

测试用例设计文档

一、单元测试用例

UserService 测试用例

测试对象	src.main.java.com.demo.service.impl.UserServiceImpl.java.checkLogin(String userID, String password)
测试函数	test.main.java.com.demo.service.impl.UserServiceImplTest.java

用例编号	用例描述	预期结果	测试结果	结论
UC-U01	用户 ID 存在且密码正确	通过	通过	正确
UC-U02	用户 ID 不存在	不通过	不通过	正确
UC-U03	用户 ID 存在但密码错误	不通过	不通过	正确
UC-U04	用户 ID 为 null	不通过	不通过	正确
UC-U05	密码为 null	不通过	不通过	正确
UC-U06	用户 ID 为空字符串	不通过	不通过	正确
UC-U07	密码为空字符串	不通过	不通过	正确
UC-U08	管理员用户 ID 和正确密码	通过	通过	正确
UC-U09	创建用户 - 使用已存在的 userID	拒绝创建，提示“用户 ID 已存在”	系统未做重复检查，允许重复创建	错误
UC-U10	创建用户 - userID 包含非法字符（如@#\$\$%）	拒绝创建，提示非法 userID 格式	系统接受了非法字符 userID	错误

用例编号	用例描述	预期结果	测试结果	结论
UC-U11	创建用户 - 密码未加密存储	密码使用哈希算法加密存储，不应为明文	系统将密码以明文方式存储	错误

OrderService 测试用例

findById 方法

测试对象	src.main.java.com.demo.service.impl.OrderServiceImpl.java.findById(int OrderID)			
测试函数	test.main.java.com.demo.service.impl.OrderServiceImplTest.java.findById_OrderExists()			

用例编号	用例描述	预期结果	测试结果	结论
UC-001	查询订单-订单存在	返回订单对象	返回订单对象	正确

submit 方法

测试对象	src.main.java.com.demo.service.impl.OrderServiceImpl.java.submit(String venueName, LocalDateTime startTime, int hours, String userID)			
测试函数	test.main.java.com.demo.service.impl.OrderServiceImplTest.java.submit_NormalCase()			

用例编号	用例描述	预期结果	测试结果	结论
UC-002	提交订单-正常情况	成功创建订单	成功创建订单	正确
UC-003	提交订单-整型溢出小时数（2147483647）	拒绝请求或价格异常校验错误提示	价格计算为负值，发生整数溢出	错误
UC-004	提交订单-场馆为 null	抛出明确异常（如 IllegalArgumentException）	抛出 NullPointerException	错误

用例编号	用例描述	预期结果	测试结果	结论
UC-005	提交订单-预约时间为非营业时间（23:00）	拒绝订单创建	系统接受订单	错误
UC-006	提交订单-预约时间在过去	拒绝订单创建	系统接受订单	错误

confirmOrder 方法

测试对象

src.main.java.com.demo.service.impl.OrderServiceImpl.java.confirmOrder(int orderID)

测试函数

test.main.java.com.demo.service.impl.OrderServiceImplTest.java.confirmOrder_OrderExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-007	确认订单-订单存在	状态更新为等待	状态更新为等待	正确
UC-008	确认订单-订单不存在	抛出异常	抛出异常	正确

finishOrder 方法

测试对象

src.main.java.com.demo.service.impl.OrderServiceImpl.java.finishOrder(int orderID)

测试函数

test.main.java.com.demo.service.impl.OrderServiceImplTest.java.finishOrder_OrderExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-009	完成订单-订单存在	状态更新为完成	状态更新为完成	正确
UC-010	完成订单-订单不存在	抛出异常	抛出异常	正确

rejectOrder 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.OrderServiceImpl.java.rejectOrder(int orderID)
test.main.java.com.demo.service.impl.OrderServiceImplTest.java.rejectOrder_OrderExi
sts()

用例编号	用例描述	预期结果	测试结果	结论
UC-011	拒绝订单-订单存在	状态更新为拒绝	状态更新为拒绝	正确
UC-012	拒绝订单-订单不存在	抛出异常	抛出异常	正确

findUserOrder_Pagination 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.OrderServiceImpl.java.findUserOrder(String
userID, Pageable pageable)
test.main.java.com.demo.service.impl.OrderServiceImplTest.java.findUserOrder_Pagin
ation()

