七、测试计划

[1．引言 2](#_Toc506967652)

[1.1编写目的 2](#_Toc506967653)

[1.2项目背景 2](#_Toc506967654)

[1.3定义 2](#_Toc506967655)

[1.4参考资料 3](#_Toc506967656)

[2．任务概述 5](#_Toc506967657)

[2.1目标 5](#_Toc506967658)

[2.2运行环境 5](#_Toc506967659)

[2.3需求概述 5](#_Toc506967660)

[2.4条件与限制 6](#_Toc506967661)

[3．计划 7](#_Toc506967662)

[3.1测试方案 7](#_Toc506967663)

[3.2测试项目 8](#_Toc506967664)

[3.3测试准备 13](#_Toc506967665)

[3.4测试机构及人员 13](#_Toc506967666)

[4．测试项目说明 15](#_Toc506967667)

[4.1测试项目名称及测试内容 15](#_Toc506967668)

[4.2测试用例 15](#_Toc506967669)

[4.3进度 17](#_Toc506967670)

[4.4条件 17](#_Toc506967671)

[4.5测试资料 18](#_Toc506967672)

[5．评价 19](#_Toc506967673)

[5.1范围 19](#_Toc506967674)

[5.2准则 19](#_Toc506967675)

# 1．引言

## 1.1编写目的

本测试计划的编写旨在为项目团队提供系统的测试框架与详细的执行指南，确保软件产品能够满足预定的功能需求、性能要求和质量标准。通过对系统进行全面的测试，帮助发现潜在的缺陷、确保系统的稳定性、可靠性和安全性。

测试计划主要供项目开发人员、测试人员、产品经理及相关利益相关者参考，以指导后续的测试执行工作，并确保测试过程的一致性和高效性。

## 1.2项目背景

本项目是由软件工程课程委托开发的电商平台系统，旨在提供一个在线商品交易和支付的平台，涵盖商家、用户和平台管理员三个核心模块。项目的目标是建立一个高效、易用且安全的电商平台，帮助商家和消费者实现线上交易，同时为平台管理员提供全面的管理和数据分析功能。

该项目由奚雪峰监管，并依照相关的法律法规及行业标准进行开发。项目的开发周期预计为 10周，并且计划通过逐步交付与迭代的方式来完善系统的功能与性能。

## 1.3定义

在本测试计划中，涉及到的部分专门术语和缩写定义如下：

* **功能测试**：对系统功能进行全面检查，确保每项功能都能够按预期工作。
* **性能测试**：测试系统在高负载下的响应时间和稳定性，确保系统能够承受预期的并发用户量。
* **集成测试**：验证系统各个模块之间的接口是否能够正常工作，确保系统的组件能有效协同。
* **安全性测试**：评估系统在面对安全威胁时的抵御能力，包括数据泄露、非法访问等问题。
* **回归测试**：在修改系统后进行的测试，确保新修改没有引入新的缺陷，且系统的其他部分仍然正常工作。
* **测试用例**：测试人员根据需求文档和设计文档编写的、用于验证功能是否按预期执行的具体测试场景。
* **API**（Application Programming Interface，应用程序接口）：系统各个模块之间的交互接口，通常通过HTTP协议进行数据交换。

**缩写词：**

* **UI：用户界面 (User Interface)**
* **REST：表述性状态转移（Representational State Transfer）**
* **JWT：JSON Web Token**
* **SQL：结构化查询语言（Structured Query Language）**
* **QA：质量保证（Quality Assurance）**

