**火车购票项目**

**测试计划**

|  |  |  |
| --- | --- | --- |
| **修订人签字** | **审核人签字** | **批准人签字** |
| 日期： | 日期： | 日期： |

修订历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **变更日期** | **版本** | **说明** | **作者/操作人** |
| 1 | 2023/10/17 | 1.0 | 建立初稿 | 孟庆涛 |
| 2 | 2023/10/24 | 2.0 | 填写历史修订表  增加用户界面与安装测试  资源计划落实到具体人员 | 孟庆涛 |
| 3 | 2023/10/25 | 2.0.BL1 | 评审通过发布基线版本 | 狄锦文 |
|  |  |  |  |  |

**CCB委员会评审**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CCB评审** | **检查项描述** | **要求满足情况** | **是否满足情况** | **备注** |
| A001 | 配置项命名规范是否符合要求 | 是 | 是 |  |
| A002 | 配置项文档是否完整 | 是 | 是 |  |
| A003 | 有无修订历史表 | 是 | 是 |  |
| A004 | 是否发布基线 | 是 | 是 |  |

目录

[1. 简介 4](#_Toc149341996)

[1.1. 目的 4](#_Toc149341997)

[1.2. 背景 4](#_Toc149341998)

[1.3. 范围 4](#_Toc149341999)

[1.4. 项目核实 5](#_Toc149342000)

[2. 测试需求 6](#_Toc149342001)

[2.1. 数据和数据库完整性测试 6](#_Toc149342002)

[2.2. 功能测试 6](#_Toc149342003)

[2.3. 用户界面测试 6](#_Toc149342004)

[2.4. 性能测试 6](#_Toc149342005)

[2.5. 负载测试 6](#_Toc149342006)

[2.6. 强度测试 6](#_Toc149342007)

[2.7. 安全性和访问控制测试 6](#_Toc149342008)

[2.8. 配置测试 6](#_Toc149342009)

[2.9. 安装测试 6](#_Toc149342010)

[3. 测试策略 6](#_Toc149342011)

[3.1. 测试类型 6](#_Toc149342012)

[3.1.1. 数据和数据库完整性测试 6](#_Toc149342013)

[3.1.2. 功能测试 7](#_Toc149342014)

[3.1.3. 用户界面测试 7](#_Toc149342015)

[3.1.4. 性能测试 7](#_Toc149342016)

[3.1.5. 负载测试 8](#_Toc149342017)

[3.1.6. 强度测试 8](#_Toc149342018)

[3.1.7. 安全性和访问控制测试 9](#_Toc149342019)

[3.1.8. 配置测试 9](#_Toc149342020)

[3.1.9. 安装测试 10](#_Toc149342021)

[3.2. 工具 10](#_Toc149342022)

[4. 资源 11](#_Toc149342023)

[4.1. 角色 11](#_Toc149342024)

[4.2. 系统 12](#_Toc149342025)

[5. 项目里程碑 12](#_Toc149342026)

[6. 可交付工件 12](#_Toc149342027)

[6.1. 测试模型 12](#_Toc149342028)

[6.2. 测试记录 13](#_Toc149342029)

[6.3. 缺陷报告 13](#_Toc149342030)

# 简介

## 目的

* 1. 确定现有项目的信息和应测试的软件构件。
  2. 列出推荐的测试需求（高级需求）。
  3. 推荐可采用的测试策略，并对这些策略加以说明。
  4. 确定所需的资源，并对测试的工作量进行估计。
  5. 列出测试项目的可交付元素。

## 背景

测试计划背景：火车站购票系统是一个用于实现火车票在线购买和管理的应用程序。该系统的目标是提供方便快捷的火车票购买服务，确保用户能够准确、安全地购买火车票，并管理已购买的票务信息。

