测试总结报告模板

文档的历史记录

版本号	日期	作者	更改/更改原因/提议更改	评审记录	备注
V1	2025/04/13	21 组	初始文档	通过	

目录

测试总结报告

1 概述

1.1 测试范围

计划范围：
Service 层：com.demo.service.impl 包下的 UserService (用户登录)、OrderService (订单的创建、查询（ID）、状态变更、用户订单查询（分页）、删除、更新)、MessageService (留言的查询（ID）、状态变更、用户留言查询（分页）、待审核查询（分页）、已通过查询（分页）)、VenueService (场地信息的查询（ID、名称、所有-分页/不分页)、创建、更新、删除(ID)、名称统计)、NewsService (新闻的查询(所有-分页、ID)、创建、删除(ID)、更新)、OrderVoService (Order 到 OrderVo 的转换)、MessageVoService (Message 到 MessageVo 的转换) 中的所有公共方法。
Controller 层：com.demo.controller 包下的 admin 和 user 子包中的所有 Controller 类及其暴露的 API 端点，以及 IndexController 中的相关接口。

实际范围： 所有计划的 Service 方法和 Controller 接口都进行了测试。

1.2 测试进度

计划进度表：

任务	时间
单元测试设计与脚本开发	2025-04-01 至 2025-04-07
集成测试设计与脚本开发	2025-04-01 至 2025-04-08
测试执行(单元+集成)	2025-04-09

实际进度表：

任务	时间
单元测试设计与脚本开发	2025-04-01 至 2025-04-07
集成测试设计与脚本开发	2025-04-01 至 2025-04-08
测试执行(单元+集成)	2025-04-09

1.3 测试方法策略

计划测试策略方法： 采用白盒测试（单元测试）和灰盒/黑盒测试（集成测试）相结合的方法。利用 JUnit 5 测试框架执行测试。单元测试针对方法逻辑、分支、边界值、异常情况设计测试用例。集成测试使用 JUnit 5 和 Spring Boot Test，模拟 HTTP 请求并验证响应。测试用例设计采用等价类划分、边界值分析、错误推测等方法。

实际测试策略方法： 与计划一致

1.4 测试风险

计划风险
开发交付延期或代码质量低下，影响测试进度
需求理解不充分或需求频繁变更，导致测试用例失效
测试人员技能不足或资源短缺
测试覆盖率不足，可能遗漏缺陷
难以获取或构造有效的测试数据

实际风险
开发交付延期，影响测试进度
测试覆盖率不足，可能遗漏缺陷
难以获取或构造有效的测试数据

风险分析：
针对延期问题，定期进行进度跟踪，发现延期风险及时调整资源和计划；
针对测试数据和覆盖率问题，采用多种测试用例设计方法，通过分析数据特点和规律，手工构造有效测试数据

1.5 测试结果总结

出口准则：
所有计划执行的单元测试和集成测试用例均已执行。
所有测试用例执行结果为 "Pass"。
发现的所有严重 (Critical) 和高 (High) 优先级的缺陷都已修复并经验证关闭。
没有影响核心功能或导致系统崩溃的未解决缺陷。

单元测试总结：
测试执行情况总结:
总测试用例数: 128 个
通过: 112 个
失败: 16 个
错误: 0 个
跳过: 0 个
通过率: 87.5% (112/128)
发现的缺陷:
共发现 16 个缺陷，按优先级分类：高优先级缺陷 (5 个):SQL 注入漏洞：未能过滤 SQL 关键字和特殊字符
高危 XSS 漏洞：未能过滤 iframe、javascript 或远程脚本加载代码
XSS 脚本漏洞：未能过滤 XSS 脚本标签
整数溢出漏洞：在计算大数值时出现整数溢出
明文密码存储：系统将密码以明文形式存储，应使用密码哈希算法
中优先级缺陷 (8 个):空值处理不当：系统应主动检查参数非 null 并抛出相应异常
状态转换错误：系统不应允许将已完成(STATE_FINISH)的订单状态变更为已拒绝(STATE_REJECT)
异常处理不当：系统应该捕获懒加载异常并转换为友好的业务异常