## 1.4参考资料

本测试计划所引用的主要资料包括：

**项目需求分析**

1**. 标题：**软件商城交易平台项目需求分析说明书

**2. 编号**：001

**3. 发布单位**：项目一（13组）

**4. 发布时间：**2024-9-25

**项目任务书**

**1. 标题：**软件商城交易平台计划任务书

**2. 编号**：002

**3. 发布单位：**项目一（13组）

**4. 发布时间**：2024-10-6

**系统界面功能说明书**

**1. 标题：**软件商城交易平台系统界面功能说明书

**2. 编号：**003

**3. 发布单位**：项目一（13组）

**4. 发布时间**：2024-10-12

**需求规格说明书**

**1. 标题：**软件商城交易平台需求规格说明书

**2. 编号：**004

**3. 发布单位**：项目一（13组）

**4. 发布时间：**2024-10-16

**概要设计说明书**

**1. 标题：**软件商城交易平台概要设计说明书

**2. 编号：**005

**3. 发布单位：**项目一（13组）

**4. 发布时间：**2024-10-30

**详细设计说明书**

**1. 标题：**软件商城交易平台详细设计说明书

**2. 编号：**006

**3. 发布单位：**项目一（13组）

**4. 发布时间：**2024-11-20

**用户操作手册**

**1. 标题：**软件商城交易平台用户操作手册

**2. 编号：**007

**3. 发布单位：**项目一（13组）

**4. 发布时间：**2024-11-24

**软件开发标准与规范**

《软件工程测试文档》

《软件开发过程管理规范》

《软件工程 安全性测试规范》

# 2．任务概述

## 2.1目标

本测试说明书的目标是确保开发的软件商城交易平台管理系统符合需求规格说明书和设计说明书中的功能和非功能要求。通过全面的测试，验证系统的各项功能是否实现预期，确保系统在生产环境中的稳定性、安全性和性能。具体目标包括：

