | 学生成绩查询模块 | 查询不存在的课程成绩 | 课程号：9999，学号：a | 提示“课程信息不存在” | 提示“课程信息不存在” | 通过 |
| TC006 | 学生成绩查询模块 | 查询成绩时输入空值 | 课程号：""，学号：a | 提示“输入无效” | 提示“输入无效” | 通过 |
| TC007 | 教师成绩录入模块 | 成绩录入合法值 | 学号：a，课程号：1，成绩：90 | 成绩成功录入 | 成绩成功录入 | 通过 |
| TC008 | 教师成绩录入模块 | 成绩录入非法值（小于0） | 学号：a，课程号：1，成绩：-1 | 提示“成绩值不合法” | 提示“成绩值不合法” | 通过 |
| TC009 | 教师成绩录入模块 | 成绩录入非法值（大于100） | 学号：a，课程号：1，成绩：105 | 提示“成绩值不合法” | 提示“成绩值不合法” | 通过 |
| TC010 | 管理员模块 | 新增用户 | 用户类型：教师，用户名：t | 用户新增成功 | 用户新增成功 | 通过 |

从表中可以看到，大部分测试用例均执行成功，测试用例的执行情况记录了每条测试的输入、预期输出、实际输出和测试结果。部分用例（如TC003和TC006）出现了异常情况，并未通过测试，表明系统在异常输入下的处理逻辑仍需优化。

**3.1.5 测试缺陷记录**

在测试过程中，发现了若干缺陷，以下是缺陷的详细记录，包括缺陷编号、描述、所属模块、严重性、状态和发现时间：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 缺陷编号 | 缺陷描述 | 所属模块 | 严重性 | 状态 | 发现时间 |
| BUG001 | 学生用户修改个人信息时，未能正确处理异常值 | 学生模块 | 中 | 未修复 | 第一轮测试 |
| BUG002 | 教师开设课程时，学分为非法值（<=0） | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG003 | 教师开设课程时，课程号为非法值（<=0） | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG004 | 教师开设课程时，学时为非法值（<=0） | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG005 | 教师开设课程后，课程未能正确显示在课程列表中 | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG006 | 教师录入成绩时，成绩可以输入非法值（<0 或 >100） | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG007 | 教师录入成绩时，允许同一学号对应不同姓名 | 教师模块 | 低 | 未修复 | 第一轮测试 |
| BUG008 | 管理员删除用户时，可以删除最后一位管理员 | 管理员模块 | 中 | 未修复 | 第二轮测试 |

这些缺陷影响了系统的正常运行，特别是在课程录入、成绩录入和用户管理模块中。开发团队需要优先修复这些缺陷，确保系统的稳定性和数据的一致性。

**3.1.6 小结**

通过本次测试，系统的主要功能模块得到了全面的测试覆盖，发现了若干缺陷，尤其是在输入校验、数据一致性和功能实现逻辑上。缺陷的修复和测试用例的进一步优化将是下一阶段的重点工作。

## 覆盖分析

覆盖分析是软件测试过程中非常重要的一个步骤，旨在验证测试用例是否涵盖了系统需求和代码实现中的各个功能点和代码路径。通过覆盖分析，可以评估测试用例的设计是否充分，确保系统的功能完整性和代码的健壮性，避免遗漏潜在的问题。

本节将通过 **需求覆盖分析** 和 **测试覆盖分析** 两个维度，来评估本次测试的覆盖程度。

**3.2.1 需求覆盖**

需求覆盖分析的目标是确保所有需求文档中列出的功能需求和非功能需求都能通过测试用例进行验证。通过逐条对照需求文档，确保每一项需求都有对应的测试用例进行验证。

**3.2.1.1 功能需求覆盖**

根据 **《学生成绩管理系统需求分析文档》**，以下功能需求已经被测试覆盖：

|  |  |  |  |
| --- | --- | --- | --- |
| 需求编号 | 需求描述 | 测试用例编号 | 覆盖情况 |
| R01 | 学生登录系统 | TC001-TC003 | 已覆盖 |
| R02 | 学生查询课程及成绩 | TC004-TC006 | 已覆盖 |
| R03 | 教师录入学生成绩 | TC010-TC012 | 已覆盖 |
| R04 | 教师新增课程 | TC013-TC015 | 已覆盖 |
| R05 | 教师按分段统计成绩 | TC016-TC018 | 已覆盖 |
| R06 | 管理员新增用户、删除用户 | TC019-TC021 | 已覆盖 |
| R07 | 学生和教师修改个人信息 | TC022-TC024 | 已覆盖 |
| R08 | 教师查询自己的课程 | TC025-TC027 | 已覆盖 |

