**武汉大学计算机学院**

**本科生实验报告**

**验收测试计划书**

专 业 名 称 ：软件工程

课 程 名 称 ：软件工程

指 导 教 师 ：何璐璐

团 队 成 员 一：莫韬

团 队 成 员 二：余嵘

团 队 成 员 三：周锦耀

团 队 成 员 四：王皓霖

团 队 成 员 五：蔡昀燃

团 队 成 员 六：张开宇

二○二四年四月

**郑 重 声 明**

本团队呈交的实验报告，是在指导老师的指导下，独立进行实

验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，

除文中已经注明引用的内容外，本实验报告不包含他人享有著作权

的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以

明确的方式标明。本实验报告的知识产权归属于培养单位。

团队成员签名：

日期：

**目 录**

**1 测试概述** ............................................... 5

**1.1 项目背景和目标** ..............................................5

**1.2 测试范围和重点** ..............................................5

**1.3 测试环境和系统依赖** ..........................................7

**2 测试计划** ................................................ 7

**2.1 测试周期和时间安排** ..........................................7

**2.2 测试人员分工和责任** ..........................................7

**2.3 测试流程和方法** ..............................................8

**2.4 测试用例设计原则** ............................................8

**3 功能测试** ................................................ 9

**3.1 用户注册和登录** ..............................................9

**3.2 诗词智能检索** ..................................................10

**3.3 每日推荐系统**................................................11

**3.4 诗词深度阅读**...............................................12

**3.5 智能创作评分**.................................................13

**3.6 诗词交互游戏**.................................................14

**3.7 教学辅助功能**.................................................16

**3.8 社区互动系统**...............................................18

**3.9 语音交互功能**.................................................20

**4 性能测试** .............................................. 26

**4.1 页面加载速度** ...............................................26

**4.2 并发用户访问** ...............................................27

**4.3 大文件上传** .................................................27

**4.4 编辑器响应时间** .............................................28

**5 兼容性测试** ............................................ 28

**5.1 主流浏览器兼容性** ...........................................28

**5.2 移动端设备适配** .............................................29

**5.3 屏幕分辨率适配** .............................................29

**1 测试概述**

**1.1 项目背景和目标**  
**项目背景**  
在信息化与传统文化深度融合的背景下，"云舟词渡"作为一款以诗词为核心的创新型软件系统，旨在通过技术手段解决传统诗词学习中的痛点：

1. **资源整合需求**：整合分散的诗词资源，构建统一的高效检索体系；
2. **场景化学习需求**：将诗词学习与课堂教学、个人休闲、创作实践结合，拓展应用场景；
3. **互动性提升需求**：通过社区交流、AI评分、语音识别等技术增强用户参与感；
4. **创作激励需求**：提供创作指导与多维评分机制，激发用户创作积极性。

**测试目标**  
本测试计划旨在验证"云舟词渡"系统的以下核心能力：

* **功能完整性**：覆盖诗词查询、创作评分、社区互动等核心功能模块；
* **性能可靠性**：支持高并发访问与快速响应，确保系统稳定性；
* **用户体验友好性**：界面交互流畅，操作符合用户直觉；
* **安全合规性**：保障用户数据安全，防范未授权访问与恶意攻击；
* **技术适配性**：兼容主流浏览器与移动设备，适配多种分辨率环境。

**1.2 测试范围和重点**  
本测试涵盖功能、性能、兼容性、易用性及安全性五大维度，具体范围及重点如下：

##### ****功能测试****

| **模块** | **测试内容** | **优先级** |
| --- | --- | --- |
| **用户管理** | 注册/登录流程、权限分级（普通用户/管理员）、第三方登录集成（微信/QQ） | 高 |
| **诗词检索** | 关键词搜索、分类筛选（朝代/作者/主题）、模糊匹配与纠错功能 | 高 |
| **内容展示** | 原文/译文/注释分层显示、多语言切换、字体/背景色自定义 | 中 |
| **AI评分系统** | NLP语义分析评分、韵律检测、与人工评分结果一致性验证 | 高 |
| **语音交互** | 朗诵音频采集质量、WebRTC延迟测试、评分算法准确性（音准/情感识别） | 中 |
| **社区功能** | 帖子发布/评论/点赞、敏感词过滤、用户积分体系与等级制度 | 高 |
| **诗词创作** | 在线编辑器稳定性、草稿自动保存、多格式导出（PDF/HTML） | 高 |
| **游戏化功能** | 飞花令匹配算法、答题计时容错机制、积分排行榜实时更新 | 中 |

