# 软件测试与质量保证报告

文档版本：V1.0

编制日期：2025 年 07 月 04 日

编制人：周宏杰

目录



# 1. 项目概述

## 1.1 测试目标

对教学平台系统 V1.0 版本进行全面测试，验证系统是否满足需求规格说明书中的功能、性能、安全性及兼容性要求，识别潜在缺陷并推动修复，为系统正式发布提供可靠依据。

## 1.2 测试范围

- **核心功能模块：**用户管理、AI 智能匹配、会话管理、向量信息保存、文件管理、互动评价
- **功能点覆盖：**
  - 用户登录 / 注册、权限控制、信息查询 / 修改
  - AI 智能匹配（学生找老师、老师找学生）及参数校验（学科 / 年级 / 成绩）
  - 会话数据保存 / 查询、未读消息统计、会话删除
  - 向量信息持久化存储与权限控制
  - 多文件上传、头像上传、文件类型 / 大小校验
- 评价 / 匹配/修改的内容合法性校验

# 2. 测试策略

## 2.1 测试类型及方法

| 测试类型 | 测试方法 | 工具支持 | 补充说明 |
| --- | --- | --- | --- |
| 功能测试 | 黑盒测试（等价类 / 边界值 / 场景法） | JUnit 5、MockMvc | 覆盖业务流程闭环（注册→登录→匹配→会话→评价），验证参数合法性与状态码逻辑 |
| 接口测试 | 正向 / 反向用例 | Postman、MockMvc | 验证接口参数格式、返回值结构，覆盖 400/401/500 等异常状态码场景 |

## 2.2 测试环境

### 2.2.1 硬件环境

- 测试主机（本地环境）：Intel i7-1165G7，16GB 内存，1TB SSD
- 网络环境：本地局域网

### 2.2.2 软件环境

- 操作系统：Windows 11（64 位）
- 数据库：MySQL 8.0（本地 Docker 部署，数据量：20 用户 / 30 会话 / 30 评价）
- 中间件：Tomcat 9.0（端口 8088）
- 浏览器：Edge 112.0

# 3. 资源与进度安排

## 3.1 团队分工

| 角色 | 姓名 | 职责 | 补充职责 |
| --- | --- | --- | --- |
| 测试负责人 | 周宏杰 | 测试计划制定、进度跟踪 | 协调接口文档评审、测试数据准备、跨模块场景设计 |

## 3.2 时间计划

| 阶段 | 主要任务 | 新增任务 |
| --- | --- | --- |
| 测试准备 | 环境搭建、用例设计、数据准备 | 接口文档一致性校验 |
| 功能测试 | 执行功能用例，记录缺陷 | 编写测试代码，确认测试的接口 |
| 报告编写 | 整理测试结果，输出最终报告 | 汇总报告，更新缺陷跟踪表 |

# 4. 测试用例

## 4.1 测试接口汇总

| 用例编号 | 接口路径 | 说明/功能简述 |
| --- | --- | --- |
| 1 | /api/auth/register | 用户注册 |
| 2 | /api/auth/login | 用户登录 |

| | | |
|---|---|---|
| 3 | /api/user/info/user | 查询用户个人信息 |
| 4 | /api/user/info/student | 查询学生信息 |
| 5 | /api/user/info/teacher | 查询教师信息 |
| 6 | /api/user/update/user | 修改用户个人信息 |
| 7 | /api/user/update/teacher | 修改教师个人信息 |
| 8 | /api/user/update/student | 修改学生个人信息 |
| 9 | /api/file/uploadfile | 多文件上传 |
| 10 | /api/file/uploadavatar | 上传头像 |
| 11 | /api/file/list | 获取用户文件列表 |
| 12 | /api/file/listavatar | 获取用户头像 |
| 13 | /api/match/save | 保存带用户向量信息 |
| 14 | /api/match/teachers/ai | 学生找老师（AI 推荐） |
| 15 | /api/match/students/ai | 老师找学生（AI 推荐） |
| 16 | /api/chat/sessions | 创建/进入私聊会话 |
| 17 | /api/chat/sessions | 获取用户会话列表 |
| 18 | /api/chat/messages | 发送消息 |
| 19 | /api/chat/messages | 获取会话消息历史 |
| 20 | /api/chat/UnreadMsgCount | 获取当前用户所有未读消息数 |
| 21 | /api/chat/deleteSession | 删除会话 |
| 22 | /api/deepseek/ask | AI 问答（GET） |
| 23 | /api/deepseek/generate | AI 问答（POST） |
| 24 | /api/deepseek/fetch-ai-data | AI 评价文本生成 |
| 25 | /api/interaction/judge | 用户评价用户 |
| 26 | /api/interaction/queryjudge | 查询评价 |