测试对象：

构件：火车站购票系统的各个模块、组件，包括用户界面、订单管理、支付系统、票务查询等。

应用程序：整个火车站购票系统的集成应用，包括前端和后端的交互。

目标：

确保系统的各项功能能够按照需求规格说明书的要求正常工作； 评估系统在高负载情况下的性能表现，包括响应时间、吞吐量等指标，以确保系统能够处理大量用户并保持良好的性能；验证系统的安全性，包括用户身份验证、数据保护等方面，以确保用户的信息和交易安全。

## 范围

测试阶段：

在软件测试的初期，测试主要采用单元测试，对完成的每一个小的功能模块进行测试，确保每一个模块都能实现其功能。接着对软件进行集成测试，在单元测试的基础上，将所有的模块根据软件设计师的要求组装成为一个个子系统，分别对每个子系统进行测试。最后将子系统组装成系统进行测试。

测试类型：

数据库完整性测试、功能测试、用户界面测试、性能测试、负载测试、强度测试、安全性测试、配置测试、安装测试。

假设：

假设测试环境已经搭建并可用，包括硬件、软件和网络环境。

假设测试数据已经准备好，包括各种场景的测试数据和边界值数据。

假设测试团队具备必要的测试技能和知识，能够有效地进行测试工作。

风险或意外事件：

系统性能不符合预期，无法满足高负载情况下的需求。

系统安全性存在漏洞，容易受到攻击或数据泄露。

兼容性问题导致系统在某些平台或设备上无法正常工作。

用户界面设计不符合用户期望，导致系统的可用性较低。

系统在异常情况下出现崩溃或数据丢失等可靠性问题。

约束：

测试资源有限，可能无法覆盖所有测试场景和情况。

测试时间有限，可能无法进行完整的测试覆盖。

## 项目核实

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档  （版本/日期） | 已创建或可用 | 是否评审 | 作者或来源 | 备注 |
| 软件开发计划 | 是 ■ 否□ | 是 ■ 否□ | 安姿铜 |  |
| 软件需求规约 | 是 ■ 否□ | 是 ■ 否□ | 赵欣宇 |  |
| 界面设计文档 | 是 ■ 否□ | 是 ■ 否□ | 罗香雨 |  |
|  |  |  |  |  |

# 测试需求

## 数据和数据库完整性测试

1. 数据完整性测试

* 确保用户输入的数据在存储过程中被正确保存。
* 确保系统能够正确处理各种数据类型（如日期、时间、数字等）。
* 确保系统能够正确处理边界值和异常情况下的数据。
* 确保系统能够正确处理数据的关联性和依赖性。

1. 数据一致性测试

* 确保系统各个模块和组件之间的数据传递和同步正确。
* 确保系统在进行数据更新、删除或修改操作时能够正确更新相关的数据。
* 确保系统在数据冗余或重复的情况下能够正确处理。

1. 数据完整性约束测试

* 确保系统能够正确验证用户输入的数据是否符合预定义的规则和约束。
* 确保系统在数据违反完整性约束时能够正确处理，如拒绝非法数据或给出适当的错误提示。

1. 数据库操作测试

* 验证系统能够正确执行数据库的增、删、改、查操作。
* 确保系统能够正确处理数据库连接、事务和并发操作。
* 检查系统在异常情况下（如数据库故障或连接中断）是否能够正确处理。

1. 数据备份和恢复测试

* 确保系统能够正确执行数据备份和恢复操作。
* 确保系统能够正确恢复数据到指定的状态。
* 确保系统在数据丢失或损坏的情况下能够正确恢复数据。

## 功能测试

1. 用户注册和登录功能测试

* 确保用户能够成功注册新账户，并且账户信息被正确保存。
* 确保用户能够使用注册的账户成功登录系统。
* 确保系统对无效或错误的登录信息能够给出适当的错误提示。

1. 车票查询和预订功能测试

* 确保用户能够根据出发地、目的地和日期等条件成功查询车票信息。
* 确保系统能够正确显示可用的车次、座位和票价信息。
* 确保用户能够成功预订车票，并且相应的座位数和票价正确更新。