• **功能验证：**确保系统的各项功能模块（如用户管理、商品管理、订单管理、支付模块等）能够正常运行，满足业务需求。

• **性能测试：**确保系统在高并发、大数据量的情况下能够高效响应，符合性能要求。

• **安全性测试：**确保用户数据、支付信息等敏感数据得到保护，防止系统被攻击。

• **兼容性测试：**确保系统能够在不同的浏览器、设备上正常使用。

## 2.2运行环境

**硬件环境：**

服务器：至少支持 4 核 8G 内存，1TB 硬盘的云服务器。

客户端：PC端与移动端，支持主流浏览器（Chrome、Firefox、Safari等）和设备（Android、iOS）。

**软件环境：**

操作系统：Linux/Windows Server 2016及以上。

数据库：MySQL 5.7及以上。

开发框架：Vue.js、Node.js、Express。

第三方服务：支付接口（支付宝、微信支付）、短信接口等。

测试工具：Postman（API测试）、JMeter（性能测试）、Selenium（自动化测试）。

**网络环境：**

高速网络连接，支持至少10Gbps带宽，避免网络延迟对测试结果的干扰。

## 2.3需求概述

**1. 功能需求：**

1. 用户模块：用户注册、登录、修改密码、浏览商品、购物车管理等。

2. 商家模块：商品添加、删除、修改、查询，订单管理，数据统计与分析。

3. 管理员模块：商家管理、商品审核、用户管理、销售数据统计等。

**2. 非功能需求：**

1. 性能需求：系统需支持至少1000个并发用户在线访问。

2. 安全性需求：用户密码存储应使用加密（如哈希加盐）方式，支付信息应符合PCI-DSS标准。

3. 响应时间：商品查询响应时间不超过500毫秒，支付过程应在5秒内完成。

4. 可靠性：系统应具有较高的可用性，故障恢复时间应不超过5分钟。

## 2.4条件与限制

**1. 测试数据限制：**

测试数据将根据开发团队提供的模拟数据进行。

在高并发负载测试中，测试数据量将逐步增加，确保系统能够处理大量请求。

**2. 时间限制：**

每个测试模块的执行时间应根据计划时间控制，确保在预定时间内完成所有必要的测试。

**3. 环境限制：**

在测试过程中，可能存在与实际生产环境的差异，例如硬件配置和网络带宽，这可能会对性能测试的精确度产生影响。

# 3．计划

## 3.1测试方案

**3.1.1测试方法**

**1. 黑盒测试**

* **描述**：不考虑系统内部逻辑，仅测试输入与输出是否符合预期。
* **适用场景**：

**功能测试**：验证各功能模块是否满足需求（如用户登录、商品添加、支付流程）。

**界面测试**：检查界面元素（按钮、链接、布局）的正确性。

**边界值测试**：输入边界值（如密码长度、商品库存上下限），确保系统处理正确。

* **测试方式**：

根据需求文档逐项测试。

制定输入数据并对比实际输出与预期输出。

**2. 白盒测试**

* **描述**：基于代码逻辑，分析系统内部结构，检查执行路径的覆盖程度。
* **适用场景**：

**单元测试**：测试最小的功能单元（如单个方法或函数）。

**路径覆盖测试**：验证代码的所有执行路径是否能被触发。

**异常处理测试**：检查系统在异常输入下的行为（如空指针、超时）。

* **测试方式**：

使用工具（如JUnit、PyTest）编写单元测试用例。

检查覆盖率工具的报告，确保覆盖率达到预定目标（如90%以上）。

**3. 自动化测试**

* **描述**：通过编写脚本和工具自动执行测试，减少重复性人工测试。
* **适用场景**：

**回归测试**：验证功能调整或问题修复后，系统是否仍能正常运行。

**接口测试**：检查API的输入输出与数据交互是否正确。

* **测试方式**：

使用Selenium模拟用户操作测试UI。

用Postman或JMeter测试接口性能和功能。

**4. 性能测试**

* **描述**：评估系统在高负载条件下的响应速度、吞吐量和稳定性。
* **适用场景**：

检查系统在高并发访问时的表现。

验证响应时间是否符合预期（如2秒以内）。

* **测试方式**：

使用JMeter或LoadRunner模拟大量用户并发。

设置负载测试场景，记录响应时间和系统瓶颈。

**5. 安全测试**

* **描述**：检查系统安全性，防止恶意攻击和数据泄露。
* **适用场景**：

SQL注入攻击。

XSS（跨站脚本攻击）验证。

用户权限管理和密码强度测试。

* **测试方式**：

使用工具（如OWASP ZAP、Burp Suite）进行渗透测试。

验证系统是否加密敏感数据（如密码、支付信息）。

**6 兼容性测试**

* **描述:** 确保系统在不同设备和浏览器上表现一致。
* **测试方法:**

在主流浏览器（Chrome、Firefox、Safari、Edge）和不同设备（PC、平板、手机）上运行测试。

检查界面显示、操作逻辑是否一致。

**1.5 回归测试**

* **描述：** 确保新功能或修复不破坏已有功能。
* **测试方法:**

编写自动化测试脚本，覆盖关键功能点。

每次迭代后运行完整回归测试。

**3.1.2选取测试用例的原则**

1. **覆盖核心功能：**确保核心业务流程如用户注册、登录、商品搜索、购物车操作、支付、订单管理等场景都有对应的测试用例。
2. **边界值分析：**测试输入的最小值、最大值、超出范围值、空值，验证系统的容错性。例如，商品搜索中输入关键词的最大长度。
3. **等价类划分：**根据输入条件将测试用例划分为等价类，减少冗余测试用例。如用户密码强度校验，可测试弱密码、强密码和无效密码。
4. **正常与异常场景覆盖原则：**验证系统对异常输入的处理能力。如用户登录时输入错误密码、重复注册同一邮箱。
5. **高风险优先原则：**优先测试业务流程中的核心功能和高频使用的功能，如用户注册、登录、支付等。
6. **场景组合测试原则：**测试用例需覆盖完整的业务流程，包括前置条件、操作步骤和后置处理。
7. **性能和压力点：**选择高频操作（如商品搜索、支付）作为测试用例，评估系统性能。
8. **安全测试用例：**模拟恶意输入、非法操作（如篡改支付金额），验证系统的安全性。
9. **兼容性测试用例：**针对主流浏览器和设备设计用例，验证界面和功能的一致性。

## 3.2测试项目

**3.2.1组装测试**

组装测试的目标是验证系统中各模块之间的交互是否正确。

**接口测试**

* **内容：**  
  测试模块间的接口是否正常工作，包括数据格式、调用顺序和响应正确性。
* **名称：** 接口功能测试。
* **目的：**  
  确保模块间的数据传递和交互逻辑符合设计要求。
* **进度：**
  + **第一周：** 测试核心模块（如用户模块与支付模块）的接口交互。
  + **第二周：** 测试次要模块（如商品模块与库存模块）的接口。