##### ****性能测试****

1. **基准性能**：单用户操作响应时间；
2. **负载测试**：模拟1000并发用户访问社区及创作模块，CPU利用率≤80%；
3. **压力测试**：持续24小时高负载运行，检测内存泄漏及服务降级风险；
4. **容量测试**：Redis缓存命中率≥95%，MySQL索引优化效果验证。

##### ****兼容性测试****

1. **浏览器适配**：Chrome 100+、Firefox 90+、Safari 15+、Edge 95+；
2. **移动端适配**：iOS 14+/Android 10+主流机型（华为P50、iPhone 13等）；
3. **分辨率覆盖**：4K/2K/1080P屏幕显示优化，横竖屏切换无布局错位。

##### ****易用性测试****

1. **交互设计**：功能入口可见性评分≥4.5/5（基于Nielsen启发式评估）；
2. **学习成本**：新用户完成核心功能首次操作时间≤3分钟；
3. **错误恢复**：操作中断后数据恢复成功率100%，错误提示信息清晰度达标。

##### ****安全性测试****

1. **数据保护**：JWT令牌加密强度（HS512算法）、敏感信息传输加密（TLS 1.3）；
2. **漏洞防御**：SQL注入/XSS/CSRF攻击模拟，OWASP TOP 10漏洞覆盖率100%；
3. **权限管控**：越权访问检测（普通用户无法访问管理接口）。

**1.3 测试环境和系统依赖**

##### ****硬件环境****

| **组件** | **配置要求** | **用途** |
| --- | --- | --- |
| **服务器** | 4核CPU/16GB内存/500GB SSD | 后端服务与数据库部署 |
| **测试终端** | Intel i5/8GB内存/1920×1080分辨率 | 功能与兼容性测试 |
| **网络环境** | 千兆局域网/5G移动网络 | 多场景网络性能模拟 |

##### ****软件环境****

* **开发框架**：Vue 3.2.47（前端）、Spring Boot 3.0.6（后端）
* **数据库**：MySQL 8.0.32
* **容器化**：Docker 23.0.1、Nginx 1.23.4（反向代理）
* **AI服务**：Python 3.9+NLTK库（NLP评分）、TensorFlow Lite（语音识别）

##### ****系统依赖****

1. **第三方服务**：
   * 阿里云OSS
   * SMTP邮件服务
2. **测试工具链**：
   * **功能测试**：Selenium 4.8、Postman 10.14
   * **性能测试**：JMeter 5.5、Gatling 3.9
   * **安全测试**：Burp Suite 2023.3、OWASP ZAP 2.12
   * **兼容性测试**：BrowserStack、Xcode Simulator

# 2 测试计划

## 2.1 测试周期和时间安排

### 2.1.1 整体测试周期规划

本测试计划采用迭代式测试策略，总周期为**12周**，具体安排如下表所示：

| **测试阶段** | **时间跨度** | **主要工作内容** |
| --- | --- | --- |
| 单元测试 | 第1-3周 | 对诗词检索模块、用户认证模块、AI评分模块等核心组件进行代码级测试 |
| 集成测试 | 第4-5周 | 验证前后端接口交互，包括RESTful API集成测试、Redis缓存一致性测试 |
| 系统测试 | 第6-9周 | 执行功能测试全场景覆盖、性能压力测试、安全渗透测试 |
| 验收测试 | 第10周 | 用户代表参与UAT测试，重点验证诗词创作评分、社区交互等核心业务流程 |
| 回归测试 | 第11周 | 修复缺陷验证、跨版本功能回归 |
| 上线前预发布测试 | 第12周 | 生产环境模拟测试，包括Nginx负载均衡测试、Docker容器化部署验证 |

### 2.1.2 关键里程碑节点

1. **接口联调完成**（第5周末）：完成所有RESTful API的集成测试，接口响应达标率100%
2. **核心功能冻结**（第8周末）：诗词检索、创作评分、社区交互等主要功能通过全量测试用例
3. **性能基线达标**（第9周末）：系统支持1000并发用户访问，平均响应时间≤1.5秒
4. **安全审计通过**（第10周末）：通过OWASP Top 10安全测试标准，高危漏洞修复率100%