1. 订单管理功能测试

* 确保用户能够查看自己的订单信息，包括已购买的车票和相关的行程信息。
* 确保用户能够取消或修改订单，并且相应的座位数和票价正确更新。
* 确保系统能够正确处理订单支付和退款的流程。

1. 乘客信息管理功能测试

* 确保管理员能够添加、修改和删除乘客信息。
* 确保系统能够正确保存和管理乘客信息，并且在预订车票时能够正确关联乘客信息。

1. 车次和座位管理功能测试

* 确保管理员能够添加、修改和删除车次信息。
* 确保系统能够正确管理车次的座位信息，包括座位数、座位类型和票价等。

1. 支付和票务管理功能测试

* 确保用户能够选择合适的支付方式进行支付，并且支付流程能够正确完成。
* 确保系统能够正确处理退票和改签的流程，并且相应的座位数和票价正确更新。

## 用户界面测试

1. 页面布局和导航测试

* 确保系统的页面布局合理，包括元素的位置、大小和对齐等。
* 确保系统的导航菜单和链接能够正确导航用户到不同的页面。

1. 输入表单和验证测试

* 确保系统的输入表单清晰明确，用户能够准确理解输入的要求。
* 确保系统能够正确验证用户输入的数据，并给出适当的错误提示信息。

1. 用户交互和操作测试

* 确保系统的按钮、链接和其他交互元素能够正确响应用户的操作。
* 确保用户能够顺利完成各种操作，如查询车票、预订车票、支付订单等。

1. 多设备和多浏览器兼容性测试

* 确保系统在不同设备（如手机、平板、笔记本电脑等）上的界面适配良好。
* 确保系统在不同浏览器上的显示和功能正常。

1. 可用性和易用性测试

* 确保系统的用户界面直观、易于理解和操作。
* 确保系统的界面设计符合用户的期望，提供良好的用户体验。

## 性能测试

1. 车票查询和预订性能测试

* 测试系统能否在高并发情况下及时响应用户的车票查询和预订请求。
* 测试系统在处理大量查询和预订请求时拥有良好的性能表现，包括响应时间和吞吐量。

1. 订单管理性能测试

* 测试系统在订单管理操作（如订单查询、修改、取消等）时的性能表现。
* 测试系统在高并发情况下能否及时处理订单管理请求，并保持良好的响应时间。

1. 数据库性能测试

* 测试系统对数据库的读写操作的性能表现，包括查询、插入、更新和删除等操作。
* 测试系统在高负载情况下能否保持数据库的稳定性和性能。

1. 并发用户测试

* 测试系统能够同时处理多个用户的请求，并保持良好的性能表现。
* 测试系统在高并发用户访问时是否会出现性能瓶颈或系统崩溃的情况。

## 负载测试

1. 正常负载测试

* 在正常使用情况下，模拟多个用户同时进行车票查询、预订和订单管理等操作。
* 测试系统在正常负载下的性能表现，包括响应时间、吞吐量和资源利用率等指标。

1. 峰值负载测试

* 在系统预期的高峰时段，模拟大量用户同时进行车票查询和预订操作。
* 测试系统在高峰负载下的性能表现，包括响应时间、吞吐量和系统稳定性等指标。

1. 超负载测试

* 在系统容量的极限情况下，模拟超出系统承载能力的大量用户同时进行操作。
* 测试系统在超负载下的性能表现，包括响应时间、吞吐量和系统崩溃恢复能力等指标。

1. 长时间负载测试

* 在持续运行一段时间的情况下，模拟多个用户进行连续的车票查询、预订和订单管理操作。
* 测试系统在长时间负载下的性能表现，包括内存泄漏、资源释放和系统稳定性等指标。