业务规则验证不足：
系统不应接受营业时间外的订单预订
系统不应接受过去时间的订单预订
系统不应接受负数小时数或计算负数总价
数据验证不足：
系统应该拒绝包含特殊字符的用户 ID
系统应该检查并拒绝创建重复的 userID
低优先级缺陷 (3 个):弱密码策略：系统应该拒绝弱密码(123456)，但接受了它
缺少空对象检查：系统应主动检查并抛出带有明确错误消息的异常
服务测试覆盖情况:
MessageService: 31 个测试，全部通过
MessageVoService:3 个测试，全部通过
NewsService: 35 个测试，5 个失败
OrderService: 12 个测试，7 个失败
OrderVoService:3 个测试，全部通过
UserService:11 个测试，4 个失败
VenueService: 32 个测试，全部通过

集成测试总结：
测试执行情况总结:
总测试用例数: 113 个
通过: 92 个
失败: 0 个
错误: 21 个
跳过: 0 个
通过率: 81.42% (92/113)

发现的缺陷:
共发现 21 个缺陷，按优先级分类：
高优先级缺陷 (8 个):
SQL 注入漏洞 (MessageControllerTest.testsqlInjectionVulnerability:443, VenueControllerTest.testsqlInjectionVulnerabilityInVenueId:193): 系统未能过滤 SQL 关键字和特殊字符，存在 SQL 注入风险。
高危 XSS 漏洞 (MessageControllerTest.testXssvulnerability:410): 系统未能过滤 javascript 脚本代码，存在高危 XSS 漏洞。
整数溢出漏洞 (OrderControllerTest.testIntegerverflowvulnerability:414): 允许使用 MAX_VALUE 作为小时数，可能导致价格计算错误。
明文密码存储 (UserControllerTest.testPasswordstoragesecurity:501, UserControllerTest.testPlaintextPasswordstored:302): 系统将密码以明文形式存储，应使用密码哈希算法。
允许用户修改他人留言漏洞 (MessageControllerTest.testModifyothersMessageVulnerability:374): 系统存在安全漏洞，允许用户修改他人的留言。
允许用户访问他人的订单漏洞 (OrderControllerTest.testordersecurityvulnerability:556): 系统存在安全漏洞，

允许用户访问他人的订单。

中优先级缺陷 (9 个):
空指针异常，未正确处理未登录状态 (MessageControllerTest.testMessageListPagewithoutLoginUser:158): 系统存在空指针异常，未正确处理未登录状态。
不应接受负数小时数 (OrderControllerTest.testNegativeValueVulnerability:450): 系统不应接受负数小时数。
缺少空对象检查 (OrderControllerTest.testNullVenueOrder:578): 系统应主动检查并抛出带有明确错误信息的异常。
不应接受营业时间外的订单预订 (OrderControllerTest.testoutsideBusinessHoursVulnerability:485): 系统不应接受营业时间外的订单预订 (2025-06-01T23:00)。
不应接受过去时间的订单预订 (OrderControllerTest.testPastTimeVulnerability:528): 系统不应接受过去时间的订单预订 (2025-04-12T16:48)。
应该拒绝包含特殊字符的用户 ID (UserControllerTest.testInvalidUserIdHandling:374): 系统应该拒绝包含特殊字符的用户 ID。
未正确验证场馆 ID (VenueControllerTest.testsqlInjectionVulnerabilityInVenueId:193): 系统没有正确验证场馆 ID，可能存在 SQL 注入风险。

低优先级缺陷 (4 个):
接受了简单弱密码 (UserControllerTest.testWeakPasswordAccepted:337): 系统接受了简单弱密码。