用例编号	用例描述	预期结果	测试结果	结论
UC-0013	查询用户订单-分页查询	返回用户订单分页列表	返回用户订单分页列表	正确

delOrder 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.OrderServiceImpl.java.delOrder(int
orderID)
test.main.java.com.demo.service.impl.OrderServiceImplTest.java.delOrder()

用例编号	用例描述	预期结果	测试结果	结论
UC-014	删除订单	订单被删除	订单被删除	正确

updateOrder 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.OrderServiceImpl.java.updateOrder(int orderID, String venueName, LocalDateTime startTime, int hours, String userID)

test.main.java.com.demo.service.impl.OrderServiceImplTest.java.updateOrder()

用例编号	用例描述	预期结果	测试结果	结论
UC-015	更新订单	订单被更新	订单被更新	正确
UC-016	更新订单-小时数为负（-5）	拒绝更新，应提示非法参数	接受请求，计算得出负价格	错误
UC-017	更新订单-订单不存在	抛出明确异常提示“订单不存在”	抛出 NullPointerException	错误

MessageService 测试用例

findById 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.MessageServiceImpl.java.findById(int messageID)

test.main.java.com.demo.service.impl.MessageServiceImplTest.java.findById_MessageExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-M01	根据 ID 查找留言-留言存在	返回留言对象	返回留言对象	正确

confirmMessage 方法

测试对象

src.main.java.com.demo.service.impl.MessageServiceImpl.java.confirmMessage(int messageID)

测试对象
测试函数

src.main.java.com.demo.service.impl.MessageServiceImpl.java.confirmMessage(int
messageID)

test.main.java.com.demo.service.impl.MessageServiceImplTest.java.confirmMessage_M
essageExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-M05	通过留言审核-留言存在	状态更新为通过	状态更新为通过	正确
UC-M06	通过留言审核-留言不存在	抛出异常	抛出异常	正确

rejectMessage 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.MessageServiceImpl.java.rejectMessage(int
messageID)

test.main.java.com.demo.service.impl.MessageServiceImplTest.java.rejectMessage_Mes
sageExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-M07	拒绝留言-留言存在	状态更新为拒绝	状态更新为拒绝	正确
UC-M08	拒绝留言-留言不存在	抛出异常	抛出异常	正确

findByUser_Pagination 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.MessageServiceImpl.java.findByUser(String
userID, Pageable pageable)

test.main.java.com.demo.service.impl.MessageServiceImplTest.java.findByUser_Pagina
tion()

用例编号	用例描述	预期结果	测试结果	结论
UC-M09	查询用户留言-分页查询	返回用户留言分页列表	返回用户留言分页列表	正确

findWaitState_Pagination 方法

测

试

对

象

src.main.java.com.demo.service.impl.MessageServiceImpl.java.findWaitState(Pageable pageable)

测

试

函

数

test.main.java.com.demo.service.impl.MessageServiceImplTest.java.findWaitState_Pagination()

用例编号	用例描述	预期结果	测试结果	结论
UC-M10	查询待审核留言-分页查询	返回待审核留言分页列表	返回待审核留言分页列表	正确

findPassState_Pagination 方法

测

试

对

象

src.main.java.com.demo.service.impl.MessageServiceImpl.java.findPassState(Pageable pageable)

测

试

函

数

test.main.java.com.demo.service.impl.MessageServiceImplTest.java.findPassState_Pagination()

用例编号	用例描述	预期结果	测试结果	结论
UC-M11	查询已通过留言-分页查询	返回已通过留言分页列表	返回已通过留言分页列表	正确

VenueService 测试用例

findByVenueID 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.VenueServiceImpl.java.findByVenueID(int id)
test.main.java.com.demo.service.impl.VenueServiceImplTest.java.findByVenueID_VenueExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-V01	根据 ID 查找场馆-场馆存在	返回场馆对象	返回场馆对象	正确

findByVenueName 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.VenueServiceImpl.java.findByVenueName(String venueName)
test.main.java.com.demo.service.impl.VenueServiceImplTest.java.findByVenueName_VenueExists()

用例编号	用例描述	预期结果	测试结果	结论
UC-V02	根据名称查找场馆-场馆存在	返回场馆对象	返回场馆对象	正确

findAll_Pagination 方法

测试对象
测试函数

src.main.java.com.demo.service.impl.VenueServiceImpl.java.findAll(Pageable pageable)
test.main.java.com.demo.service.impl.VenueServiceImplTest.java.findAll_Pagination()