## 强度测试

1. 车票查询和预订强度测试

* 在短时间内连续进行大量的车票查询和预订操作，测试系统的强度承受能力。
* 测试系统在高强度负载下的性能表现，包括响应时间、吞吐量和系统稳定性等指标。

1. 订单管理强度测试

* 在短时间内连续进行大量的订单管理操作，如订单查询、修改、取消等，测试系统的强度承受能力。
* 测试系统在高强度负载下的性能表现，包括响应时间、吞吐量和系统稳定性等指标。

1. 并发用户强度测试

* 模拟大量并发用户同时进行车票查询、预订和订单管理等操作，测试系统的并发处理能力。
* 测试系统在高并发用户访问下的性能表现，包括响应时间、吞吐量和并发用户数的上限等指标。

1. 数据库强度测试

* 在短时间内进行大量的数据库读写操作，如查询、插入、更新和删除等，测试系统对数据库的强度承受能力。
* 测试系统在高强度负载下对数据库的性能表现，包括响应时间、吞吐量和数据库稳定性等指标。

1. 网络强度测试

* 模拟网络环境不稳定或带宽受限的情况，测试系统在网络压力下的性能表现。
* 测试系统在高网络强度负载下的响应时间、数据传输速度和网络连接稳定性等指标。

1. 故障恢复强度测试

* 在强度测试过程中，模拟系统故障或异常情况，如服务器宕机、数据库故障等。
* 测试系统在故障发生后的恢复能力和容错性，包括系统重启时间和数据完整性等指标。

## 安全性和访问控制测试

1、用户认证测试

* 测试系统的用户认证功能，包括用户名和密码验证、单点登录等。
* 确保系统能够正确验证用户身份，并限制未经授权的用户访问。

2、权限控制测试

* 测试系统的权限管理功能，包括用户角色和权限的分配、权限验证等。
* 确保系统能够正确限制用户的操作权限，防止未经授权的用户进行敏感操作。

3、数据保护测试

* 测试系统对用户敏感数据的保护措施，如用户个人信息、支付信息等。
* 确保系统对敏感数据进行加密存储、传输和访问控制，防止数据泄露和篡改。

4、安全漏洞测试

* 测试系统是否存在常见的安全漏洞，如跨站脚本攻击、SQL注入、跨站请求伪造等。
* 确保系统在面对恶意攻击时能够正确防御和处理，保护系统和用户数据的安全。

5、防止恶意攻击测试

* 模拟常见的恶意攻击，如暴力破解密码、恶意文件上传等，测试系统的防御能力。
* 确保系统能够正确识别和阻止恶意攻击，保护系统的安全和稳定。

## 配置测试

1、系统环境配置测试

* 确保系统能够正确部署和配置在各种环境中，并满足系统的运行要求。

2、数据库配置测试

* 测试系统与数据库之间的连接配置，包括数据库类型、连接池设置等。
* 确保系统能够正确连接和配置数据库，并能够正常进行数据读写操作。

3、网络配置测试

* 测试系统的网络配置，包括IP地址、域名解析等设置。
* 确保系统能够正确配置网络参数，确保网络通信正常和稳定。

4、配置文件加载测试

* 测试系统在加载配置文件时的正确性和完整性，包括配置文件路径、格式等。
* 确保系统能够正确加载和解析配置文件，以保证系统的功能和性能正常运行。

5、服务配置测试

* 测试系统各个服务组件的配置情况，如Web服务器、应用服务器等。
* 确保系统的服务组件能够正确配置和启动，以满足系统的功能需求。

6、第三方服务配置测试

* 测试系统与第三方服务的集成配置，如支付接口、短信接口等。
* 确保系统能够正确配置和调用第三方服务，以实现系统的扩展功能。

## 安装测试

1、安装过程测试

* 测试系统的安装过程，包括软件包的下载、解压、安装和配置等步骤。
* 确保系统的安装过程能够顺利进行，并能够正确地完成各项配置和依赖项的安装。

2、系统依赖测试

* 测试系统所依赖的软件和组件是否能够正确安装和配置。
* 确保系统所需的依赖项能够满足系统的运行要求，并能够正确地与系统进行集成。

3、硬件兼容性测试

* 测试系统在不同硬件环境下的安装和运行情况。
* 确保系统能够在各种硬件配置的计算机上正确安装和运行，并能够充分利用硬件资源。

4、配置验证测试

* 验证系统安装后的配置是否正确，包括数据库连接、系统参数设置等。