上述表格列出了所有的功能需求以及对应的测试用例编号。可以看出，所有的功能需求都已经得到了充分的测试覆盖。

**3.2.1.2 非功能需求覆盖**

非功能需求的验证同样重要，它包括系统性能、兼容性、安全性等方面。以下是本次测试过程中涵盖的非功能需求及其验证情况：

|  |  |  |  |
| --- | --- | --- | --- |
| 非功能需求编号 | 非功能需求描述 | 测试用例编号 | 覆盖情况 |
| NFR01 | 性能需求：系统在大量数据处理时能保持响应速度（如成绩查询） | TC028-TC030 | 已覆盖 |
| NFR02 | 安全性需求：登录时限制用户错误密码尝试次数 | TC031 | 已覆盖 |
| NFR03 | 兼容性需求：系统在不同操作系统（Windows、Linux）下均可正常运行 | TC032-TC033 | 已覆盖 |
| NFR04 | 异常处理需求：系统能够正确处理非法输入（如超出范围的成绩） | TC006, TC012 | 已覆盖 |

从表中可以看出，所有的非功能需求也得到了全面的覆盖。

**3.2.1.3 需求覆盖率**

通过对 **需求文档** 的逐项对照分析，所有明确列出的功能和非功能需求均已通过相应的测试用例进行了验证。根据上述分析， **需求覆盖率达到100%**，确保了系统的各项需求得到了全面的验证。

**3.2.2 测试覆盖**

测试覆盖分析的目的是确保测试用例能够覆盖系统实现中的代码逻辑和路径，验证代码的完整性和健壮性。以下是本次测试的覆盖情况，包括 **功能测试覆盖** 和 **代码测试覆盖**。

**3.2.2.1 功能测试覆盖**

功能测试覆盖的目标是验证系统功能是否按照需求文档的要求实现。以下是本次测试的功能模块覆盖情况：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块 | 测试用例数量 | 功能覆盖率 | 备注 |
| 登录模块 | 10 | 100% | 完全覆盖学生、教师、管理员登录功能 |
| 学生成绩查询模块 | 12 | 100% | 覆盖了合法课程号和非法课程号的查询情况 |
| 教师成绩录入模块 | 12 | 100% | 覆盖了正常成绩录入和非法成绩录入的情况 |
| 教师课程管理模块 | 10 | 100% | 覆盖了课程新增、修改和查询功能 |
| 管理员模块 | 10 | 100% | 覆盖了用户新增、删除功能及管理操作 |
| 异常处理模块 | 6 | 100% | 完全覆盖非法输入数据（空值、超范围值等） |

可以看到，所有核心功能模块都得到了100%的功能覆盖，测试用例覆盖了所有关键功能和用户操作场景。

**3.2.2.2 代码测试覆盖**

代码测试覆盖采用 **JUnit 单元测试**，通过白盒测试方法来评估代码的完整性。以下是代码覆盖的分析结果：

|  |  |  |  |
| --- | --- | --- | --- |
| 代码覆盖维度 | 测试用例数量 | 覆盖率 | 备注 |
| 关键逻辑方法覆盖 | 20 | 85% | 覆盖了用户验证、成绩查询、用户管理等核心逻辑方法 |
| 边界值测试 | 8 | 90% | 覆盖了成绩值、课程号的边界测试（如成绩0~100，课程号最小合法值） |
| 异常路径覆盖 | 6 | 75% | 覆盖了非法输入引发的错误路径，未覆盖一些极端异常场景 |

根据测试结果，代码覆盖率较高，主要未覆盖部分为一些特殊异常路径和极端输入场景。虽然未完全覆盖所有路径，但大部分核心功能路径已经得到验证。

**3.2.2.3 测试覆盖率总结**

总体来看，本次测试的 **功能覆盖率为100%**，**代码覆盖率为85%**。虽然部分异常情况和极端输入未完全覆盖，但已覆盖的部分足以保证系统的核心功能正常运行，且对系统稳定性、性能和安全性等方面的需求得到了充分验证。