用例编号	用例描述	预期结果	测试结果	结论
UC-V03	查询所有场馆-分页	返回场馆分页列表	返回场馆分页列表	正确

findAll_NoPage 方法

测试

对象 `src.main.java.com.demo.service.impl.VenueServiceImpl.java.findAll()`

测试函数 `test.main.java.com.demo.service.impl.VenueServiceImplTest.java.findAll_NoPage()`

用例编号	用例描述	预期结果	测试结果	结论
UC-V04	查询所有场馆-不分页	返回场馆列表	返回场馆列表	正确

create_Success 方法

测试对

象 `src.main.java.com.demo.service.impl.VenueServiceImpl.java.create(Venue venue)`

测试函数 `test.main.java.com.demo.service.impl.VenueServiceImplTest.java.create_Success()`

用例编号	用例描述	预期结果	测试结果	结论
UC-V05	创建场馆	返回场馆 ID	返回场馆 ID	正确

update_Success 方法

测试

对象 `src.main.java.com.demo.service.impl.VenueServiceImpl.java.update(Venue venue)`

测试函数 `test.main.java.com.demo.service.impl.VenueServiceImplTest.java.update_Success()`

用例编号	用例描述	预期结果	测试结果	结论
UC-V06	更新场馆	场馆被更新	场馆被更新	正确

delById_Success 方法

测试

对象 `src.main.java.com.demo.service.impl.VenueServiceImpl.java.delById(int id)`

测试函数 `test.main.java.com.demo.service.impl.VenueServiceImplTest.java.delById_Success()`

用例编号	用例描述	预期结果	测试结果	结论
UC-V07	删除场馆	场馆被删除	场馆被删除	正确

countVenueName 方法

测试对象	src.main.java.com.demo.service.impl.VenueServiceImpl.java.countVenueName(String venueName)			
测试函数	test.main.java.com.demo.service.impl.VenueServiceImplTest.java.countVenueName()			

用例编号	用例描述	预期结果	测试结果	结论
UC-V08	统计场馆名称数量	返回名称匹配的场馆数量	返回名称匹配的场馆数量	正确

NewsService 测试用例

findById 方法

测试对象	src.main.java.com.demo.service.impl.NewsServiceImpl.java.findById(int newsID)			
测试函数	test.main.java.com.demo.service.impl.NewsServiceImplTest.java.findById_NewsExists()			

用例编号	用例描述	预期结果	测试结果	结论
UC-N01	根据 ID 查找新闻-新闻存在	返回新闻对象	返回新闻对象	正确
UC-N02	根据 ID 查找新闻-新闻不存在	抛出业务异常或提示信息	抛出底层异常，未转换为提示	错误

findAll_Pagination 方法

测试对象	src.main.java.com.demo.service.impl.NewsServiceImpl.java.findAll(Pageable pageable)
测试函数	test.main.java.com.demo.service.impl.NewsServiceImplTest.java.findAll_Pagination()

用例编号	用例描述	预期结果	测试结果	结论
UC-N03	查询所有新闻-分页	返回新闻分页列表	返回新闻分页列表	正确

create_Success 方法

测试对象	src.main.java.com.demo.service.impl.NewsServiceImpl.java.create(News news)
测试函数	test.main.java.com.demo.service.impl.NewsServiceImplTest.java.create_Success()

用例编号	用例描述	预期结果	测试结果	结论
UC-N04	创建新闻	返回新闻 ID	返回新闻 ID	正确
UC-N05	创建新闻-高危 XSS 内容 (iframe/JS)	拒绝或转义危险脚本内容	未能过滤 iframe 或 JS 脚本，存在 XSS 漏洞	错误
UC-N06	创建新闻-SQL 注入内容	拒绝包含 SQL 关键字、特殊字符	存在 SQL 注入漏洞	错误
UC-N07	创建新闻-XSS 脚本标签 (如	拒绝或转义脚本标签	未能过滤	错误

delById_Success 方法

测试对象	src.main.java.com.demo.service.impl.NewsServiceImpl.java.delById(int id)
测试函数	test.main.java.com.demo.service.impl.NewsServiceImplTest.java.delById_Success()