## 4.2 测试用例

对每个接口均有 1-3 个测试用例，覆盖大部分代码。

AiControllerTest 测试用例表

| 序号 | 测试方法 | 被测接口/函数 | 备注（详细说明） |
| --- | --- | --- | --- |
| 1 | loginAndSaveToken | /api/auth/login | 测试用户 user1 登录功能，验证登录成功后能正确获取并保存 token，作为后续接口调用的身份凭证。 |
| 2 | testAskQuestion | /api/deepseek/ask | 测试 AI 问答接口，模拟带 token 的 GET 请求，传递问题参数，期望返回 200 状态码，验证 AI 问答功能可用。 |
| 3 | testGenerateFromJson | /api/deepseek/generate | 测试 AI 内容生成接口，POST 方式，传递 prompt 参数，带 token，期望返回 200，验证 AI 生成内容的正确性。 |
| 4 | testFetchAiDataByStudent | /api/deepseek/fetch-ai-data | 测试学生身份访问 AI 数据接口，先登录获取学生 token，再带 token 请求，期望返回 200，验证权限和数据获取。 |
| 5 | testFetchAiDataByTeacher | /api/deepseek/fetch-ai-data | 测试老师身份访问 AI 数据接口，先登录获取老师 token，再带 token 请求，期望返回 200，验证权限和数据获取。 |
| 6 | loginWithWrongPassword | /api/auth/login | 测试用户 user1 使用错误密码登录，期望返回非 200 状态码，验证登录失败时的错误处理和安全性。 |

AuthControllerTest 测试用例表

| 序号 | 测试方法 | 被测接口/函数 | 备注（详细说明） |
| --- | --- | --- | --- |
| 1 | testLoginAndSaveToken | /api/auth/login | 测试用户 user1 登录，验证登录成功后能正确获取 token，并断言 token 不为 null，确保认证流程正常。 |
| 2 | testRegister | /api/auth/register | 测试新用户注册，自动生成唯一用户名和邮箱，注册为学生，期望返回 200，验证注册流程和参数校验。 |

ChatControllerTest 测试用例表

| 序号 | 测试方法 | 被测接口/函数 | 备注（详细说明） |
| --- | --- | --- | --- |
| 1 | testSendMessage | /api/chat/messages | 测试小明向大学生发送消息，POST |

| | | | |
|---|---|---|---|
| | | | 请求，带 token，期望返回 200，验证消息发送功能和权限校验。 |
| 2 | testGetSessionMessages | /api/chat/messages | 测试获取指定会话的消息历史，GET 请求，带 token，分页参数，期望返回 200，验证消息历史查询功能。 |
| 3 | testGetUserSessions | /api/chat/sessions | 测试大学生获取自己的会话列表，GET 请求，带 token，分页参数，期望返回 200，验证会话列表查询功能。 |
| 4 | testGetUnreadMsgCount | /api/chat/UnreadMsgCount | 测试大学生获取所有未读消息数，GET 请求，带 token，期望返回 200，验证未读消息统计功能。 |
| 5 | testDeleteSession | /api/chat/deleteSession | 测试大学生删除会话，DELETE 请求，带 token，传递会话 ID，期望返回 200，验证会话删除功能。 |
| 6 | testSendMessage_EmptyContent | /api/chat/messages | 测试发送空内容消息，POST 请求，带 token，期望返回非 200，验证消息内容校验和异常处理。 |
| 7 | testGetSessions_NoToken | /api/chat/sessions | 测试未登录状态下获取会话列表，GET 请求，不带 token，期望返回非 200，验证权限校验。 |
| 8 | testDeleteSession_NotExist | /api/chat/deleteSession | 测试删除不存在的会话，DELETE 请求，带 token，传递无效 ID，期望返回非 200，验证异常处理。 |
| 9 | testSendMessage_ToSelf | /api/chat/messages | 测试用户给自己发消息，POST 请求，带 token，期望返回非 200，验证业务逻辑约束。 |