测试模块覆盖情况:
MessageController: 4 个测试错误 (MessageControllerTest.testMessageListPagewithoutLoginUser, MessageControllerTest.testModifyothersMessageVulnerability, MessageControllerTest.testsqlInjectionVulnerability, MessageControllerTest.testXssvulnerability)
OrderController: 6 个测试错误 (OrderControllerTest.testIntegerverflowvulnerability, OrderControllerTest.testNegativeValueVulnerability, OrderControllerTest.testNullVenueOrder, OrderControllerTest.testordersecurityvulnerability, OrderControllerTest.testoutsideBusinessHoursVulnerability, OrderControllerTest.testPastTimeVulnerability)
UserController: 4 个测试错误 (UserControllerTest.testInvalidUserIdHandling, UserControllerTest.testPasswordstoragesecurity, UserControllerTest.testPlaintextPasswordstored, UserControllerTest.testWeakPasswordAccepted)
VenueController: 1 个测试错误 (VenueControllerTest.testsqlInjectionVulnerabilityInVenueId)

2 单元测试

2.1 单元测试进度表

负责人：韩创意

时间	进度
2025-04-01 至 2025-04-02	UserService 测试用例

2025-04-02 至 2025-04-04	OrderService 测试用例
2025-04-04 至 2025-04-05	MessageService 测试用例
2025-04-05 至 2025-04-07	MessageVoService 测试用例

负责人：侯一夫

时间	进度
2025-04-01 至 2025-04-03	VenueService 测试用例
2025-04-03 至 2025-04-05	NewsService 测试用例
2025-04-05 至 2025-04-07	OrderVoService 测试用例

2.2 单元测试方法汇总

1. 黑盒测试：基于等价类划分和边界值分析
2. 白盒测试：基于语句覆盖和判定覆盖
3. 单元测试：使用 JUnit 5 和 Mockito 框架进行隔离测试

2.3 单元测试结果

范围：
覆盖项目中所有关键的 Service 实现类： - UserServiceImpl - 用户服务 - OrderServiceImpl - 订单服务 - MessageServiceImpl - 留言服务 - VenueServiceImpl - 场馆服务 - NewsServiceImpl - 新闻服务 - OrderVoServiceImpl - 订单视图对象服务 - MessageVoServiceImpl - 留言视图对象服务

发现的缺陷：
订单管理系统问题:
没有检测订单时间冲突 ("应抛出 RuntimeException，提示'所选时间段已被预订'")
允许对已完成订单进行状态变更 ("应该抛出 IllegalStateException 异常，消息为'不能拒绝已完成的订单'")
接受营业时间外的预订 ("应抛出 IllegalArgumentException，提示'预订时间必须在场馆营业时间内(9:00-22:00)'")
接受过去时间的预订 ("应抛出 IllegalArgumentException，提示'预订时间不能是过去时间'")
用户管理系统问题:
密码以明文形式存储，没有使用哈希算法 ("应使用 BCrypt 或其他哈希算法处理密码，而不是存储明文")
接受包含特殊字符的用户 ID ("应抛出 IllegalArgumentException，提示'用户 ID 只能包含字母、数字和下划线'")

数据验证问题:
缺乏输入验证: 系统接受负小时数、零小时数和过大的小时数值
缺乏边界检查: 系统未能检测并拒绝无效的订单 ID、过去的时间等
缺乏状态转换验证: 系统允许无效的状态转换，如拒绝已完成的订单

异常处理问题:
未处理的空指针异常: 多个方法在接收 null 参数时会抛出未处理的 NullPointerException
异常传播: 所有 DAO 层的异常都直接传播到表示层，没有适当的转换或处理

3 集成测试

3.1 集成测试进度表

负责人：王驭风

时间	进度
2025-04-01 至 2025-04-03	AdminMessageController 测试用例
2025-04-01 至 2025-04-03	MessageController 测试用例
2025-04-03 至 2025-04-04	AdminVenueController 测试用例
2025-04-04 至 2025-04-05	IndexController 测试用例
2025-04-04 至 2025-04-08	AdminNewsController 测试用例
2025-04-04 至 2025-04-08	AdminUserController 测试用例

负责人：傅文杰

时间	进度
2025-04-01 至 2025-04-04	VenueController 测试用例
2025-04-01 至 2025-04-04	NewsController 测试用例
2025-04-01 至 2025-04-04	UserController 测试用例
2025-04-04 至 2025-04-06	OrderController 测试用例
2025-04-06 至 2025-04-08	AdminOrderController 测试用例