用例编号	用例描述	预期结果	测试结果	结论
------	------	------	------	----

用例编号	用例描述	预期结果	测试结果	结论
UC-N08	删除新闻	新闻被删除	新闻被删除	正确

update_Success 方法

测试对象	src.main.java.com.demo.service.impl.NewsServiceImpl.java.update(News news)
测试函数	test.main.java.com.demo.service.impl.NewsServiceImplTest.java.update_Success()

用例编号	用例描述	预期结果	测试结果	结论
UC-N09	更新新闻	新闻被更新	新闻被更新	正确
UC-N10	更新新闻-news 对象为 null	拒绝请求并提示非法参数或抛出异常	抛出 NullPointerException	错误

OrderVoService 测试用例

returnOrderVoByOrderID 方法

测试对象	src.main.java.com.demo.service.impl.OrderVoServiceImpl.java.returnOrderVoByOrderID(int orderID)
测试函数	test.main.java.com.demo.service.impl.OrderVoServiceImplTest.java.returnOrderVoByOrderID_Success()

用例编号	用例描述	预期结果	测试结果	结论
UC-OV01	根据订单 ID 返回 OrderVo-正常情况	返回 OrderVo 对象	返回 OrderVo 对象	正确

returnVo 方法

测试对象	src.main.java.com.demo.service.impl.OrderVoServiceImpl.java.returnVo(List list)
测试函数	test.main.java.com.demo.service.impl.OrderVoServiceImplTest.java.returnVo_Success()

测试对象 src.main.java.com.demo.service.impl.OrderVoServiceImpl.java.returnVo(List list)
测试函数)

用例编号	用例描述	预期结果	测试结果	结论
UC-OV02	将订单列表转换为 OrderVo 列表-正常情况	返回 OrderVo 列表	返回 OrderVo 列表	正确
UC-OV03	将空订单列表转换为 OrderVo 列表-边界情况	返回空列表	返回空列表	正确

MessageVoService 测试用例

returnMessageVoByMessageID 方法

测试对象 src.main.java.com.demo.service.impl.MessageVoServiceImpl.java.returnMessageVoByMessageID(int messageID)
测试函数 test.main.java.com.demo.service.impl.MessageVoServiceImplTest.java.returnMessageVoByMessageID_Success()

用例编号	用例描述	预期结果	测试结果	结论
UC-MV01	根据留言 ID 返回 MessageVo-正常情况	返回 MessageVo 对象	返回 MessageVo 对象	正确

returnVo 方法

测试对象 src.main.java.com.demo.service.impl.MessageVoServiceImpl.java.returnVo(List messages)
测试函数 test.main.java.com.demo.service.impl.MessageVoServiceImplTest.java.returnVo_Success()

测试对象 src.main.java.com.demo.service.impl.MessageVoServiceImpl.java.returnVo(List messages)

函数

用例编号	用例描述	预期结果	测试结果	结论
UC-MV02	将留言列表转换为 MessageVo 列表-正常情况	返回 MessageVo 列表	返回 MessageVo 列表	正确
UC-MV03	将空留言列表转换为 MessageVo 列表-边界情况	返回空列表	返回空列表	正确

二、集成测试用例

测试对象 1: src.main.java.com.demo.controller.admin.AdminMessageController.java

测试脚本: src.test.java.com.demo.controller.admin.AdminMessageControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取消息管理页面 获取待审核消息	AM-01	测试获取消息管理页面	返回状态码 200，模型包含 total 属性，视图名为 admin/message_manage	返回状态码 200，模型包含 total 属性，视图名为 admin/message_manage	通过
	AM-02	测试获取消息列表(正常分页, page=1)	返回状态码 200，响应 JSON 包含 5 条记录	返回状态码 200，响应 JSON 包含 5 条记录	通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
列表					
获取待审核消息列表	AM-03	测试获取消息列表(超出范围页码, page=9999)	返回状态码 200, 响应 JSON 为空列表	返回状态码 200, 响应 JSON 为空列表	通过
通过消息	AM-04	测试通过消息 (正常 ID)	返回状态码 200, 响应为 true	返回状态码 200, 响应为 true	通过
拒绝消息	AM-05	测试拒绝消息 (正常 ID)	返回状态码 200, 响应为 true	返回状态码 200, 响应为 true	通过
删除消息	AM-06	测试删除消息 (正常 ID)	返回状态码 200, 响应为 true	返回状态码 200, 响应为 true	通过
删除消息	AM-07	测试删除消息 (非法 ID, -1)	返回状态码 200, 响应为 true	返回状态码 200, 响应为 true	通过