**数据流测试**

* **内容：**  
  验证模块间数据的流转是否正确，数据是否完整、不丢失或被篡改。
* **名称：** 数据完整性测试。
* **目的：**  
  确保系统在数据流转过程中，各模块输入输出的准确性。
* **进度：**
  + **第一周：** 测试关键数据流转场景，如用户下单后的数据传递到支付模块。
  + **第二周：** 测试数据边界和异常情况下的数据流转。

**错误处理测试**

* **内容：**  
  模拟错误情况，如接口调用超时、数据缺失等，检查系统的错误处理能力。
* **名称：** 异常交互测试。
* **目的：**  
  确保系统在出现错误时能正确处理，并提供用户友好的错误提示。
* **进度：**
  + **第一周：** 测试核心接口的异常处理能力。
  + **第二周：** 测试次要接口和边缘情况的异常处理。

**业务流程测试**

* **内容：**  
  验证多个模块协同工作的完整业务流程是否正确执行。
* **名称：** 业务流程集成测试。
* **目的：**  
  确保系统的关键业务流程（如商品购买、支付、订单生成）无逻辑错误。
* **进度：**
  + **第一周：** 测试简单业务流程。
  + **第二周：** 测试复杂业务流程和边界情况。

**性能测试（集成部分）**

* **内容：**  
  测试各模块间交互在高并发情况下的性能表现。
* **名称：** 组装性能测试。
* **目的：**  
  验证模块间协作性能是否符合需求，在压力下是否能稳定运行。
* **进度：**
  + **第一周：** 测试低并发场景。
  + **第二周：** 测试高并发和极端压力场景。

**3.2.2确认测试**

确认测试的目标是验证系统是否完全满足需求文档和业务目标。

**1. 功能确认测试**

* **内容：**  
  测试系统的每个功能模块是否达到预期要求。
* **名称：** 功能需求验证。
* **目的：**  
  确保所有功能符合需求文档。
* **进度：**
  + **第一周：** 验证核心功能模块（如用户注册、支付）。
  + **第二周：** 验证次要功能模块（如商品推荐、收藏）。

**1. 用户管理功能**

1. **用户注册**
   * 测试用户注册时是否需要输入所有必填信息（用户名、密码、邮箱等）。
   * 验证用户名、邮箱的唯一性，重复时是否给出正确提示。
   * 测试弱密码或非法格式时的提示功能。
2. **用户登录**
   * 验证用户名和密码是否正确匹配。
   * 测试登录失败的提示信息（用户名或密码错误）。
   * 检查账户锁定（多次尝试错误）或账户禁用状态的处理。
3. **密码重置**
   * 测试密码重置功能是否正常。
   * 验证邮箱或手机验证码的正确性和时效性。
4. **个人信息维护**
   * 测试用户能否更新个人资料，如联系方式、地址等。
   * 验证信息更新后的即时保存。

**2. 商品管理功能**

1. **商品浏览与查询**
   * 测试商品列表是否能正确展示商品信息（名称、价格、库存等）。
   * 检查商品分类和筛选功能的正确性。
   * 验证商品搜索功能是否返回正确结果。
2. **商品详情**
   * 测试商品详情页面是否展示完整信息（图片、描述、规格参数等）。
   * 验证用户是否能正确查看其他用户的评价和评分。

**3. 购物车功能**

1. **添加商品到购物车**
   * 测试用户是否可以成功将商品加入购物车。
   * 检查库存不足时是否提示。
2. **更新购物车**
   * 测试购物车内商品数量的增减功能是否正常。
   * 验证删除购物车商品是否正常生效。
3. **购物车数据保存**
   * 测试用户退出登录后，购物车数据是否仍然保留。

**4. 订单管理功能**

1. **订单创建**
   * 测试用户是否可以成功提交订单。
   * 验证用户填写订单信息（地址、备注）后是否保存成功。
   * 检查订单号生成和显示是否正确。
2. **订单状态更新**
   * 测试订单状态是否根据支付、发货等操作正确更新。
   * 验证用户能否查询订单状态（未支付、已支付、待发货等）。
3. **订单历史**
   * 检查用户是否可以查看历史订单详情。