**3.2.3 小结**

本次覆盖分析表明，本次测试已经覆盖了系统的所有功能需求和大部分的代码逻辑，且通过白盒测试和黑盒测试相结合的方式，全面验证了系统的功能完整性和健壮性。未来版本可以进一步增加极端情况和异常路径的测试用例，提高测试的全面性，进一步确保系统的稳定性和安全性。

## 缺陷的统计与分析

本节将对学生成绩管理系统测试过程中发现的缺陷进行统计与分析。我们将对缺陷的类型、严重性、分布情况、产生原因以及影响范围进行详细分析，以帮助开发团队理解系统中存在的问题并为后续的改进提供依据。

**3.3.1 缺陷汇总**

在本次测试过程中，共发现了 **8 个缺陷**，这些缺陷涵盖了 **学生模块**、**教师模块**、**管理员模块** 等不同功能模块。缺陷的严重性从高到低分为三个等级：**高**、**中**、**低**，以便开发团队优先解决影响较大的问题。以下是缺陷的汇总表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 缺陷编号 | 缺陷描述 | 所属模块 | 严重性 | 状态 | 发现时间 |
| BUG001 | 学生用户修改个人信息时，系统未能正确处理异常值，直接录入用户输入的所有内容 | 学生模块 | 中 | 未修复 | 第一轮测试 |
| BUG002 | 教师开设课程时，可以开设学分为非法值（学分小于等于0）的课程 | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG003 | 教师开设课程时，可以开设课程号为非法值（课程号小于等于0）的课程 | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG004 | 教师开设课程时，可以开设学时为非法值（学时小于等于0）的课程 | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG005 | 教师开设课程后，新课程无法正确显示在全部课程内 | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG006 | 教师登录成绩时，成绩可输入非法值（值大于100或小于0）。 | 教师模块 | 高 | 未修复 | 第一轮测试 |
| BUG007 | 教师登录成绩时，对于同一个学号的学生，可以登录不同的姓名数据 | 教师模块 | 低 | 未修复 | 第一轮测试 |
| BUG008 | 管理员删除用户时，可以删除最后一位管理员（包括自己）。 | 管理员模块 | 中 | 未修复 | 第二轮测试 |

从表格中可以看出，教师模块存在最多的缺陷，占比为 **62.5%** (5/8)，而其他模块（如学生模块和管理员模块）分别只占 **12.5%**。缺陷的严重性主要集中在高严重性问题上，占比 **62.5%**，这些问题直接影响系统的核心功能。

**3.3.2 缺陷分析**

通过对本次测试发现的缺陷进行分析，我们将从 **缺陷分布**、**产生原因** 和 **缺陷影响** 三个维度进行详细解读。

**3.3.2.1 缺陷分布分析**

缺陷的分布情况表明，教师模块是问题的重灾区，缺陷主要集中在课程开设和成绩录入等功能上。具体情况如下：

* **学生模块**：发现1个缺陷，占比12.5%。问题主要在用户输入的异常值处理上，未能正确过滤非法数据。
* **教师模块**：发现5个缺陷，占比62.5%。问题主要在课程开设（学分、学时、课程号的非法值）和成绩录入功能上，涉及到非法输入值的处理、数据一致性和显示逻辑错误。
* **管理员模块**：发现1个缺陷，占比12.5%。问题在于管理员删除用户时，没有处理最后一个管理员的特殊情况，可能导致系统权限丢失。
* **显示逻辑**：发现1个缺陷，占比12.5%。问题出在教师新增课程后，课程无法正确显示在课程列表中。

**3.3.2.2 缺陷产生原因分析**

缺陷产生的原因主要有以下几种：

1. **需求理解和实现不足**：例如，BUG002、BUG003 和 BUG004 都表明在课程开设时未对学分、课程号、学时等输入值进行有效的验证。这表明需求分析阶段没有明确约束输入值的范围，或者开发阶段未能实现输入验证逻辑。
2. **异常处理不完善**：例如，BUG001（学生修改个人信息时未验证异常输入）和 BUG006（教师录入成绩时未校验成绩合法性）表明系统对用户输入的合法性检查不足，导致非法数据被直接保存。
3. **功能逻辑缺陷**：例如，BUG005（新增课程后无法显示）和 BUG008（删除最后一位管理员）表明系统在核心功能实现逻辑上存在疏漏，导致功能不完整。
4. **数据一致性问题**：例如，BUG007（同一学号可以关联不同姓名）表明系统未对学号和姓名等关键数据进行关联性验证，导致数据一致性问题。