* 确保系统的配置项能够正确地与系统进行集成和生效，以满足系统的功能和性能要求。

5、安装文档测试

* 测试系统的安装文档的准确性和完整性。
* 确保安装文档能够提供清晰的安装指导，帮助用户正确地安装和配置系统。

6、升级和卸载测试

* 测试系统的升级和卸载过程，包括升级包的安装、数据迁移、配置更新和卸载操作等。
* 确保系统的升级和卸载过程能够顺利进行，并不会导致系统数据丢失或配置冲突。

7、安装日志和错误处理测试

* 测试系统安装过程中的日志记录和错误处理功能。
* 确保系统能够正确记录安装过程中的日志信息，并能够处理安装过程中的错误和异常情况。

# 测试策略

## 测试类型

### 数据和数据库完整性测试

|  |  |
| --- | --- |
| 测试目标 | 确保数据库访问方法和进程正常运行，数据不会遭到损坏。 |
| 技术 | 1、调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据（或对数据的请求）。  2、检查数据库，确保数据已按预期的方式填充， 并且所有的数据库事件都已正常发生；或者检查所返回的数据，确保为正当的理由检索到了正确的数据。 |
| 完成标准 | 所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。 |
| 需考虑的特殊事项 | 1、测试可能需要 DBMS 开发环境或驱动程序在数据库中直接输入或修改数据。  2、进程应该以手工方式调用。  3、应使用小型或最小的数据库（记录的数量有限） 来使所有无法接受的事件具有更大的可视度。 |

### 功能测试

|  |  |
| --- | --- |
| 测试目标 | 确保测试对象的功能正常，其中包括导航、数据输入、处理和检索等功能。 |
| 技术 | 利用有效的和无效的数据来执行各个用例、用例流或功能，以核实以下内容：  1、在使用有效数据时得到预期的结果。  2、在使用无效数据时显示相应的错误消息或警告消息。  3、各业务规则都得到了正确的应用。 |
| 完成标准 | 所计划的测试已全部执行，  所发现的缺陷已全部解决。 |
| 需考虑的特殊事项 | 1. 对需求理解不充分可能导致测试结果的不可信。   2、GUI设计的不合理会影响软件测试的进程。 |

### 用户界面测试

|  |  |
| --- | --- |
| 测试目标 | 核实以下内容：  1、通过测试对象进行的浏览可正确反映业务的功能和需求，这种浏览包括窗口与窗口之间、字段与字段之间的浏览，以及各种访问方法（Tab 健、鼠标移动、 和快捷键）的使用。  2、窗口的对象和特征（例如，菜单、大小、位置、状态和中心）都符合标准。 |
| 技术 | 为每个窗口创建或修改测试，以核实各个应用程序窗口和对象都可正确地进行浏览，并处于正常的对象状态。 |
| 完成标准 | 成功地核实出各个窗口都与基准版本保持一致，或符合可接受标准。 |
| 需考虑的特殊事项 | 并不是所有定制或第三方对象的特征都可访问。 |

### 性能测试

|  |  |
| --- | --- |
| 测试目标 | 核实所指定的事务或业务功能在以下情况下的性能行为：  1、正常的预期工作量  2、预期的最繁重工作量 |
| 技术 | 1、使用为功能或业务周期测试制定的测试过程。  2、通过修改数据文件来增加事务数量。  3、脚本应该在一台计算机上运行（最好是以单个用户、单个事务为基准），并在多个客户机上重复。 |
| 完成标准 | 1、单个事务或单个用户：在每个事务所预期或要求的时间范围内成功地完 成测试脚本，没有发生任何故障。  2、多个事务或多个用户： 在可接受的时间范围内成功地完成测试脚本，没有发生任何故障。 |
| 需考虑的特殊事项 | 1、综合的性能测试还包括在服务器上添加后台工作量。  2、性能测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  3、性能测试所用的数据库应该是实际大小或相同缩放比例的数据库。 |

