1. 引言

1.1 项目背景

随着测试工作的深入开展，测试用例数量不断增加，人工执行效率低下且易出错。为提升测试效率，实现测试用例自动化执行、结果自动记录及缺陷自动跟踪，特开发本自动化测试平台。引入 AI 技术辅助测试流程智能化，进一步解决测试点遗漏、覆盖不全面等问题。

1.2 文档目的

明确自动化测试平台的功能需求、非功能需求及实现目标，为系统设计、开发和使用提供依据，重点包含 AI 增强功能的设计与应用场景。

1.3 定义与缩略语

测试用例：包含测试步骤、预期结果等信息的最小测试单元，可被平台自动执行。

Bug：测试用例执行失败时发现的软件缺陷。

公共操作：可被多个测试用例复用的标准化操作（如登录、退出等）。

AI 测试助手：集成大语言模型能力的模块，用于自动生成测试用例、分析测试覆盖度等。

2. 总体描述

2.1 产品前景

实现测试流程自动化，减少人工干预，通过 AI 技术提升测试用例覆盖率和精准度，提高测试效率和准确性，实现测试用例、执行结果、缺陷的一体化智能管理。

2.2 产品功能

项目与模块管理：创建和管理项目及下属模块。

测试用例管理：支持创建、编辑、参数化及批量 / 单个执行测试用例；新增 AI 自动生成、智能补全测试用例功能。

公共操作管理：提供专门页面管理可复用的公共操作，支持手动添加参数；新增 AI 推荐可复用操作组合功能。

测试执行：支持多环境、多浏览器执行，可配置最大并发任务数，按模块隔离账号避免资源竞争；新增 AI 预测高风险测试点并优先执行功能。

Bug 管理：测试用例执行失败自动创建 Bug，支持状态流转、手动修改严重程度及自定义模板，关闭时需填写原因；新增 AI 辅助分析 Bug 根因、推荐修复方案功能。

测试报告：生成并存储 HTML 格式的测试报告；新增 AI 智能分析报告，识别未覆盖测试点并推荐补充方案功能。

2.3 用户特征

仅支持单用户（管理员）使用，负责所有测试相关操作，可通过 AI 助手减少手动设计测试用例的工作量。

2.4 运行环境

服务器端：Windows 系统。

数据库：本地 Windows 安装的 MySQL。

客户端：通过浏览器在 Windows 上访问。

AI 服务依赖：可接入第三方大语言模型 API（如 GPT-4、文心一言等），或部署本地轻量化模型。

3. 具体需求

3.1 功能需求

3.1.1 项目与模块管理

支持项目的创建、编辑、删除。

每个项目可创建多个模块，模块用于划分测试用例。

3.1.2 测试用例管理

基础功能：支持创建、编辑、删除测试用例，用例包含富文本格式的步骤和预期结果，可上传图片；支持参数化测试用例，可配置多组输入参数；提供执行按钮，支持单个或勾选批量执行测试用例；用例代码存储在平台数据库中。

AI 增强功能：

自动生成测试用例：输入需求描述（如 “用户登录功能需支持手机号验证码登录”），AI 自动生成覆盖正常场景、异常场景（如手机号为空、验证码错误）的测试用例，包含步骤和预期结果。

智能补全测试点：基于已有测试用例，AI 分析功能逻辑，推荐遗漏的测试场景（如边界值、异常流），例如：当用户创建 “输入正确密码登录” 用例时，AI 推荐 “密码长度超限”“密码含特殊字符” 等补充用例。

参数化用例优化：AI 自动生成多样化参数组合，确保覆盖等价类划分（如不同长度、类型的输入值）。

3.1.3 公共操作管理

基础功能：提供专门页面管理公共操作，可创建、编辑、删除公共操作；公共操作支持手动添加参数，供测试用例调用。

AI 增强功能：

操作推荐：当用户创建新测试用例时，AI 分析用例步骤，推荐可复用的公共操作（如 “登录”“退出”），减少重复编写。

参数智能填充：基于公共操作的历史调用记录，AI 预测常用参数值（如默认用户名、环境地址），辅助快速配置。

3.1.4 测试执行

基础功能：支持选择开发环境或测试环境执行测试用例，默认使用 Chrome 浏览器；可配置最大同时执行的测试任务数；按模块为测试用例分配独立账号，避免资源竞争；执行过程中自动捕获截图和日志，存储在 Windows 本地。

AI 增强功能：

风险优先执行：AI 基于历史执行结果和用例优先级，预测高风险测试点（如最近修改的功能、历史 Bug 频发模块），优先执行相关用例，提前暴露问题。

执行异常智能中断：当连续多个用例因同一类错误失败时（如环境连接超时），AI 自动暂停执行并提示可能的根因（如 “测试环境服务器宕机”），减少无效执行。

3.1.5 Bug 管理

基础功能：测试用例执行失败时自动创建 Bug，标题格式为「[用例名称] 在 [环境名称] 下执行失败」，描述包含失败步骤、实际结果、错误日志等，自动关联测试用例、执行记录和环境信息，附加截图；同一用例在同一环境因相同错误（错误信息相似度≥80%）连续失败时，不重复创建 Bug，仅更新现有 Bug 的日志、截图和时间；Bug 状态流转：新建→已分配→处理中→已解决→已关闭，中间状态需留痕；测试用例再次执行成功，Bug 自动标记为已解决；支持手动修改 Bug 的严重程度，提供自定义 Bug 模板功能；手动关闭 Bug 时，需强制填写关闭原因；Bug 描述支持富文本格式，可上传图片。

AI 增强功能：

根因分析：AI 解析错误日志和测试步骤，自动判断 Bug 类型（如 “前端样式错误”“后端接口异常”），并推荐可能的原因（如 “接口返回字段缺失”）。

修复方案建议：基于同类 Bug 的历史修复记录，AI 推荐修复思路（如 “参考 Bug#123 的修复方式，补充参数校验”）。

3.1.6 测试报告

基础功能：测试执行完成后生成 HTML 格式的测试报告，包含测试结果、通过率等信息；测试报告存储在本地，支持查看和下载。

AI 增强功能：

覆盖度分析：AI 对比测试用例与需求文档，识别未覆盖的功能点（如 “需求中‘密码重置’功能未对应测试用例”），并推荐补充用例的设计方向。

趋势预测：分析多轮测试报告数据，AI 预测潜在风险（如 “支付模块通过率连续下降，建议重点排查”）。

3.2 非功能需求

系统响应时间合理，页面加载和操作响应不超过 3 秒；AI 功能（如用例生成、报告分析）单次响应不超过 10 秒。

数据存储安全，定期备份数据库和文件；AI 交互数据需脱敏处理（如去除日志中的敏感信息）。

AI 准确性要求：测试用例自动生成的准确率≥80%（人工审核通过比例）；覆盖度分析的遗漏识别率≥90%。