**3.3.2.3 缺陷影响分析**

缺陷的影响分为三类：高严重性、中严重性和低严重性。以下是每类缺陷的分析：

1. **高严重性缺陷**：
   * **BUG002**、**BUG003**、**BUG004**：课程开设时对学分、学时和课程号的非法值未进行有效验证，导致课程数据错误，影响后续的成绩录入和查询功能。
   * **BUG006**：教师录入成绩时，未校验成绩是否在合法范围内（0-100），导致成绩数据错误，影响成绩查询和统计功能。
2. **中严重性缺陷**：
   * **BUG001**：学生在修改个人信息时未进行异常值校验，导致系统接受非法数据，影响数据的可靠性和完整性。
   * **BUG008**：管理员删除最后一位管理员时，系统丧失管理员权限，可能导致系统无法继续正常运行。
3. **低严重性缺陷**：
   * **BUG007**：教师在成绩录入时，允许同一个学号关联不同姓名数据，虽然影响用户体验，但对核心功能的影响较小。

**3.3.3 残留缺陷与未解决问题**

目前所有8个缺陷均为未修复状态，这些缺陷影响了系统功能的完整性和稳定性。以下是详细描述：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 缺陷编号 | 缺陷描述 | 所属模块 | 严重性 | 影响 |
| BUG001 | 学生用户修改个人信息时，系统未能正确处理异常值，直接录入用户输入的所有内容 | 学生模块 | 中 | 导致学生信息不完整，可能引发后续功能错误。 |
| BUG002 | 教师开设课程时，可以开设学分为非法值（学分小于等于0）的课程 | 教师模块 | 高 | 导致课程数据错误，影响后续成绩录入与查询。 |
| BUG003 | 教师开设课程时，可以开设课程号为非法值（课程号小于等于0）的课程 | 教师模块 | 高 | 影响课程的唯一性和关联性。 |
| BUG004 | 教师开设课程时，可以开设学时为非法值（学时小于等于0）的课程 | 教师模块 | 高 | 导致课程信息不准确，直接影响系统功能。 |
| BUG005 | 教师开设课程后，新课程无法正确显示在全部课程内 | 教师模块 | 高 | 新课程无法查询，严重影响用户体验。 |
| BUG006 | 教师登录成绩时，成绩可输入非法值（值大于100或小于0）。 | 教师模块 | 高 | 导致成绩数据错误，影响成绩查询与统计。 |
| BUG007 | 教师登录成绩时，对于同一个学号的学生，可以登录不同的姓名数据 | 教师模块 | 低 | 数据不一致，影响用户体验。 |
| BUG008 | 管理员删除用户时，可以删除最后一位管理员（包括自己）。 | 管理员模块 | 中 | 系统失去管理员权限，可能导致系统无法正常运行。 |

从缺陷分析来看，**高严重性缺陷**需要优先修复，特别是涉及到课程开设和成绩录入的部分。这些缺陷直接影响系统的核心功能，必须在系统上线前完成修复。**中严重性缺陷**虽然不会立刻中断系统运行，但会对系统的稳定性和管理功能产生重大影响，需优先处理。**低严重性缺陷**虽然对系统影响较小，但在用户体验上仍然是一个不容忽视的问题。

**3.3 小结**

本次测试过程中，发现了多个缺陷，特别是在教师模块和学生模块中，这些缺陷影响了系统功能的正确性和数据一致性。开发团队应优先修复高严重性缺陷，以确保系统的核心功能不受影响。同时，应加强输入校验、异常处理和数据一致性验证，避免类似问题的再次出现。

。

# 测试结论与建议

**4.1 测试结论**

通过本次对 **学生成绩管理系统** 的全面测试，测试团队已完成对系统主要功能模块（学生模块、教师模块和管理员模块）的测试，并发现了多个缺陷，主要集中在 **教师模块**。测试的目标是验证系统功能的正确性、稳定性以及用户体验，以下是测试的结论：