## 2.2 测试人员分工和责任

### 2.2.1 测试组织架构

测试总监（1人）

├── 功能测试组（3人）

├── 性能测试组（2人）

├── 安全测试组（2人）

└── 自动化测试组（2人）

### 2.2.2 详细职责矩阵

| **角色** | **主要职责** | **具体任务** |
| --- | --- | --- |
| 测试经理 | 制定测试策略、协调资源分配、监控测试进度 | 编制测试计划、组织缺陷评审会、输出测试总结报告 |
| 功能测试工程师 | 执行功能测试用例，验证诗词查询、创作评分等业务场景 | 编写测试用例、执行手工测试、记录测试结果 |
| 性能测试工程师 | 开展压力测试和性能调优 | 使用JMeter设计压测场景、分析TPS曲线、定位性能瓶颈 |
| 安全测试工程师 | 执行渗透测试和安全审计 | 实施SQL注入测试、XSS攻击模拟、JWT令牌安全性验证 |
| 自动化测试工程师 | 构建自动化测试框架 | 开发Selenium/Pytest测试脚本、维护CI/CD流水线 |
| 用户体验测试专员 | 评估界面交互和易用性 | 组织可用性测试、收集用户反馈、优化操作流程 |

## 2.3 测试流程和方法

### 2.3.1 标准化测试流程

A[需求分析] --> B[测试计划]

B --> C[用例设计]

C --> D[环境搭建]

D --> E[测试执行]

E --> F{缺陷处理}

F -->|是| G[回归测试]

F -->|否| H[测试报告]

**2.3.2 测试方法论**

1. **黑盒测试：**

边界值分析法验证诗词检索的查询条件边界（如标题长度限制、朝代枚举值）

等价类划分法测试AI评分系统的输入范围（押韵评分0-100分区间）

1. **白盒测试**

路径覆盖法检测诗词解析模块的条件分支

1. **自动化测试**：

#python伪代码

def test\_poetry\_search():

payload = {"keyword": "明月", "dynasty": "唐"}

response = requests.get(API\_URL, params=payload)

assert response.status\_code == 200

assert len(response.json()["data"]) > 0

## 2.4 测试用例设计原则

### 2.4.1 核心功能覆盖原则

1. **诗词查询模块**：
   * 支持组合查询条件（朝代+作者+关键词的三维筛选）
   * 验证特殊字符处理能力（测试包含「《》·」等符号的标题检索）
2. **创作评分系统**：
   * 设计NLP算法验证用例（平仄检测、对仗工整度、意象匹配度评分规则）
   * 构建边界测试数据集（包含0字/500字超长/包含敏感词的创作内容）
3. **社区交互功能**：
   * 设计并发评论测试场景（模拟100用户同时发表诗词点评）
   * 验证敏感词过滤机制（政治敏感词、广告信息的自动屏蔽）

### 2.4.2 非功能测试准则

**1.性能测试标准**：

* + - 诗词检索接口：95%请求响应时间≤800ms（99分位值≤1.2s）
  + - AI评分服务：单次评分耗时≤3s（GPU加速条件下）
  + - WebSocket推送：消息延迟≤500ms

**2.安全测试要求**：

* + 实施OWASP ZAP扫描，确保无Critical级别漏洞
  + JWT令牌需包含有效期（exp）、颁发者（iss）等标准声明
  + 敏感操作（如删除评论）必须二次认证

### 3 功能测试

**3.1 用户注册和登录**

测试用例1：诗人身份注册验证

步骤：

1. 访问注册页面选择"诗人"身份

2. 填写诗词创作偏好（擅长体裁、代表作品）

3. 提交有效注册信息

预期：

- 用户画像字段存入数据库

- 个性化推荐模块初始化

测试用例2：教育机构批量注册

步骤：

1. 使用学校edu邮箱注册

2. 上传教师资质证明文件

3. 申请10个学生子账号

预期：

- 自动开通班级管理功能

- 账号关联关系正确建立

**3.2 诗词智能检索**

测试用例1：多维度复合搜索

步骤：

1. 选择"宋词"朝代筛选

2. 输入"明月"关键词

3. 添加"婉约派"风格标签

预期：

- 返回李清照《一剪梅》等匹配结果

- 结果按相关度排序

测试用例2：语音输入搜索

步骤：

1. 点击麦克风图标

2. 朗诵"包含春雨的五言绝句"

