|  |
| --- |
| UESTC |
| 软件测试报告 |
| 软件工程大作业 |

|  |
| --- |
| 王哲宇 2023090912005  2024-11-30 |

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

本测试报告旨在详细记录 **学生成绩管理系统** 的测试过程及结果，以确保系统满足预期的功能需求、性能要求和用户体验目标。通过本报告，开发团队、测试团队以及项目相关方能够了解系统的测试覆盖范围、测试结果以及系统可能存在的问题，为系统的改进和发布提供依据。主要目标包括：

* 验证系统是否满足需求分析中的功能性和非功能性需求。
* 记录测试过程中的缺陷及其修复状态。
* 确保系统在不同的环境中正常运行。

## 项目背景

**学生成绩管理系统** 是为高校定制开发的一款信息化管理工具，旨在提升学生成绩管理的效率和准确性。系统的主要用户包括学生、教师和管理员，功能涵盖成绩查询、成绩录入、课程管理以及用户管理等操作。系统采用 **C/S 架构**，通过 Java 技术实现，主要功能模块包括：

* **学生模块**：学生可以查询自己的课程和成绩，并修改个人信息。
* **教师模块：**教师可以开设课程、录入学生成绩、并查询所教授的课程的全部信息。
* **管理员模块：**管理员负责用户管理（新增和删除用户）和查看全局信息。

本系统的开发由 **软件工程项目团队** 负责，测试阶段旨在确保系统上线以前没有重大的系统功能缺陷，并确保用户可以获得良好的且符合预期的用户体验。

## 系统简介

**学生成绩管理系统** 是一款面向高校的教学辅助工具，旨在简化成绩管理操作，提升数据处理效率。系统的核心功能包括：

* **用户登录：**支持学生、教师和管理员身份的登录验证。
* **成绩管理：**
  + 学生查询自己的课程成绩。
  + 教师录入学生成绩并进行分段统计。
* **课程管理：**
  + 学生查看所选课程。
  + 教师开设课程并查询课程信息。
  + 管理员查看所有课程。
* **用户管理**：
  + 管理员新增或删除用户。
  + 学生和教师修改个人信息。

系统采用 **Java 语言** 开发，使用 **Eclipse IDE** 作为开发工具，基于 **GBK 文件编码** 存储数据。未来系统可扩展为支持数据库存储。

## 术语和缩写词

表1-1 术语和缩写词定义表

|  |  |
| --- | --- |
| **术语\缩写词** | **定义** |
| **C/S 架构** | Client/Server 架构，客户端与服务器分离的一种架构模式。 |
| **MVC 架构** | Model-View-Controller 架构，分离数据模型、视图和控制逻辑的一种设计模式。 |
| **JDK** | Java Development Kit，Java 开发工具包，用于开发和运行 Java 程序。 |
| **Eclipse IDE** | 集成开发环境，用于 Java 应用开发，支持插件扩展和代码调试。 |
| **GBK 文件编码** | 一种中文字符编码标准，用于支持简体和繁体中文字符。 |
| **用例** | 用户与系统交互的典型场景或操作步骤的描述，常用于需求分析和测试设计。 |
| **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。 |
| **功能测试** | 验证系统功能是否满足需求的测试类型，通常基于用例执行。 |
| **自动化测试** | 使用工具或脚本执行测试用例的一种方法，用于提高测试效率和重复性。 |
| **缺陷** | 系统中未满足预期需求或技术规范的问题，通常需要开发团队修复。 |

## 参考资料

* 《软件工程：实践者的研究方法》（第8版） - Roger S. Pressman
* 学生成绩管理系统需求分析报告 - 软件工程项目团队
* 学生成绩管理系统详细设计报告 - 软件工程项目团队
* Java 官方文档 - Oracle Corporation
* Eclipse IDE 用户手册 - Eclipse 基金会
* 《软件测试教程》（第3版） - 张莉
* MySQL 官方手册 - Oracle Corporation

# 测试概要

本部分详细描述了测试用例设计的原则、范围及示例，明确了测试环境和配置要求，并列出了核心测试方法和工具。通过功能测试、性能测试、兼容性测试和安全性测试相结合的方式，确保系统的功能完整性和稳定性。