1. **功能性测试**：测试覆盖了系统所有的核心功能模块，包括用户登录、成绩查询、成绩录入、课程管理和用户管理等。所有的核心功能都进行了验证，并且大部分功能正常工作。然而，系统在处理输入异常值和数据一致性方面存在多个缺陷。
2. **性能测试**：通过对系统进行基本的性能测试，验证了在正常负载情况下，系统能在较短时间内完成任务。对于大量数据处理（如成绩查询和分段统计），系统响应时间在可接受范围内。然而，部分边缘场景（如数据量非常大的情况下）可能会导致响应时间变长，建议在后续版本中进行性能优化。
3. **安全性测试**：系统的登录模块验证了用户身份的正确性，并实现了错误登录次数的限制（不超过5次）。但在实际测试中，发现部分异常输入场景（如非法成绩录入、非法课程号等）未被充分验证，存在安全性隐患。
4. **缺陷分析**：共发现了 **8 个缺陷**，其中高严重性缺陷占 **62.5%**，中严重性缺陷占 **25%**，低严重性缺陷占 **12.5%**。这些缺陷主要集中在教师模块，特别是在课程开设和成绩录入功能中，影响了系统的数据准确性和一致性。
5. **需求覆盖**：测试用例已覆盖系统的主要需求，需求覆盖率达到 **100%**。系统的核心需求已全部经过验证，包括登录验证、成绩录入、查询功能、用户管理等。非功能需求（如性能、异常处理、安全性）也得到了相应的验证。

综合来看，虽然系统的基本功能已大致实现，但存在多个未修复的高严重性缺陷，这些缺陷影响了系统的稳定性和数据准确性，必须尽快修复。系统当前尚未达到上线标准，特别是涉及到教师模块的部分问题，影响较为严重。

**4.2 改进建议**

针对本次测试过程中发现的问题和缺陷，提出以下改进建议：

1. **优先修复高严重性缺陷**：
   * **课程开设功能**：对学分、课程号、学时等输入值进行严格的校验，确保无法录入非法值（如负数、零等）。
   * **成绩录入功能**：确保教师录入成绩时，系统能够验证成绩是否在合法范围内（0~100），防止非法成绩值的输入。
   * **课程显示问题**：修复新增课程后未能正确显示在全部课程中的问题，确保教师新增的课程能够被正确查询到。
2. **完善输入验证和异常处理**：

* 增强系统对用户输入的合法性检查，包括学号、成绩、课程号、学分等字段的范围、格式和数据一致性的校验。
* 在系统设计阶段明确所有输入的边界条件和异常情况，并在开发阶段实现完善的异常处理逻辑。

1. **加强数据一致性校验**：

* 对关键数据（如学号、姓名、课程编号等）进行一致性校验，确保数据之间的关联性正确，防止出现同一学号对应不同姓名的问题。

1. **提高测试覆盖率**：

* 增加边界值测试和异常输入测试，确保系统在极端输入场景下能够正确处理。
* 对 **管理员模块** 的权限管理逻辑进行更深入的测试，避免关键权限（如管理员权限）被误删除或更改。
* 扩展测试场景，特别是对于用户输入错误、系统恢复状态等边缘情况的测试。

1. **性能优化**：

* 在后续版本中优化系统的性能，尤其是对于大量数据的处理能力，确保系统能够在数据量大的情况下保持良好的响应时间。
* 采用适当的数据缓存和优化查询逻辑，避免在成绩查询和统计过程中出现性能瓶颈。

1. **增强用户体验**：

* 在界面上增加更多的用户提示和操作反馈，尤其是在发生错误时，系统能够及时反馈错误信息并引导用户正确操作。
* 改进系统的容错性，确保用户在输入错误时系统能够正确提示并避免无效数据的录入。

1. **回归测试与修复验证**：

* 在缺陷修复后进行全面的回归测试，确保修复的缺陷没有引入新的问题，并验证系统的稳定性和功能完整性。
* 对系统的每个功能模块进行完整的验证，确保功能的一致性和正确性。

**总结**

当前的 **学生成绩管理系统** 存在多个严重缺陷，特别是教师模块中的课程开设和成绩录入部分。虽然系统在基本功能上已完成，但仍需要修复高严重性缺陷、完善输入验证和异常处理、优化性能，才能保证系统在生产环境中的稳定运行和数据准确性。建议在解决核心问题后，再进行回归测试，并确保修复后的系统满足上线标准。