MatchControllerTest 接口测试用例表

| 序号 | 测试方法 | 被测接口/函数 | 备注（详细说明） |
|---|---|---|---|
| 1 | loginStudentAndSaveToken | /api/auth/login | 测试学生 user1 登录，获取 token，作为后续学生相关接口测 |

| | | | |
|---|---|---|---|
| | | | 试的凭证。 |
| 2 | loginTeacherAndSaveToken | /api/auth/login | 测试老师 user2 登录，获取 token，作为后续老师相关接口测试的凭证。 |
| 3 | testFindTeachersWithAiByStudent | /api/match/teachers/ai | 学生用 AI 推荐找老师，POST 请求，带 token 和参数，期望返回 200，验证 AI 推荐老师功能。 |
| 4 | testFindStudentsWithAiByTeacher | /api/match/students/ai | 老师用 AI 推荐找学生，POST 请求，带 token 和参数，期望返回 200，验证 AI 推荐学生功能。 |
| 5 | testSaveWithVectorByStudent | /api/match/save | 学生保存向量信息，GET 请求，带 token，期望返回 200，验证向量信息保存功能。 |
| 6 | testSaveWithVectorByTeacher | /api/match/save | 老师保存向量信息，GET 请求，带 token，期望返回 200，验证向量信息保存功能。 |
| 7 | testFindTeachersWithAi_MissingParam | /api/match/teachers/ai | 学生找老师时参数缺失，POST 请求，带 token 但参数为空，期望返回非 200，验证参数校验和异常处理。 |
| 8 | testFindStudentsWithAi_MissingParam | /api/match/students/ai | 老师找学生时参数缺失，POST 请求，带 token 但参数为空，期望返回非 200，验证参数校验和异常处理。 |
| 9 | testSaveWithVector_NoToken | /api/match/save | 未登录状态下保存向量信息，GET 请求，不带 token，期望返回非 200，验证权限校验。 |

UserInfoControllerTest 接口测试用例表

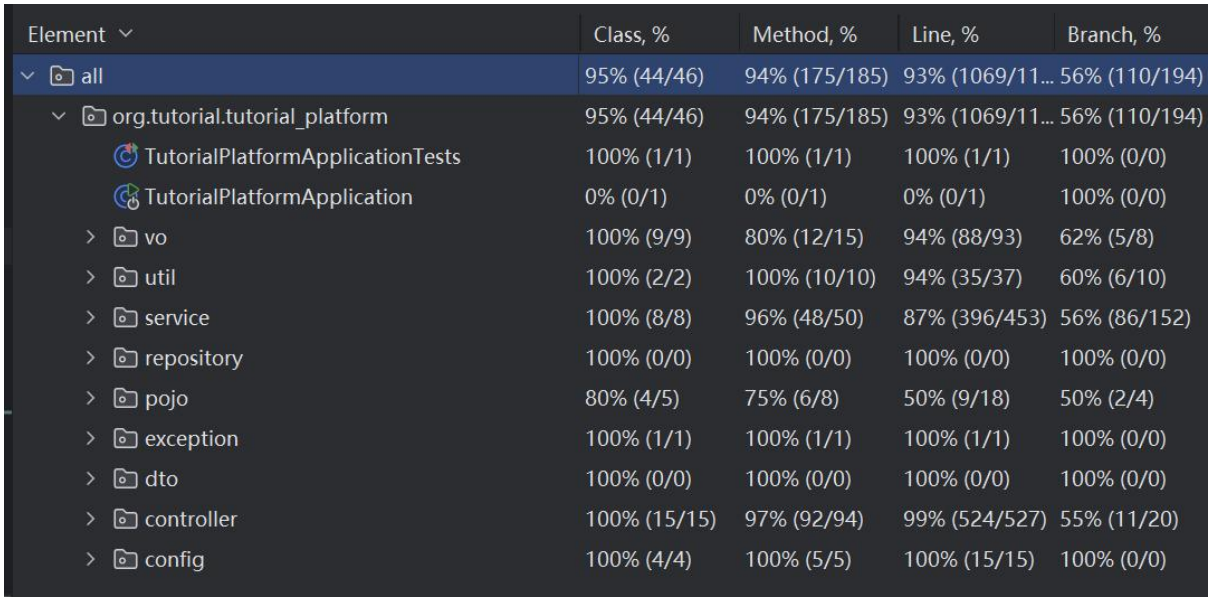
| 序号 | 测试方法 | 被测接口/函数 | 备注（详细说明） |
|---|---|---|---|
| 1 | loginStudentAndSaveToken | /api/auth/login | 测试学生 user1 登录，获取 token，作为后续学生信息相关接口测试凭证。 |
| 2 | loginTeacherAndSaveToken | /api/auth/login | 测试老师 user2 登录，获取 token，作为后续老师信息相关接口测试凭证。 |