## 测试用例设计

测试用例设计基于 **需求分析文档** 和 **详细设计报告** 中的功能模块与用户操作场景。本系统采用 **黑盒测试** 方法，结合功能测试和异常测试，确保系统功能的正确性、健壮性和用户体验。以下为测试用例设计的主要原则和覆盖范围：

1. **测试用例设计原则**：

* **等价类划分**：根据输入条件划分有效和无效等价类，确保覆盖所有输入场景。
* **边界值分析**：针对输入数据的边界值进行测试，如成绩范围（0~100）。
* **场景驱动**：基于用例场景设计测试用例，确保覆盖所有用户操作流程。
* **异常处理**：验证系统在输入错误数据或出现异常情况时的处理能力。

1. **测试用例覆盖范围**：

* **登录模块**：验证用户登录的正确性，包括不同用户类型的身份验证。
* **学生模块**：包括课程查询、成绩查询、修改个人信息等功能测试。
* **教师模块**：包括成绩录入、课程管理、成绩分段统计等功能测试。
* **管理员模块**：包括新增用户、删除用户、查看课程等功能测试。
* **异常测试**：验证系统在输入非法数据（如空值、超范围值等）时的处理情况。

1. **测试用例示例**：

表2-1 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 | 成绩查询模块 | 输入不存在的课程号 | 课程号：9999 | 提示“课程信息不存在” | 界面无响应 | 失败 |

## 测试环境与配置

测试环境和配置确保系统能够在不同硬件、软件环境下稳定运行，同时验证其兼容性和性能。以下是本系统的测试环境和配置：

* **软件环境**：
* **操作系统**：
* Windows 10 Professional (64-bit)
  + Ubuntu 20.04 LTS (64-bit)
* **开发工具**：
  + Eclipse IDE 2023-06 R
  + JDK 17.0.8
* **文件编码**：GBK
* **测试工具**：
  + JUnit（单元测试）
  + Selenium（自动化功能测试）
  + Apache JMeter（性能测试）
* **硬件环境**：
* **开发设备**：
  + 处理器：Intel Core i5-10210U @ 1.6 GHz
  + 内存：8 GB DDR4
  + 硬盘：256 GB SSD
* **测试设备**：
  + 处理器：AMD Ryzen 3 3250U @ 2.1 GHz
  + 内存：8 GB DDR4
  + 硬盘：512 GB HDD
* **普通运行设备**：
  + 处理器：AMD Ryzen 7 PRO 6850U @ 2.1 GHz
  + 内存：16 GB LPDDR4x
  + 硬盘：512 GB NVMe SSD
* **测试数据**：
* 预置用户数据（学生、教师、管理员）：
  + 学生账号：a，密码：123456
  + 教师账号：t，密码：123456
  + 管理员账号：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
* 预置成绩数据：
  + 学生编号：a，课程编号：1，成绩：90

## 测试方法和工具

1. **测试方法**：

* **功能测试**：
* 验证系统是否满足需求分析中的功能性需求。
* 测试用例覆盖所有用户场景，包括正常操作和异常输入。
* **兼容性测试：**
  + 验证系统在不同操作系统（如Windows、Linux）和硬件配置下的兼容性。
* **安全性测试：**
  + 验证用户登录模块的安全性，确保密码和敏感信息的保护。

**2. 测试工具：**

* **JUnit**：
  + 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CheckInfo模块的用户验证逻辑。
* **Selenium**：
  + 用于界面自动化测试，模拟用户操作测试主要功能，如登录、查询成绩等。
  + 示例：测试登录界面中输入用户名、密码并点击登录按钮的操作。
* **手动测试**：
  + 针对用户界面和异常场景进行手动测试，确保用户体验和异常处理逻辑符合预期。

# 测试结果及缺陷分析

## 测试执行情况与记录

* + 1. **测试组织**

测试活动由项目团队的 **测试组** 负责执行，团队成员包括：

* **测试负责人：**负责制定测试计划、分配任务并监督测试过程。
* **功能测试工程师：**负责功能测试的用例设计与执行。
* **性能测试工程师：**负责性能测试和压力测试的执行与分析。
* **安全测试工程师：**负责系统安全性测试并分析潜在漏洞。
* **开发工程师：**负责修复测试过程中发现的缺陷。

测试活动分为以下阶段：

* **测试准备阶段**：设计测试用例、搭建测试环境。
* **测试执行阶段**：执行功能测试、性能测试和安全测试。
* **测试反馈阶段**：记录测试结果，提交缺陷报告并跟踪修复过程。
  + 1. **测试时间**

