**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** | **变更位置** | **变更原因** | **修订人** | **审核人** | **批准人** |
| 1 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

说明：“变更原因”主要是分为：

1. 建立初稿
2. 内容修订
3. 正式发布

目录

[1. 简介 4](#_Toc148900023)

[1.1. 目的 4](#_Toc148900024)

[1.2. 背景 4](#_Toc148900025)

[1.3. 范围 4](#_Toc148900026)

[1.4. 项目核实 5](#_Toc148900027)

[2. 测试需求 5](#_Toc148900028)

[2.1. 用例 5](#_Toc148900029)

[2.2. 功能性需求 6](#_Toc148900030)

[2.3. 非功能性需求 6](#_Toc148900031)

[3. 测试策略 6](#_Toc148900032)

[3.1. 测试类型 6](#_Toc148900033)

[3.1.1. 数据和数据库完整性测试 6](#_Toc148900034)

[3.1.2. 功能测试 7](#_Toc148900035)

[3.1.3. 用户界面测试 7](#_Toc148900036)

[3.1.4. 性能评测 7](#_Toc148900037)

[3.1.5. 负载测试 8](#_Toc148900038)

[3.1.6. 强度测试 8](#_Toc148900039)

[3.1.7. 安全性和访问控制测试 9](#_Toc148900040)

[3.1.8. 配置测试 9](#_Toc148900041)

[3.2. 工具 10](#_Toc148900042)

[4. 资源 10](#_Toc148900043)

[4.1. 角色 10](#_Toc148900044)

[4.2. 系统 11](#_Toc148900045)

[5. 项目里程碑 11](#_Toc148900046)

[6. 可交付工件 12](#_Toc148900047)

[6.1. 测试模型 12](#_Toc148900048)

[6.2. 测试记录 12](#_Toc148900049)

[6.3. 缺陷报告 12](#_Toc148900050)

# 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。

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□ | 项目经理 |  |
| 软件需求规约 | 是 ■ 否□ | 是 ■ 否□ | 系统分析员 |  |
| 界面设计文档 | 是 ■ 否□ | 是 ■ 否□ | 界面设计师 |  |
|  |  |  |  |  |

# 测试需求

## 用例

用户：登录、注册、订票、改签、退票、修订历史记录。

管理员：登录、管理用户、管理火车车次、管理车票价格。

## 功能性需求

该软件面向普通用户及管理员，要求符合常规的操作交互方式。 使用该软件，可以支持用户购买火车票，支付，改签，退票等购票需求；管理员调整用户、车票价格、火车车次等信息。

## 非功能性需求

性能需求：系统需在高负载情况下的拥有良好的性能表现，包括响应时间、吞吐量等。

安全性需求：系统需满足安全性需求，包括用户身份验证、数据保护等方面。

可用性需求：系统需满足易用性需求，用户界面设计符合用户期望。

可靠性需求：系统在各种异常情况下的需保持稳定性和可靠性。

# 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。 |

### 性能评测

|  |  |
| --- | --- |
| 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. |

## 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 | XX人/月 | 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 | XX人/月 | 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 | XX人/月 | 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 | XX人/月 | 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 | XX人/月 | 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 |

## 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。