3. 等待语音识别转化

预期：

- 正确识别口语化查询

- 显示杜甫《春夜喜雨》等结果

**3.3 每日推荐系统**

测试用例1：节日特推验证

步骤：

1. 在端午节当天登录

2. 查看每日推荐模块

预期：

- 首推屈原相关诗词

- 包含赛龙舟主题作品

测试用例2：学习进度跟踪

步骤：

1. 连续三天阅读边塞诗

2. 第四天查看推荐

预期：

- 推荐同类风格作品

- 出现"边塞诗专题"入口

#### 3.4 诗词深度阅读

测试用例1：多版本对照

步骤：

1. 打开《静夜思》阅读页

2. 切换"宋刻本"、"明刊本"版本

预期：

- 显示"床前看月光"异文

- 版本差异处高亮标注

测试用例2：名家朗诵播放

步骤：

1. 点击叶嘉莹朗诵版本

2. 调节播放速度为0.8倍

预期：

- 音频缓冲时间＜1s

- 播放进度条可拖动

**3.5 智能创作评分**

测试用例1：格律自动检测

步骤：

1. 输入不符合《鹧鸪天》词牌的作品

2. 提交AI评分

预期：

- 平仄错误位置标红

- 押韵偏差数值提示

测试用例2：跨平台评分对比

步骤：

1. 同一作品提交系统评分

2. 同步分享至诗词中国社区

预期：

- AI评分与人工评分差异＜15%

- 评价维度雷达图可视化

**3.6 诗词交互游戏**

测试用例1：飞花令多人对战

步骤：

1. 创建"春"字房间

2. 3名玩家轮流接龙

3. 输入非诗句内容

预期：

- 语音识别过滤无效答案

- 自动触发"锦囊"提示

测试用例2：AR对对联

步骤：

1. 扫描实体书桌

2. 通过手势放置虚拟下联

3. 调整字体风格为瘦金体

预期：

- 对仗工整度实时评分

- AR场景中保持文字透视正确

**3.7 教学辅助功能**

测试用例1：课堂进度同步

步骤：

1. 教师端发布《滕王阁序》任务

2. 学生端完成注释批注

3. 教师查看完成情况

预期：

- 批注数据实时同步

- 未完成学生名单标红

测试用例2：诗词地图联动

步骤：

1. 在南昌打开定位功能

2. 点击滕王阁地标

3. 选择"情景模式"

预期：

- 自动播放《滕王阁序》VR场景

- 出现方言诵读可选版本

**3.8 社区互动系统**

测试用例1：诗评智能过滤

步骤：

1. 发布包含网络用语的评论

2. 提交含敏感词的内容

预期：

- 自动转换为文言表述

- 违规内容进入审核队列

测试用例2：作品版权存证

步骤：

1. 完成原创律诗创作

2. 点击"区块链存证"

3. 支付数字版权费用

预期：

- 生成不可篡改的存证证书

- 在版权局官网可查验

**3.9 语音交互功能**

测试用例1：方言朗诵适配

步骤：

1. 选择粤语朗诵模式

2. 朗读《念奴娇·赤壁怀古》

3. 开启实时注音功能

预期：

- 生成粤语拼音对照

- 情感分析得分＞85

测试用例2：多人联句创作

步骤：

1. 发起语音联句房间

2. 三位用户轮流续句

3. 中途网络波动断线

预期：

- 自动保存临时作品

- 断线用户可重连续作

**4.1 页面加载速度**

**测试目标**  
验证用户访问平台时页面加载的响应速度，确保在多种网络条件下均能提供流畅体验。

**测试用例与步骤**

1. **单个页面加载速度**
   * **步骤**
     1. 使用浏览器开发者工具记录主页面、编辑页面、导出页面的加载时间。
     2. 模拟不同网络条件（如高速网络、3G网络、高延迟网络）下的加载情况。
   * **预期结果**  
     页面完全加载时间≤3秒（高速网络）且无明显卡顿。
2. **首次页面加载速度**
   * **步骤**
     1. 清除浏览器缓存后首次访问平台，记录从请求到首屏内容渲染完成的时间。
     2. 在移动端（iOS/Android）重复测试并对比结果。
   * **预期结果**  
     首屏渲染时间≤2秒，跨设备加载性能一致。
3. **静态资源加载速度**
   * **步骤**
     1. 监测CSS、JavaScript、图片等静态资源加载耗时。
     2. 验证CDN加速是否生效，优化资源压缩策略。
   * **预期结果**  
     静态资源加载无阻塞，总耗时≤1.5秒。