**5. 支付功能**

1. **支付方式**
   * 测试支持的支付方式是否正常运行（如支付宝、微信、信用卡等）。
   * 验证支付页面加载是否正常。
2. **支付验证**
   * 测试支付成功和失败的处理是否正确。
   * 验证支付后的反馈信息（如支付成功、失败原因提示）。

**6. 用户评价与反馈**

1. **评价功能**
   * 测试用户能否对购买过的商品进行评分和评论。
   * 验证评价是否即时显示，并确保不允许未购买用户评论。
2. **反馈功能**
   * 测试用户能否提交意见或问题。
   * 验证反馈信息是否正确传递到后台。

**7. 下载功能**

1. **下载权限**
   * 测试用户能否在支付后正确下载购买的软件。
   * 验证未支付订单是否无法下载。
2. **下载记录**
   * 检查下载次数和时间是否记录正确。

**8. 安全功能**

1. **数据加密**
   * 测试用户密码是否加密存储。
   * 验证支付信息传输的加密性。
2. **权限管理**
   * 验证未登录用户是否能访问需要权限的页面。
   * 测试不同用户（普通用户、管理员）的权限区分是否正确。

**9. 界面与用户体验**

1. **跨设备兼容性**
   * 测试系统在PC端和移动端的显示效果是否一致。
   * 验证主流浏览器（Chrome、Firefox、Edge、Safari）下的兼容性。
2. **用户界面**
   * 测试UI的易用性，包括按钮功能、导航逻辑等。
   * 验证多语言或主题切换功能（若有）。

**10. 性能及边界测试**

1. **大批量数据**
   * 测试在购物车或订单中添加大量商品的系统响应。
   * 检查查询结果分页的性能。
2. **异常处理**
   * 测试断网、支付中断等异常场景下的处理逻辑。
   * 验证系统错误是否提供用户友好的提示信息。

**2. 用户场景测试**

* **内容：**  
  模拟真实用户场景，测试系统能否正常运行。
* **名称：** 用户体验确认测试。
* **目的：**  
  验证系统是否满足用户需求和操作逻辑。
* **进度：**
  + **第一周：** 模拟核心场景（如购物流程）。
  + **第二周：** 模拟边缘场景（如取消订单、超时支付）。

**3. 性能确认测试**

* **内容：**  
  测试系统在高并发环境下的性能是否达到需求要求。
* **名称：** 性能需求验证测试。
* **目的：**  
  确保系统能稳定支持预期的用户访问量。
* **进度：**
  + **第一周：** 模拟中等流量场景。
  + **第二周：** 模拟峰值流量场景。

**4. 安全性确认测试**

* **内容：**  
  验证系统的安全防护措施是否有效，能否抵御常见安全威胁。
* **名称：** 安全需求验证测试。
* **目的：**  
  确保系统数据和用户信息的安全性符合行业标准。
* **进度：**
  + **第一周：** 检查基础防护（如 SQL 注入防护）。
  + **第二周：** 模拟复杂攻击场景（如 CSRF、XSS 攻击）。

**5. 兼容性确认测试**

* **内容：**  
  测试系统在不同设备、浏览器上的表现是否一致。
* **名称：** 跨平台验证测试。
* **目的：**  
  确保系统在不同运行环境下均表现正常。
* **进度：**
  + **第一周：** 测试主流设备（PC、手机）。
  + **第二周：** 测试主流浏览器（Chrome、Safari、Edge）。

**3.3.3总结进度表**

|  |  |  |
| --- | --- | --- |
| **测试类别** | **第一周** | **第二周** |
| **组装测试** | 核心接口、数据流测试 | 次要模块、异常场景测试 |
| **确认测试** | 核心功能、用户场景验证 | 性能测试、边缘功能确认 |
| **兼容性测试** | 主流设备测试 | 浏览器测试 |
| **性能与安全测试** | 中等流量、基础安全测试 | 高并发场景、复杂攻击验证 |