### 负载测试

|  |  |
| --- | --- |
| 测试目标 | 核实所指定的事务或商业理由在不同的工作量条件下的性能行为时间。 |
| 技术 | 1、使用为功能或业务周期测试制定的测试。  2、通过修改数据文件来增加事务数量， 或通过修改测试来增加每项事务发生的次数。 |
| 完成标准 | 多个事务或多个用户： 在可接受的时间范围内成功地完成测试，没有发生任何故障。 |
| 需考虑的特殊事项 | 1、负载测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。  2、负载测试所用的数据库应该是实际大小或相同缩放比例的数据库。 |

### 强度测试

|  |  |
| --- | --- |
| 测试目标 | 确定和记录那些使系统无法继续正常运行的情况或条件。 |
| 技术 | 1、使用为性能评测或负载测试制定的测试。  2、要对有限的资源进行测试，就应该在一台计算机上运行测试，而且应该减少或限制服务器上的RAM 和DASD。  3、对于其他强度测试，应该使用多台客户机来运行相同的测试或互补的测试，以产生最繁重的事务量或最差的事务组合。 |
| 完成标准 | 所计划的测试已全部执行，并且在达到或超出指定的系统限制时没有出现任何软件故障，或者导致系统出现故障的条件并不在指定的条件范围之内。 |
| 需考虑的特殊事项 | 1、如果要增加网络工作强度，可能会需要使用网络工具来给网络加载消息或信息包。  2、应该暂时减少用于系统的DASD，以限制数据库可用空间的增长。  3、使多个客户机对相同的记录或数据账户同时进行的访问达到同步。 |

### 安全性和访问控制测试

|  |  |
| --- | --- |
| 测试目标 | 1、应用程序级别的安全性：核实参与者只能访问其所属用户类型已被授权访问的那些功能或数据。  2、系统级别的安全性：核实只有具备系统和应用程序访问权限的参与者才能访问系统和应用程序。 |
| 技术 | 应用程序级别的安全性：  1、确定并列出各用户类型及其被授权访问的功能或数据。  2、为各用户类型创建测试，并通过创建各用户类型所特有的事务来核实其权限。  3、修改用户类型并为相同的用户重新运行测试。 对于每种用户类型，确保正确地提供或拒绝了这些附加的功能或数据。  系统级别的访问：  必须与相应的网络或系统管理员一起对系统访问权进行检查和讨论。 |
| 完成标准 | 各种已知的Actor 类型都可访问相应的功能或数据，而且所有事务都按照预期的方式运行，并在先前的应用程序功能测试中运行了所有的事务。 |
| 需考虑的特殊事项 | 由于此测试可能是网络管理或系统管理的职能，可能会不需要执行此测试。 |

### 配置测试

|  |  |
| --- | --- |
| 测试目标 | 核实测试对象可在所需的硬件和软件配置中正常运行。 |
| 技术 | 1、使用功能测试脚本。  2、在测试过程中或在测试开始之前，打开各种与非测试对象相关的软件，然后将其关闭。  3、执行所选的事务，以模拟参与者与测试对象软件和非测试对象软件之间的交互。  4、重复上述步骤，尽量减少客户机工作站上的常规可用内存。 |
| 完成标准 | 对于测试对象软件和非测试对象软件的各种组合，所有事务都成功完成，没有出现任何故障。 |
| 需考虑的特殊事项 | 1、需要、可以使用并可以通过桌面访问哪种非测试对象软件。  2、通常使用的是哪些应用程序。  3、应用程序正在运行什么数据。  4、作为此测试的一部分，应将整个系统、Netware、网络服务器、 数据库等都记录下来. |