测试活动贯穿以下时间段：

* **测试准备阶段**：2024 年 11 月 15 日至 2024 年 11 月 20 日
* **测试执行阶段：**2024 年 11 月 21 日至 2024 年 11 月 27 日
* **测试反馈阶段：**2024 年 11 月 28 日至 2024 年 11 月 29 日
  + 1. **测试版本**
* **系统版本：**学生成绩管理系统 v1.0
* **测试版本说明：**
  + 功能模块：包括学生模块、教师模块和管理员模块的所有主要功能。
  + 测试数据：使用预置的用户、课程和成绩数据进行测试。
  + 修复版本：v1.0-beta（初始版本）、v1.0-RC（修正版本）

## 覆盖分析

覆盖分析是软件测试过程中的重要步骤，用于衡量测试工作对系统需求和代码实现的覆盖程度。通过需求覆盖和测试覆盖的分析，确保测试用例的设计和执行能够全面验证系统功能的正确性和完整性。以下是对本次测试的覆盖分析。

* + 1. **需求覆盖**

需求覆盖分析的目标是验证测试用例是否涵盖了需求文档中列出的所有功能需求和非功能需求。需求覆盖的关键在于确保每一项需求都有相应的测试用例进行验证。

* + - 1. **功能需求覆盖**

根据《学生成绩管理系统需求分析文档》，以下功能需求已被测试覆盖：

* 学生模块
  + 学生登录系统。
  + 查询个人信息，包括姓名、学号、性别、专业等。
  + 查询所选课程与成绩
  + 修改个人信息。
* 教师模块
  + 教师登录系统
  + 查询所教授课程信息，包括姓名、学号、性别、专业等。
  + 录入学生成绩。
  + 新增课程。
  + 按照成绩分段统计（不及格、及格、优良、优秀）。
* 管理员模块
  + 管理员登录系统。
  + 新增用户（学生、教师、管理员）。
  + 删除用户。
  + 查看用户全局信息。

对于上述所有功能需求，共设计了**约30条测试用例**，涵盖了系统的所有核心功能模块和用户交互场景。

**3.3.1.2 非功能需求覆盖**

以下非功能需求在测试中也得到了验证：

* 性能需求
  + 测试了在课程数据量较大（大于100条）时系统相应的查询时间。
* 安全性需求
  + 验证了用户登录模块的密码校验逻辑，包括错误登录次数的限制（不超过5次）。
* 异常处理需求
  + 测试了各模块对于非法输入值（如空值、超范围值、重复值）的处理。
* 兼容性需求
  + 系统在Windows 系统和 Linux 两种操作系统环境下的运行情况。

**3.2.1.3 需求覆盖率**

通过对需求文档的逐条对照分析，所有明确列出的需求均已被测试覆盖，需求覆盖率达到**100%。**

需求覆盖分析表如表3-1所示。

表3-1 需求覆盖分析表

|  |  |  |  |
| --- | --- | --- | --- |
| **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 | 管理员新增用户 | TC020-TC021 | 已覆盖 |
| R05 | 教师按分段统计成绩 | TC013-TC014 | 已覆盖 |
| R06 | 系统限制登录错误次数不超过 5 次 | TC002 | 已覆盖 |
| R07 | 系统对非法输入值的异常处理 | TC006, TC012 | 已覆盖 |

* + 1. **测试覆盖**

测试覆盖分析的目标是验证测试用例是否覆盖了系统实现中的代码逻辑和路径，确保代码的完整性和健壮性。本次测试覆盖分析分为以下两个方面：

**3.2.2.1 功能测试覆盖**

功能测试覆盖的重点是验证系统实现的主要功能是否通过测试用例得到了充分测试。本次测试采用 **黑盒测试** 方法，结合需求文档和设计文档，覆盖了以下功能模块：

* 登录模块
  + 测试用户身份验证的正确性，包括学生、教师、管理员三种用户类型的登录。
  + 验证登录错误次数限制功能。
* 成绩管理模块
  + 学生成绩查询功能，包括合法课程号、不合法课程号和空值的输入。
  + 教师成绩录入功能，包括合法成绩范围（0~100）和超范围值（如 -5、105）的处理。
  + 教师按分数段统计成绩的功能，测试了分段标准的合法性及统计的准确性。
* 课程管理模块
  + 教师新增课程功能，测试了课程号重复的处理逻辑。
  + 学生查询所选课程的功能。
* 用户管理模块
  + 管理员新增用户功能和删除用户功能，测试了字段完整性校验和重复用户编号的处理。