## 3.3测试准备

在进行测试之前，需要完成以下准备工作：

• **测试环境搭建：**根据运行环境要求，搭建相应的硬件与软件环境，并确保测试环境与生产环境的配置尽可能一致。

• **测试数据准备：**根据需求设计测试用例，并准备相应的模拟数据。

• **测试工具准备：**安装并配置相关测试工具（如Postman、JMeter、Selenium等）。

• **测试人员培训：**确保测试团队了解系统的功能、架构以及测试重点。

## 3.4测试机构及人员

**测试机构名称：**计算机科学与技术22届项目一（13组）开发团队

负责人：

• 项目经理：郁竹一

职责：

负责整体测试计划的制定与协调，确保测试过程按时完成。

协调开发和测试工作，确保测试用例与需求规格的准确性与一致性。

跟踪项目进度，确保测试工作与开发进度同步，及时解决测试过程中遇到的问题。

负责测试结果的汇总与评审，确保系统的稳定性和功能的完整性。

测试成员：

**主程序员**：韩守坤

**职责**：

* + 负责系统功能模块的开发与测试，确保系统的核心功能按预期实现。
  + 协助测试人员理解开发逻辑，提供技术支持与修复开发中发现的bug。
  + 参与编写和执行单元测试、集成测试和系统测试用例，确保开发的代码符合需求。
  + 对测试中发现的问题进行分析和修复，并验证修复结果。

**主程序员**：吴纡怀

**职责**：

* + 负责后端系统的开发、接口实现及与前端的接口对接，确保各模块的数据交互和业务逻辑正确。
  + 协同测试工程师执行API接口测试，确保API的功能性、可靠性与安全性。
  + 参与编写测试用例，执行集成测试与接口测试，确保后端系统与其他模块的无缝对接。
  + 解决测试过程中涉及到的后端开发问题，并对系统的性能进行优化。

**项目经理**：陈松

**职责**：

* + 协调团队成员的工作，确保项目按时、按质量交付。
  + 协调测试与开发之间的沟通，确保测试的需求被准确理解和实现。
  + 负责监控项目进展，确保测试活动按照计划执行，并评估测试过程中的问题与风险。
  + 汇总测试结果，提出优化建议，并与利益相关者进行沟通，确保项目的质量与交付。

**其他成员职责：**

**• 全体成员（韩守坤、吴纡怀、郁竹一、陈松）：**

* + **接口文档制定**：全体成员共同讨论与确认接口文档，确保前后端接口的一致性和可扩展性。
  + **代码评审**：定期进行代码评审，确保代码的质量与一致性，及时发现并解决潜在问题。
  + **测试与质量保证**：全体成员共同参与功能测试、单元测试、集成测试和回归测试，确保系统的稳定性和高质量交付。
  + **问题解决与技术决策**：共同分析开发与测试过程中遇到的问题，做出技术决策并实施解决方案，确保项目的顺利推进。

# 4．测试项目说明

## 4.1测试项目名称及测试内容

测试项目分为功能测试、性能测试、安全性测试、兼容性测试和回归测试，具体内容如下：

1. **功能测试**：验证用户注册、登录、商品查询、购物车管理、订单支付等功能是否符合需求文档。
2. **性能测试**：测试系统在不同并发用户量下的响应时间和稳定性。
3. **安全性测试**：检查系统是否能抵御 SQL 注入、跨站脚本攻击 (XSS)、跨站请求伪造 (CSRF) 等安全威胁。
4. **兼容性测试**：确保系统在不同设备（PC、手机、平板）和浏览器（Chrome、Firefox、Safari、Edge）上表现一致。
5. **回归测试**：验证新功能或修复后，是否影响已有功能的正确性。