测试对象 2: src.main.java.com.demo.controller.user.MessageController.java

测试脚本: src.test.java.com.demo.controller.user.MessageControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
-----	------	------	------	------	----

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取消息列表页面	UM-01	测试获取消息列表页面(用户已登录)	返回状态码 200，模型包含 total 和 user_total 属性，视图名为 message_list	返回状态码 200，模型包含 total 和 user_total 属性，视图名为 message_list	通过
获取消息列表页面	UM-02	测试获取消息列表页面(用户未登录)	捕获 NullPointerException 或 LoginException 异常	捕获 NullPointerException 或 LoginException 异常	通过
获取已通过消息列表	UM-03	测试获取通过的消息列表(正常分页, page=1)	返回状态码 200，响应 JSON 包含 5 条记录	返回状态码 200，响应 JSON 包含 5 条记录	通过
获取用户消息列表	UM-04	测试获取用户的消息列表(用户已登录)	返回状态码 200，响应 JSON 包含 5 条记录	返回状态码 200，响应 JSON 包含 5 条记录	通过
获取用户消息列表	UM-05	测试获取用户的消息列表(用户未登录)	捕获 LoginException 异常，消息为“请登录！”	捕获 LoginException 异常，消息为“请登录！”	通过
发送消息	UM-06	测试发送消息(正常参数)	返回状态码 302，重定向到/message_list	返回状态码 302，重定向到/message_list	通过
修改消息	UM-07	测试修改消息(正常 ID)	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
删除消息	UM-08	测试删除消息(正常 ID)	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
消息内容校验	UM-09	系统应拒绝提交空内容的留言	系统允许提交空内容的留言	提示内容不能为空	不通过
权限与会话校验	UM-10	未登录用户请求分页接口，系统应抛出异常或提示登录	系统存在空指针异常，未正确处理未登录状态	提示未登录或跳转登录页	不通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
安全性校验	UM-11	用户尝试修改他人留言，系统应拒绝并提示无权限	系统允许用户修改他人的留言	提示无权限操作	不通过
SQL 注入防御	UM-12	提交内容中包含 SQL 关键字时，系统应进行过滤	系统未过滤 SQL 关键字，存在 SQL 注入漏洞	拦截 SQL 关键字，防止注入	不通过
XSS 注入防御	UM-13	提交内容中包含 JavaScript 脚本时，系统应进行转义或过滤	系统未过滤 JavaScript 脚本，存在 XSS 漏洞	防止 XSS 注入，脚本内容被转义或拒绝	不通过

测试对象 3: src.main.java.com.demo.controller.admin.AdminVenueController.java

测试脚本: src.test.java.com.demo.controller.admin.AdminVenueControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取场馆管理页面	AV-01	测试获取场馆管理页面	返回状态码 200，模型包含 total 属性，视图名为 admin/venue_manage	返回状态码 200，模型包含 total 属性，视图名为 admin/venue_manage	通过
场馆编辑页面	AV-02	测试场馆编辑页面(正常 ID)	返回状态码 200，模型包含 venue 属性，视图名为 /admin/venue_edit	返回状态码 200，模型包含 venue 属性，视图名为 /admin/venue_edit	通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
场馆添加页面	AV-03	测试场馆添加页面	返回状态码 200，视图名为 /admin/venue_add	返回状态码 200，视图名为 /admin/venue_add	通过
获取场馆列表	AV-04	测试获取场馆列表(正常分页, page=1)	返回状态码 200，响应 JSON 包含 5 条记录	返回状态码 200，响应 JSON 包含 5 条记录	通过
添加场馆	AV-05	测试添加场馆(带图片上传)	返回状态码 302，重定向到 venue_manage	返回状态码 302，重定向到 venue_manage	通过
修改场馆	AV-06	测试修改场馆(带图片上传)	返回状态码 302，重定向到 venue_manage	返回状态码 302，重定向到 venue_manage	通过
删除场馆	AV-07	测试删除场馆(正常 ID)	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
检查场馆名称	AV-08	测试检查场馆名称(可用名称)	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
检查场馆名	AV-09	测试检查场馆名称(不可用名称)	返回状态码 200，响应为 false	返回状态码 200，响应为 false	通过