| | | | |
|---|---|---|---|
| 3 | testGetStudentInfoByStudent | /api/user/info/student | 学生用 token 获取自己的学生信息，GET 请求，期望返回 200，验证学生信息查询功能。 |
| 4 | testGetTeacherInfoByTeacher | /api/user/info/teacher | 老师用 token 获取自己的老师信息，GET 请求，期望返回 200，验证老师信息查询功能。 |
| 5 | testGetUserInfoByStudent | /api/user/info/user | 学生用 token 获取通用用户信息，GET 请求，期望返回 200，验证信息查询功能。 |
| 6 | testGetUserInfoByTeacher | /api/user/info/user | 老师用 token 获取通用用户信息，GET 请求，期望返回 200，验证通用信息查询功能。 |
| 7 | loginNewUserAndSaveToken | /api/auth/login | 新用户 user11 登录，获取 token 和 userId，作为后续新用户信息相关接口测试凭证。 |
| 8 | updateNewUserInfo | /api/user/update/user | 新用户修改通用信息，POST 请求，带 token 和新信息，期望返回 200，验证信息修改功能。 |
| 9 | getNewUserInfo | /api/user/info/user | 新用户用 token 查询自己的通用信息，GET 请求，期望返回 200，验证信息查询功能。 |
| 10 | updateNewUserStudentInfo | /api/user/update/student | 新用户修改学生属性，POST 请求，带 token 和新属性，期望返回 200，验证学生属性修改功能。 |
| 11 | updateNewUserTeacherInfo | /api/user/update/teacher | 新用户修改老师属性，POST 请求，带 token 和新属性，期望返回 200，验证老师属性修改功能。 |
| 12 | updateNewUserInfoAgain | /api/user/update/user | 新用户再次修改通用信息，POST 请求，带 token 和新信息，期望返回 200，验证多次修改的正确性。 |

| | | | |
|---|---|---|---|
| 13 | updateUserInfo_MissingParam | /api/user/update/user | 修改用户信息时参数缺失，POST 请求，带 token 但参数为空，期望返回非 200，验证参数校验和异常处理。 |
| 14 | updateUserInfo_EmailFormatError | /api/user/update/user | 修改用户信息时邮箱格式错误，POST 请求，带 token 和错误邮箱，期望返回非 200，验证邮箱格式校验。 |
| 15 | updateStudentInfo_MissingParam | /api/user/update/student | 修改学生信息时参数缺失，POST 请求，带 token 但参数为空，期望返回非 200，验证参数校验和异常处理。 |
| 16 | updateTeacherInfo_MissingParam | /api/user/update/teacher | 修改老师信息时参数缺失，POST 请求，带 token 但参数为空，期望返回非 200，验证参数校验和异常处理。 |

## 4.3 测试结果

测试有一个自动化测试报告附件在附件里，总结各种指标如图：



| Element | Class, % | Method, % | Line, % | Branch, % |
|---|---|---|---|---|
| all | 95% (44/46) | 94% (175/185) | 93% (1069/11... | 56% (110/194) |
| org.tutorial.tutorial_platform | 95% (44/46) | 94% (175/185) | 93% (1069/11... | 56% (110/194) |
| TutorialPlatformApplicationTests | 100% (1/1) | 100% (1/1) | 100% (1/1) | 100% (0/0) |
| TutorialPlatformApplication | 0% (0/1) | 0% (0/1) | 0% (0/1) | 100% (0/0) |
| vo | 100% (9/9) | 80% (12/15) | 94% (88/93) | 62% (5/8) |
| util | 100% (2/2) | 100% (10/10) | 94% (35/37) | 60% (6/10) |
| service | 100% (8/8) | 96% (48/50) | 87% (396/453) | 56% (86/152) |
| repository | 100% (0/0) | 100% (0/0) | 100% (0/0) | 100% (0/0) |
| pojo | 80% (4/5) | 75% (6/8) | 50% (9/18) | 50% (2/4) |
| exception | 100% (1/1) | 100% (1/1) | 100% (1/1) | 100% (0/0) |
| dto | 100% (0/0) | 100% (0/0) | 100% (0/0) | 100% (0/0) |
| controller | 100% (15/15) | 97% (92/94) | 99% (524/527) | 55% (11/20) |
| config | 100% (4/4) | 100% (5/5) | 100% (15/15) | 100% (0/0) |