## 4.2测试用例

### 4.2.1输入

描述每个测试用例的输入数据和命令。

1. **功能测试：用户注册功能**
   * **输入：** 用户名 "testUser"，密码 "Test@123"，电子邮件 "test@example.com"。
2. **性能测试：订单支付功能**
   * **输入：** 并发用户数 1000，同时提交支付请求的订单编号。
3. **安全性测试：SQL 注入**
   * **输入：** 用户名 " OR 1=1; --"，密码 " OR 1=1; --"。
4. **兼容性测试：商品查询功能**
   * **输入：** 商品关键词 "智能手机"。
5. **回归测试：购物车添加功能**
   * **输入：** 用户登录后，选择商品编号 101 添加到购物车。

### 4.2.2输出

描述每个测试用例的预期输出。

1. **功能测试：**用户注册功能
   * **输出：** 返回状态码 200，提示 "注册成功"。
2. **性能测试：**订单支付功能
   * **输出：** 每个支付请求的平均响应时间小于 2 秒，无超时。
3. **安全性测试：**SQL 注入
   * **输出：** 系统返回错误提示，无敏感信息泄露，未允许非法登录。
4. **兼容性测试：**商品查询功能
   * **输出：** 商品列表显示正确，界面显示无异常。
5. **回归测试：**购物车添加功能
   * **输出：** 返回状态码 200，提示 "添加成功"，购物车显示商品编号 101。

### 4.2.3步骤及操作

**功能测试：用户注册功能**

* 打开注册页面。
* 输入用户名 "testUser"、密码 "Test@123"、电子邮件 "test@example.com"。
* 点击“注册”按钮。

**性能测试：订单支付功能**

* 使用性能测试工具（如 JMeter）配置 1000 并发用户。
* 设置请求目标为订单支付接口，参数为订单编号。
* 启动测试并记录响应时间。

**安全性测试：SQL 注入**

* 打开登录页面。
* 输入用户名 " OR 1=1; --"，密码 " OR 1=1; --"。
* 点击“登录”按钮。

**兼容性测试：商品查询功能**

* 使用不同浏览器（Chrome、Firefox、Safari、Edge）分别打开系统主页。
* 输入商品关键词 "智能手机"。
* 点击“搜索”按钮。

**回归测试：购物车添加功能**

* 用户登录后，选择商品编号 101。
* 点击“添加到购物车”按钮。

### 4.2.4允许偏差

给出实测结果与预期结果之间的允许偏差范围。

1. **功能测试：用户注册功能**
   * **允许偏差：** 响应时间 ±10%，提示信息字符长度偏差 ±5%。
2. **性能测试：订单支付功能**
   * **允许偏差：** 平均响应时间不超过 2.5 秒，最大响应时间不超过 5 秒。
3. **安全性测试：SQL 注入**
   * **允许偏差：** 系统错误提示内容格式可有 ±10% 差异，但不泄露敏感信息。
4. **兼容性测试：商品查询功能**
   * **允许偏差：** 界面显示内容可有 ±5% 的字体或排版差异，但功能正常。
5. **回归测试：购物车添加功能**
   * **允许偏差：** 操作完成时间不超过预期时间 ±10%。

## 4.3进度

**功能测试**

* **开始时间：** 项目开发完成后的第1周
* **结束时间：** 项目开发完成后的第2周
* **目标：** 完成全部核心功能模块的测试并提交测试报告。

**性能测试**

* **开始时间：** 功能测试完成后第1周
* **结束时间：** 功能测试完成后第2周
* **目标：** 测试关键场景的响应时间与系统承载能力，优化性能瓶颈。

**安全性测试**

* **开始时间：** 性能测试完成后第1周
* **结束时间：** 性能测试完成后第2周
* **目标：** 验证系统的安全性，修复潜在漏洞。

**兼容性测试**

* **开始时间：** 性能测试完成后第2周
* **结束时间：** 安全性测试完成后的第1周
* **目标：** 确保系统在不同设备和浏览器上表现一致。

**回归测试**

* **开始时间：** 每次功能修改或新增完成后
* **结束时间：** 项目交付前1周
* **目标：** 确保新功能不破坏已有功能。

## 4.4条件

**设备**

* 服务器设备：1台测试环境服务器，配置与生产环境一致。
* 客户端设备：包含Windows PC、Android设备、iOS设备。
* 浏览器版本：最新版本的Chrome、Firefox、Safari、Edge。