功能点名称	用例编号	用例描述	测试结果	预期结果	结论
-------	------	------	------	------	----

测试对象 4: src.main.java.com.demo.controller.IndexController.java

测试脚本: src.test.java.com.demo.controller.IndexControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
首页	IN-01	测试首页	返回状态码 200，模型包含 news_list、venue_list 和 message_list 属性，视图名为 index	返回状态码 200，模型包含 news_list、venue_list 和 message_list 属性，视图名为 index	通过
管理员首页	IN-02	测试管理员首页	返回状态码 200，视图名为 admin/admin_index	返回状态码 200，视图名为 admin/admin_index	通过

测试对象 5: src.main.java.com.demo.controller.user.VenueController.java

测试脚本: src.test.java.com.demo.controller.user.VenueControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取场馆详情页面	UV-01	测试获取场馆详情页面	返回状态码 200，模型包含 venue 属性，视图名为 venue	返回状态码 200，模型包含 venue 属性，视图名为 venue	通过
获取场馆列表页面	UV-02	测试获取场馆列表页面	返回状态码 200，模型包含 venue_list 和 total 属性，视图名为 venue_list	返回状态码 200，模型包含 venue_list 和 total 属性，视图名为 venue_list	通过
获取场馆	UV-03	测试获取场馆列表(正常分页, page=1)	返回状态码 200，响应 JSON 包含 5 条记	返回状态码 200，响应 JSON 包含 5 条记	通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
列表			录	录	
获取场馆列表	UV-04	测试获取场馆列表(超出范围页码, page=9999)	返回状态码 200, 响应 JSON 为空列表	返回状态码 200, 响应 JSON 为空列表	通过
SQL 注入防御	UV-05	场馆 ID 传入非法 SQL 关键字, 验证是否存在注入风险	系统未正确验证场馆 ID, 可能存在 SQL 注入风险	拦截非法 ID 输入, 防止 SQL 注入	不通过

测试对象 6: src.main.java.com.demo.controller.user.NewsController.java

测试脚本: src.test.java.com.demo.controller.user.NewsControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取新闻详情页面	UN-01	测试获取新闻详情页面	返回状态码 200, 模型包含 news 属性, 视图名为 news	返回状态码 200, 模型包含 news 属性, 视图名为 news	通过
获取新闻列表页面	UN-02	测试获取新闻列表页面	返回状态码 200, 模型包含 news_list 和 total 属性, 视图名为 news_list	返回状态码 200, 模型包含 news_list 和 total 属性, 视图名为 news_list	通过
获取新闻列表	UN-03	测试获取新闻列表(正常分页, page=1)	返回状态码 200, 响应 JSON 包含 5 条记录	返回状态码 200, 响应 JSON 包含 5 条记录	通过
获取新闻列表	UN-04	测试获取新闻列表(超出范围页码, page=9999)	返回状态码 200, 响应 JSON 为空列表	返回状态码 200, 响应 JSON 为空列表	通过

测试对象 7: src.main.java.com.demo.controller.user.UserController.java

测试脚本: src.test.java.com.demo.controller.user.UserControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取注册页面	UU-01	测试注册页面	返回状态码 200, 视图名为 signup	返回状态码 200, 视图名为 signup	通过