### 安装测试

|  |  |
| --- | --- |
| 测试目标 | 核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中：  1、首次安装。以前从未安装过本火车票购票软件的新计算机。  2、更新。以前安装过相同版本的本火车票购票软件的计算机。  3、更新。以前安装过本火车票购票软件的较早版本的计算机。 |
| 技术 | 1、手工开发脚本或开发自动脚本，以验证目标计算机的状况。  2、启动或执行安装。  3、使用预先确定的功能测试脚本子集来运行事务。 |
| 完成标准 | 本火车票购票软件事务成功执行，没有出现任何故障。 |
| 需考虑的特殊事项 | 应该选择本火车票购票软件的哪些事务才能准确地测试出本火车票购票软件应用程序已经成功安装，而且没有遗漏主要的软件构件。 |

## 工具

|  |  |  |
| --- | --- | --- |
|  | 工具 | 产商 |
| 测试管理 | RequisitePro | Rational |
| 缺陷跟踪 | ClearQuest | Rational |
| 用于功能性测试的 ASQ 工具 | Rational Robot | Rational |
| 用于性能测试的 ASQ 工具 | Visual Quantify | Rational |
| 测试覆盖监测器或评测器 | Visual Pure Coverage | Rational |
| 项目管理 | Project Word Excel | microsoft |

# 资源

## 角色

|  |  |  |
| --- | --- | --- |
| **角色** | **分配专职角色数量** | **具体职责** |
| 测试经理 | 1人 暂定为孟庆涛 | 进行管理监督。  职责：提供技术指导、获取适当资源、提供管理报告 |
| 测试设计员 | 1人 暂定为孟庆涛 | 确定测试用例、确定测试用例的优先级并实施测试用例  职责：生成测试计划、生成测试模型、评估测试工作的有效性 |
| 测试员 | 2人 暂定为孟庆涛、郑佳策 | 执行测试  职责：执行测试、记录结果、从错误中恢复、记录变更请求 |
| 测试系统管理员 | 1人 暂定为孟庆涛 | 确保测试环境和资产得到管理和维护  职责：管理测试系统、分配和管理角色对测试系统的访问权 |
| 数据库管理员 | 1人 暂定为狄锦文 | 确保测试数据环境和资产得到管理和维护  职责：管理测试数据 |
| 设计员 | 2人 暂定为孟庆涛、郑佳策 | 确定并定义测试类的操作、属性和关联。  职责：  确定并定义测试类、确定并定义测试包 |
| 实施员 | 2人 暂定为孟庆涛、郑佳策 | 实施测试类和测试包，并对它进行单元测试。  职责：  创建在测试模型中实施的测试类和测试包 |

## 系统

|  |  |
| --- | --- |
| 系统资源 | |
| 资源 | 名称/类型 |
| 数据库服务器  网络或子网  服务器名称  数据库名称 |  |
| TBD |
| TBD |
| TBD |
| 客户端测试PC  包括特殊的配置需求 |  |
| TBD |
| 测试存储库  网络或子网  服务器名称 |  |
| TBD |
| TBD |
| 测试开发PC | TBD |

# 项目里程碑

| **项目里程碑** | **工作量** | **开始时间** | **结束时间** |
| --- | --- | --- | --- |
| 指定测试计划 | 7 | 2023/10/17 | 2023/10/24 |
| 设计测试 | 7 | 2023/10/24 | 2023/10/31 |
| 实施测试 | 7 | 2023/10/31 | 2023/11/7 |
| 执行测试 | 14 | 2023/11/7 | 2023/11/21 |
| 对测试进行评估 | 7 | 2023/11/21 | 2023/11/28 |

# 可交付工件

## 测试模型

由测试设计员创建，预计于2023/10/31交付给测试员。

## 测试记录

由测试员创建，预计于2023/11/7交付给测试经理。

## 缺陷报告

由测试经理创建，预计于2023/11/28交付给项目经理。