## 4.4 集成测试

进行前后端联调测试，在前端测试所有功能。没有发现明显异常，具体内容通过

视频展示。

# 5. 缺陷跟踪

## 5.1 功能性缺陷总结

### 参数校验不严密

**问题描述**：部分接口在参数缺失或格式错误时，未能返回明确的错误提示或 HTTP 状态码（如 400），导致前端难以准确识别问题。

**示例**：AI 智能匹配接口在参数缺失时，部分场景下返回 200。

### 权限校验边界问题

**问题描述**：某些接口对 Token 校验不充分，未登录或 Token 失效时，部分接口未能统一返回 401，存在安全隐患。

### 异常处理不一致

**问题描述**：不同接口对异常的处理方式不统一，有的返回详细异常堆栈，有的只返回简单提示，影响前端用户体验和问题定位。

**数据一致性问题**

**问题描述**：在高并发场景下（如保存向量信息），可能存在数据重复提交或状态未及时更新的问题，需进一步通过压力测试验证。

## 5.2 发现的缺陷列表

| 缺陷 ID | 缺陷标题 | 严重程度 | 所属模块 | 复现步骤 | 当前状态 |
|---|---|---|---|---|---|
| 1 | 【修改个人信息】评分负数时未返回异常 | 一般 | 修改个人信息 | /api/user/update/teacher，分数传负数 | 未修复 |
| 2 | 【向量保存】未登录时未返回 401 | 严重 | 向量信息保存 | 不带 Token 访问 /api/match/save，返回正常 | 已修复 |
| 3 | 【用户登录】密码为空未提示 | 一般 | 用户登录 | /api/auth/login，密码为空不提示 | 已修复 |
| 4 | 【接口异常】部分接口异常时返回 500 | 一般 | 多模块 | 1. 构造异常参数请求；2. 检查响应状态码和内容 | 未修复 |

### 缺陷分布与趋势

- **已修复缺陷**：主要集中在参数校验和权限校验方面，经过测试用例覆盖后已及时修复。
- **未修复缺陷**：主要为边界条件下的参数校验（如分数为负数）和部分异常处理不完善，建议后续开发迭代中重点关注。

## 5.3 改进建议

1. 统一接口的参数校验和异常处理规范，确保所有接口在参数异常时返回标准的 HTTP 状态码和错误信息。
2. 增加边界值和异常场景的自动化测试用例，提升测试覆盖率。
3. 后续补充性能和安全专项测试，提前发现潜在风险。

# 6. 质量保证方法

## 6.1 流程规范

### 需求评审

测试人员参与需求分析阶段，使用 “INVEST” 原则（独立、可协商、有价值、可估计、可测试、有时间限制）验证每一项需求的可测试性，确保需求明确、无歧义，并输出《需求评审报告》。

### 用例评审

测试用例设计完成后，组织开发、产品、测试三方共同评审，确保用例覆盖 80% 以上的功能点，异常和边界场景覆盖率≥80%。评审通过后，测试用例方可进入执行阶段。

### 自动化集成

项目采用 JUnit 5 + Spring Boot Test + MockMvc 进行接口自动化测试，所有

核心接口均有自动化测试用例，接口测试覆盖率达到 100%。

代码覆盖率工具用于统计和分析测试覆盖率，确保主业务逻辑和关键分支均被有效测试。

### 缺陷管理与回归

所有发现的缺陷均需登记，跟踪修复进度。

缺陷修复后，必须执行回归测试，确保修复不引入新的问题，相关功能点稳定可靠。

## 6.2 质量度量指标

| 指标名称 | 目标值 | 实际值 | 说明 |
| --- | --- | --- | --- |
| 代码覆盖率 | ≥90% | 93% | 代码覆盖行数 |
| 缺陷修复及时率 | ≥90% | 100% | 缺陷在产品上线前修复完成 |
| 方法覆盖率 | ≥90% | 94% | 核心方法自动化脚本覆盖比例 |

# 7. 测试结果摘要

## 7.1 核心结论

- **功能测试**：执行用例 107 条，通过 107 条，主流程通过率 100%
- **性能测试**：ai 相关接口做到实时回传，使用并发立即回传更新状态，减少等待。
- **安全测试**：Token 认证机制有效，文件上传防护覆盖大部分恶意文件类型

# 8. 附件

附录 1：自动化测试报告（本地 html 文件，已经打包）

附件 2：前后端联调测试视频