功能点	用例 编号	用例描述	测试结果	预期结果	结 论
获取登 录页面	UU- 02	测试登录页面	返回状态码 200，视 图名为 login	返回状态码 200，视 图名为 login	通 过
获取用 户信息 页面	UU- 03	测试用户信息页 面	返回状态码 200，视 图名为 user_info	返回状态码 200，视 图名为 user_info	通 过
普通用 户登录	UU- 04	测试普通用户登 录	返回状态码 200，响 应为"/index"	返回状态码 200，响 应为"/index"	通 过
管理员 登录	UU- 05	测试管理员登录	返回状态码 200，响 应为"/admin_index"	返回状态码 200，响 应为"/admin_index"	通 过
登录失 败	UU- 06	测试登录失败	返回状态码 200，响 应为"false"	返回状态码 200，响 应为"false"	通 过
用户注 册	UU- 07	测试用户注册	返回状态码 302，重 定向到 login	返回状态码 302，重 定向到 login	通 过
用户登 出	UU- 08	测试用户登出	返回状态码 302，重 定向到/index	返回状态码 302，重 定向到/index	通 过
管理员 登出	UU- 09	测试管理员登出	返回状态码 302，重 定向到/index	返回状态码 302，重 定向到/index	通 过
检查密 码	UU- 10	测试检查密码(正 确密码)	返回状态码 200，响 应为 true	返回状态码 200，响 应为 true	通 过
检查密 码	UU- 11	测试检查密码(错 误密码)	返回状态码 200，响 应为 false	返回状态码 200，响 应为 false	通 过
更新用 户信息	UU- 12	测试更新用户信 息(不修改密码和 图片)	返回状态码 302，重 定向到 user_info	返回状态码 302，重 定向到 user_info	通 过
更新用 户信息	UU- 13	测试更新用户信 息(修改密码和图 片)	返回状态码 302，重 定向到 user_info	返回状态码 302，重 定向到 user_info	通 过
用户 ID 校 验	UU- 14	系统应拒绝包含 特殊字符的用户 ID	系统接受非法 ID，未 做校验	应提示用户 ID 非法， 禁止特殊字符	不 通 过
密码存 储安全	UU- 15	系统应使用密码 哈希算法而非明 文存储	系统将密码以明文方 式存储	应加密存储密码，使 用哈希算法	不 通 过
密码明 文存储	UU- 16	检查是否存在明 文密码存储逻辑	系统存在明文密码存 储	所有密码必须加密保 存	不 通

功能点	用例编号	用例描述	测试结果	预期结果	结论
检查					通过
弱密码校验	UU-17	系统应拒绝过于简单的弱密码（如 123456）	系统接受了简单密码	应提示密码强度不足，拒绝注册/修改	不通过

测试对象 8: src.main.java.com.demo.controller.user.OrderController.java

测试脚本: src.test.java.com.demo.controller.user.OrderControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取订单管理页面	UO-01	测试获取订单管理页面(用户已登录)	返回状态码 200，模型包含 total 属性，视图名为 order_manage	返回状态码 200，模型包含 total 属性，视图名为 order_manage	通过
获取订单管理页面	UO-02	测试获取订单管理页面(用户未登录)	捕获 LoginException 异常	捕获 LoginException 异常	通过
获取下单页面	UO-03	测试获取下单页面(指定场馆)	返回状态码 200，模型包含 venue 属性，视图名为 order_place	返回状态码 200，模型包含 venue 属性，视图名为 order_place	通过
获取下单	UO-04	测试获取下单页面(未指定场馆)	返回状态码 200，视图名为 order_place	返回状态码 200，视图名为 order_place	通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
单页面获取用户订单列表 获取用户订单列表 添加订单 添加订单 完成订单 获取订单编辑	UO-05	测试获取用户订单列表(用户已登录)	返回状态码 200	返回状态码 200	通过
	UO-06	测试获取用户订单列表(用户未登录)	捕获 LoginException 异常	捕获 LoginException 异常	通过
	UO-07	测试添加订单(用户已登录)	返回状态码 302，重定向到 order_manage	返回状态码 302，重定向到 order_manage	通过
	UO-08	测试添加订单(用户未登录)	捕获 LoginException 异常	捕获 LoginException 异常	通过
	UO-09	测试完成订单	返回状态码 200	返回状态码 200	通过
	UO-10	测试获取订单编辑页面	返回状态码 200，模型包含 venue 和 order 属性，视图名为 order_edit	返回状态码 200，模型包含 venue 和 order 属性，视图名为 order_edit	通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
编辑页面 修改订单 修改订单 删除订单 获取场馆订单列表	UO-11	测试修改订单(用户已登录)	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
	UO-12	测试修改订单(用户未登录)	捕获 LoginException 异常	捕获 LoginException 异常	通过
	UO-13	测试删除订单	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
	UO-14	测试获取场馆订单列表	返回状态码 200	返回状态码 200	通过
	UO-15	系统应拒绝超大整数小时数 (如 Integer.MAX_VALUE)	系统存在整数溢出漏洞：允许使用 MAX_VALUE 作为小时数，可能导致价格计算错误	应抛出校验异常或提示非法小时数	不通过
整数溢出校验 数值范围校验	UO-16	系统应拒绝负数小时数	系统未拒绝负数小时数：-5	应提示非法输入：小时数不能为负	不通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
空场馆校验	UO-17	系统应拒绝场馆为 null 的订单	系统未正确识别 null 场馆并抛出错误	应返回错误提示：场馆不能为空	不通过
安全性校验	UO-18	用户尝试访问他人的订单，系统应进行权限限制	系统存在安全漏洞：允许用户访问他人的订单	应提示无权限操作	不通过
营业时间校验	UO-19	系统应拒绝非营业时间的订单，如预约 2025-06-01T23:00	系统接受了不在营业时间的预约	应提示预约时间超出营业范围	不通过
历史时间校验	UO-20	系统应拒绝过去时间的订单，如预约 2025-04-12T16:48	系统接受了过去时间的预约	应提示预约时间必须大于当前时间	不通过