按功能模块统计，我们共设计了约50 条功能测试样例，覆盖了所有核心功能模块，功能覆盖率达到**100%**。

**3.2.2.2 代码测试覆盖**

代码测试覆盖采用 JUnit 单元测试 工具，结合 白盒测试 方法，分析了以下代码结构和逻辑路径：

* 关键逻辑方法覆盖：
  + CheckInfo 类：验证用户登录信息的逻辑。
  + GradeInfo 类：学生成绩查询逻辑。
  + AddUser 类：管理员新增用户的逻辑。
* 边界值覆盖：
  + 测试了成绩值的边界（如0和100）。
  + 测试了课程号的边界（如最小合法课程号和不存在的课程号）。
* 异常处理路径覆盖：
  + 测试了非法输入值（如空值、超范围值）引发的错误路径。

代码覆盖率统计中，我们的测试语句覆盖率达到85%，大部分代码逻辑都被测试覆盖，未覆盖部分为一些异常情况的边缘路径；分支覆盖率达到80%，部分分支条件由于缺少极端输入未完全覆盖。

**3.2.2.3 测试覆盖分析表**

测试覆盖分析表如表3-2所示。

表3-2 测试覆盖分析表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **模块** | **测试用例数量** | **功能覆盖率** | **代码覆盖率** | **备注** |
| 登录模块 | 10 | 100% | 90% | 未覆盖部分为某些登录异常路径。 |
| 学生成绩查询模块 | 12 | 100% | 85% | 测试了合法和非法输入，但极端输入场景未完全覆盖。 |
| 教师模块 | 12 | 100% | 80% | 未覆盖部分为新增课程功能中的某些边缘场景。 |
| 管理员模块 | 12 | 100% | 85% | 删除用户功能在极端输入（非法字符）场景下未完全覆盖。 |
| 异常处理模块 | 6 | 100% | 75% | 部分异常输入导致的系统状态恢复逻辑未被测试。 |

## 缺陷的统计与分析

本节对学生成绩管理系统测试过程中发现的缺陷进行详细统计与分析，内容包括缺陷的汇总、分类分析，以及测试后系统中可能存在的残留缺陷与未解决问题。通过对缺陷的分析，为开发团队和项目相关方提供改进系统的依据。

* + 1. **缺陷汇总**

在我们的测试中，发现了缺陷汇总表中记录的缺陷，缺陷汇总表如表3-3所示。本次测试共涵盖登录模块、学生模块、教师模块、管理员模块和系统性能。缺陷的严重性从高到低共分为三个等级：① 高：对系统核心功能或用户体验有重大影响，可能造成较为严重的后果；② 中：对系统功能或用户体验有一定影响，但不影响主要功能使用；③ 低：对用户体验或系统性能有轻微影响。

表3-3 缺陷汇总表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 |

* + 1. **缺陷分析**

本次测试共发现 8 个缺陷，涵盖学生模块、教师模块和管理员模块。其中，教师模块的缺陷数量最多，占比达 62.5%（5/8），且大部分为高严重性缺陷，直接影响系统的核心功能。以下从缺陷分布、产生原因和影响范围三个维度进行分析。