**软件**

* 测试管理工具：Jira、TestRail等。
* 性能测试工具：JMeter、LoadRunner等。
* 安全测试工具：Burp Suite、OWASP ZAP等。
* 自动化测试工具：Selenium、Appium。

**人员**

* 测试经理：1名，负责总体测试进度的管理和资源分配。
* 开发支持：1~2名，协助定位并解决测试中发现的问题。

**网络**

* 测试环境网络需与生产环境相仿，以保证测试结果的可靠性。

## 4.5测试资料

测试所需的资料包括以下内容：

1. **需求文档**
   * 包含系统功能需求说明和非功能性需求说明。
2. **设计文档**
   * 系统架构图、数据库设计文档、接口文档等。
3. **测试计划**
   * 包含测试目标、测试范围、测试策略、测试资源和测试进度的详细描述。
4. **测试用例**
   * 针对每个功能模块和非功能性需求编写的测试用例，包括输入、输出、操作步骤和预期结果。
5. **测试数据**
   * 测试输入数据，包括模拟用户账号、订单信息、商品数据等。
6. **测试工具配置文档**
   * 性能测试脚本、安全测试配置、自动化测试脚本等相关资料。
7. **缺陷管理记录**
   * 用于跟踪测试中发现的缺陷的记录文档。

# 5．评价

## 5.1范围

本测试的范围涵盖了项目的功能性、性能、安全性、兼容性和回归测试，具体如下：

1. **功能测试**
   * 测试范围：所有核心功能模块，包括用户注册、登录、商品查询、购物车、订单支付等操作。
   * 局限性：可能未完全覆盖极端或边界条件下的功能行为。
2. **性能测试**
   * 测试范围：模拟并发用户访问场景下的系统性能，包括响应时间、吞吐量、系统资源占用率。
   * 局限性：测试环境与实际生产环境可能存在差异，可能导致性能评估结果偏差。
3. **安全性测试**
   * 测试范围：验证系统抵御常见安全威胁的能力，包括SQL注入、跨站脚本攻击（XSS）、跨站请求伪造（CSRF）等。
   * 局限性：无法完全覆盖所有可能的攻击手段，可能存在未检测到的潜在漏洞。
4. **兼容性测试**
   * 测试范围：在主流设备和浏览器上验证系统的兼容性，包括显示效果、功能表现一致性等。
   * 局限性：测试未覆盖市场上所有可能的设备和浏览器版本。
5. **回归测试**
   * 测试范围：确保系统在功能修改和新增后，已有功能未受到影响。
   * 局限性：可能未覆盖所有变更的影响范围。

**局限性总结：**  
测试的范围虽然较为全面，但仍存在局限性，例如无法预测所有真实用户行为、无法完全模拟真实生产环境，以及对未知的潜在漏洞或极端条件可能存在覆盖不足。

## 5.2准则

测试结果的评价基于以下准则：

1. **功能测试准则**
   * 所有功能模块按照需求文档测试通过，输出与预期结果一致。
   * 未发现重大或阻塞型缺陷，轻微缺陷在不影响系统主要功能的情况下可延期修复。
2. **性能测试准则**
   * 在模拟的并发场景下，系统响应时间、吞吐量和资源占用符合性能指标要求。
   * 系统在高负载条件下未出现宕机或严重性能下降。
3. **安全性测试准则**
   * 无重大安全漏洞，所有发现的漏洞均已修复或制定合理的缓解措施。
   * 通过常见安全威胁的验证，包括SQL注入、XSS、CSRF等。
4. **兼容性测试准则**
   * 系统在指定设备和浏览器上的表现一致，无明显的功能缺陷或显示错误。
   * 所有关键功能在不同设备和浏览器中均能正常运行。
5. **回归测试准则**
   * 在功能新增或修改后，已有功能未出现任何缺陷。
   * 修复的缺陷在回归测试中未复现。
6. **缺陷处理准则**
   * 阻塞型缺陷需在项目上线前全部修复。
   * 严重缺陷需制定修复计划并在一定时间内完成修复。
   * 轻微缺陷可记录为后续版本改进的参考。