**4.2 并发用户访问**

**测试目标**  
评估系统在高并发场景下的稳定性和资源利用率，确保多用户同时操作时性能达标。

**测试用例与步骤**

1. **并发用户负载测试**
   * **步骤**
     1. 使用JMeter模拟100+用户同时登录、编辑画布、上传文件。
     2. 监控服务器CPU、内存、网络带宽占用率。
   * **预期结果**
     1. 响应时间≤500ms（轻载）、≤2秒（峰值负载）。
     2. 资源利用率≤80%，无宕机或数据丢失。
2. **用户请求并发处理**
   * **步骤**
     1. 模拟多用户同时执行“保存文件”操作，观察事务处理成功率。
   * **预期结果**  
     所有请求均被正确处理，无超时或冲突。

**4.3 大文件上传**

**测试目标**  
验证系统对大型设计文件（如高清图片、复杂模板）的处理能力。

**测试用例与步骤**

1. **单个大文件上传速度**
   * **步骤**
     1. 上传500MB以上的PSD或AI格式文件，记录传输时间和进度反馈。
   * **预期结果**
     1. 上传进度条实时更新，无中断。
     2. 平均上传速率≥5MB/s（高速网络）。
2. **并发大文件上传**
   * **步骤**
     1. 模拟10个用户同时上传1GB文件。
     2. 监测服务器磁盘I/O和网络队列情况。
   * **预期结果**
     1. 所有文件完整上传，无数据损坏。
     2. 服务器负载平稳，无资源耗尽现象。

**4.4 编辑器响应时间**

**测试目标**  
确保核心设计功能（如元素编辑、实时渲染）的交互流畅性。

**测试用例与步骤**

1. **文本输入响应时间**
   * **步骤**
     1. 在画布中输入多段文字（含富文本格式），观察输入延迟。
   * **预期结果**  
     输入反馈≤100ms，无卡顿或丢字。
2. **样式调整响应时间**
   * **步骤**
     1. 对图形进行缩放、旋转、颜色填充操作，记录操作到渲染完成的耗时。
   * **预期结果**  
     渲染时间≤300ms，视觉更新即时可见。
3. **撤销/重做操作响应**
   * **步骤**
     1. 连续执行10次撤销操作，验证历史记录栈的响应速度。
   * **预期结果**  
     每次操作响应≤200ms，状态回滚准确。

**5.1 主流浏览器兼容性**

**测试用例1：浏览器兼容性测试**  
步骤：

1. 在不同的主流浏览器（如 Chrome、Firefox、Safari、Edge 等）上访问应用程序，确认页面正常加载、显示和操作。
2. 检查功能、布局、样式在不同浏览器上是否一致。

**测试用例2：浏览器版本兼容性测试**  
步骤：

1. 测试应用程序在主流浏览器的不同版本（包括新版本和旧版本）上的兼容性。
2. 验证功能、布局、样式在不同版本浏览器中是否一致。

**5.2 移动端设备适配**

**测试用例1：移动设备兼容性测试**  
步骤：

1. 使用多种移动设备（如 Android 手机、iOS 平板等）访问应用程序。
2. 验证页面在移动端是否正常显示和操作，避免布局错乱或功能异常。

**测试用例2：不同屏幕尺寸适配测试**  
步骤：

1. 在不同屏幕尺寸的移动设备（小屏手机、大屏平板等）上测试页面适配性。
2. 检查布局、字体大小、按钮等元素是否适配，避免内容溢出或操作困难。

**5.3 屏幕分辨率适配**

**测试用例1：不同分辨率适配测试**  
步骤：

1. 在低分辨率和高分辨率屏幕上测试页面显示效果。
2. 验证布局、图像清晰度、字体显示是否正常，避免模糊或错位问题。

**测试用例2：缩放功能测试**  
步骤：

1. 在支持缩放的设备上测试页面缩放行为。
2. 检查不同缩放级别下布局和内容的适配性，确保用户体验一致。