1. **缺陷分布分析**：学生模块有1 个缺陷，占比 12.5%，问题主要集中在用户输入的异常值处理能力不足；教师模块有5 个缺陷，占比 62.5%，问题主要集中在课程开设和成绩录入功能，包括非法输入值的处理、数据一致性问题及显示逻辑错误；管理员模块有1 个缺陷，占比 12.5%，问题与用户删除逻辑相关，可能导致系统管理员权限丢失；系统功能的显示问题有1 个缺陷，占比 12.5%，教师新增课程后，课程信息未正确显示在课程列表中。
2. **缺陷产生原因分析：**① 需求理解和实现不足：缺陷如 BUG002、BUG003、BUG004 都表明系统未对课程开设的输入值（学分、课程号、学时）进行有效校验。这表明需求分析阶段未明确约束输入值范围，或开发阶段未实现输入校验功能；② 异常处理不完善：缺陷如 BUG001（学生信息处理异常值）和 BUG006（成绩录入非法值）表明系统未能正确验证用户输入的合法性，导致异常数据直接保存；③ 功能逻辑缺陷：BUG005（新增课程未显示）和 BUG008（管理员删除自身）表明系统在核心功能的实现逻辑上存在疏漏，影响了功能完整性；④ 数据一致性问题：BUG007（同一学号录入不同姓名）表明系统未对关键数据（如学号和姓名）进行关联性验证，导致数据一致性问题。
3. **缺陷影响分析：**① 高严重性缺陷：BUG002、BUG003、BUG004相关的非法课程信息的录入会严重影响课程数据的正确性，可能导致后续的成绩录入和查询功能出错。BUG006相关的非法成绩值的录入会直接影响成绩统计和查询的准确性；② 中严重性缺陷：BUG001相关的异常值的保存可能导致用户数据混乱，影响系统的可靠性。BUG008代表着删除最后一位管理员会导致系统无管理员权限，存在数据管理和安全隐患。③ 低严重性缺陷：BUG007导致同一学号关联不同姓名会导致数据不一致，影响用户体验和系统的可信度。
   * 1. **残留缺陷与未解决问题**

本次测试后，所有 8 个缺陷均为未修复状态，对系统功能的完整性和稳定性有显著影响。表3-4为残留缺陷的详细描述。

表3-5 残留缺陷详细描述表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | 导致用户信息的完整性和可靠性下降，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 | 导致课程数据严重错误，影响后续成绩录入与查询功能。 |
| 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 | 课程标识错误，影响课程的唯一性与关联性。 |
| 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，直接影响系统的课程管理功能。 |
| 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 | 新增课程无法查询，核心功能无法完成，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 | 导致成绩数据严重错误，直接影响成绩管理和统计功能。 |
| 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，用户体验较差，但对核心功能影响较小。 |
| 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，可能导致系统无法继续正常运行，存在数据安全隐患。 |

针对缺陷，我们可知：修复高严重性缺陷需要重新设计输入校验和数据处理逻辑，当前开发周期内未能完成；部分需求未在设计阶段详细说明（如非法输入值范围），导致开发时未进行约束性处理；部分缺陷（如 BUG005 和 BUG008）未能在开发阶段发现，表明测试用例在功能逻辑覆盖上仍有不足。

对于高风险缺陷，（如 BUG002、BUG003、BUG004、BUG006）直接影响系统核心功能，必须优先修复，否则会导致系统的功能完整性和数据准确性严重受损。对于中风险缺陷，（如 BUG001 和 BUG008）虽然不会直接中断系统运行，但会影响用户体验和系统管理的稳定性。针对低风险缺陷，（如 BUG007）对系统运行的影响较小，但长期可能导致数据一致性问题，需在后续版本中修复。

# 测试结论与建议

## 测试结论

本次测试覆盖了系统的主要功能模块，包括学生模块、教师模块和管理员模块，发现了系统在输入校验、数据一致性和功能逻辑上的多个缺陷。然而，本次测试发现的 8 个缺陷均为未修复状态，修复率为 **0%**，系统尚未达到上线标准。由于核心功能模块存在多个高严重性缺陷，系统的功能稳定性和数据准确性无法得到保证。

## 改进建议

1. **优先修复高严重性缺陷：**① 对教师模块的课程开设和成绩录入功能进行全面校验，确保非法值无法录入；② 修复课程新增后无法显示的问题，保证核心功能的完整性；③ 限制非法成绩录入，确保成绩数据的准确性。
2. **完善异常处理逻辑：**① 增强系统对用户输入的合法性检查，包括字段范围、格式和数据关联的约束；② 在系统设计阶段明确需求边界，并在开发阶段实现异常校验。
3. **加强测试覆盖：**① 增加边界测试和异常测试用例，确保所有输入值和功能逻辑路径均被覆盖；② 对管理员模块的权限管理逻辑进行深入测试，避免关键权限被删除。
4. **优化性能与用户体验：**① 在后续版本中，优化系统对大数据量的处理能力及数据一致性问题；② 增强用户操作的提示信息，提高系统的易用性和容错性。

总而言之，当前系统存在多个高严重性缺陷，功能稳定性和数据准确性无法满足上线需求，建议优先修复核心问题并进行回归测试后再考虑上线。同时在后续版本中加强需求分析和测试覆盖，提升系统的健壮性和用户体验。