测试对象 9: src.main.java.com.demo.controller.admin.AdminNewsController.java

测试脚本: src.test.java.com.demo.controller.admin.AdminNewsControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取新闻管理页面	AN-01	测试获取新闻管理页面	返回状态码 200，模型包含 total 属性，视图名为 admin/news_manage	返回状态码 200，模型包含 total 属性，视图名为 admin/news_manage	通过
新闻添加	AN-02	测试新闻	返回状态码 200，视图名为 /admin/news_add	返回状态码 200，视图名为 /admin/news_add	通

功能点	用例编号	用例描述	测试结果	预期结果	结论
页面		添加页面			通过
新闻编辑页面	AN-03	测试新闻编辑页面	返回状态码 200，模型包含 news 属性，视图名为 /admin/news_edit	返回状态码 200，模型包含 news 属性，视图名为 /admin/news_edit	通过
获取新闻列表	AN-04	测试获取新闻列表	返回状态码 200，响应 JSON 包含 10 条记录	返回状态码 200，响应 JSON 包含 10 条记录	通过
删除新闻	AN-05	测试删除新闻	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
修改新闻	AN-06	测试修改新闻	返回状态码 302，重定向到 news_manage	返回状态码 302，重定向到 news_manage	通过
添加新闻	AN-07	测试添加新闻	返回状态码 302，重定向到 news_manage	返回状态码 302，重定向到 news_manage	通过

测试对象 10: src.main.java.com.demo.controller.admin.AdminOrderController.java

测试脚本: src.test.java.com.demo.controller.admin.AdminOrderControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取预约管理页面	AO-01	测试获取预约管理页面	返回状态码 200，模型包含 order_list 和 total 属性，视图名为 admin/reservation_manage	返回状态码 200，模型包含 order_list 和 total 属性，视图名为 admin/reservation_manage	通过

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取未审核订单列表	AO-02	测试获取未审核订单列表	返回状态码 200	返回状态码 200	通过
通过订单	AO-03	测试通过订单	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
拒绝订单	AO-04	测试拒绝订单	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过

测试对象 11: src.main.java.com.demo.controller.admin.AdminUserController.java

测试脚本: src.test.java.com.demo.controller.admin.AdminUserControllerTest.java

功能点列表

功能点	用例编号	用例描述	测试结果	预期结果	结论
获取用户管理页面	AU-01	测试获取用户管理页面	返回状态码 200，模型包含 total 属性，视图名为 admin/user_manage	返回状态码 200，模型包含 total 属性，视图名为 admin/user_manage	通过
用户	AU-	测试用	返回状态码 200，视图名为	返回状态码 200，视图名为	通

功能点	用例编号	用例描述	测试结果	预期结果	结论
添加页面	02	户添加页面	admin/user_add	admin/user_add	通过
获取用户列表	AU-03	测试获取用户列表	返回状态码 200，响应 JSON 包含 10 条记录	返回状态码 200，响应 JSON 包含 10 条记录	通过
用户编辑页面	AU-04	测试用户编辑页面	返回状态码 200，模型包含 user 属性，视图名为 admin/user_edit	返回状态码 200，模型包含 user 属性，视图名为 admin/user_edit	通过
修改用户	AU-05	测试修改用户	返回状态码 302，重定向到 user_manage	返回状态码 302，重定向到 user_manage	通过
添加用户	AU-06	测试添加用户	返回状态码 302，重定向到 user_manage	返回状态码 302，重定向到 user_manage	通过
检查用户 ID	AU-07	测试检查用户 ID(可用)	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过
检查用户 ID	AU-08	测试检查用户 ID(不可用)	返回状态码 200，响应为 false	返回状态码 200，响应为 false	通过
删除用户	AU-09	测试删除用户	返回状态码 200，响应为 true	返回状态码 200，响应为 true	通过