3.2 集成测试方法汇总

1. 黑盒测试：基于等价类划分和边界值分析
2. 白盒测试：基于语句覆盖和判定覆盖
3. 集成测试：使用 JUnit 5 和 Mockito 框架进行测试

3.3 集成测试结果

范围：覆盖项目中所有关键的 Controller 类： - MessageController - 留言控制器测试 - OrderController - 订单控制器测试 - UserController - 用户控制器测试 - VenueController - 场馆控制器测试等等

发现的缺陷：
留言管理系统：
允许空留言提交 ("应在提交时校验，拒绝空留言")
未处理未登录状态空指针异常 ("应检查登录状态，避免空指针异常")
存在修改他人留言安全漏洞 ("应进行权限校验，禁止修改他人留言")
有 SQL 注入、XSS 漏洞 ("应过滤 SQL 关键字、特殊字符及 javascript 脚本代码")

订单管理系统：
存在整数溢出风险（允许 MAX_VALUE 作小时数） ("应限制小时数取值，避免溢出")
接受负小时数 ("应校验输入，拒绝负数小时数")

存在访问他人订单安全漏洞 ("应校验权限，只允许访问本人订单")
接受营业外及过去时间订单 ("应校验时间，拒绝营业外及过去时间订单")
用户管理系统接受特殊字符用户 ID ("应校验 ID，拒绝含特殊字符的输入")
密码明文存储 ("应使用哈希算法加密存储密码")
接受简单弱密码 ("应校验密码强度，拒绝简单弱密码")

场馆管理系统：
场馆 ID 未正确验证，有 SQL 注入风险 ("应过滤 SQL 关键字，正确验证场馆 ID")
数据验证缺乏输入验证 ("应拒绝不合理数值输入，如负小时数等")
缺乏边界检查 ("应检测并拒绝无效订单 ID、过去时间等")
缺乏状态转换验证 ("应禁止无效状态转换，如修改他人留言等")

异常处理：
空指针异常未处理 ("应处理接收 null 参数时的空指针异常")
安全漏洞防范不足 ("应加强 SQL 注入、XSS 等安全漏洞防范")

4 经验总结

好的经验：
1. 测试方法结合有效：在本次测试中，采用白盒测试（单元测试）和灰盒 / 黑盒测试（集成测试）相结合的方式，搭配 JUnit 5 测试框架执行测试，同时运用等价类划分、边界值分析、错误推测等测试用例设计方法。多种测试方法的综合运用，使得测试全面且深入，成功发现订单管理系统、用户管理系统等模块的诸多缺陷，如订单时间冲突、密码明文存储等问题，有效保障了软件质量 。
2. 任务分配明确高效：单元测试阶段将任务合理分配给不同负责人，明确各时间段需完成的服务测试用例。这种明确的分工使团队成员各司其职，专注于特定模块，减少任务交叉干扰，提高了测试用例设计与开发的效率，确保单元测试阶段按时完成

需要改进的点：
1. 测试进度把控能力有待加强：测试执行阶段出现延迟，尽管原因是深入检测错误、完善用例，但仍反映出在进度管理方面存在不足，未充分考虑可能出现的问题并预留足够缓冲时间。
改进分析：制定测试计划时，对任务难度和潜在风险评估不够准确，缺乏科学的进度监控机制。在遇到问题时，调整措施不够及时有效，导致进度延迟。
改进建议：采用更科学的项目管理方法，如使用甘特图细化测试任务，明确各任务的依赖关系和时间节点，同时运用三点估算法评估任务时间，预留合理缓冲期。建立每日进度汇报和检查机制，及时发现进度偏差，针对问题快速调整资源和计划，确保测试工作按计划推进。

通过本次实验，我们深刻体会到单元测试的重要性，以及在测试过程中深入分析和挖掘潜在问题的必要性。同时，我们也认识到在保证测试质量的前提下，提高测试效率和合理规划测试时间的重要性。未来的实验和项目中，我们将积极采取以上改进建议，不断提升我们的测试能力和项目交付效率。