旅游信息管理系统

软件测试计划

T01

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2022/10/15 | V1.0 | 基本完成软件测试计划 | 陈明惠 |
| 2022/10/17 | V1.1 | 修改部分表达，增加细节 | 陈明惠 |
| 2022/10/24 | V2.0 | 完善测试需求及资源计划 | 陈明惠 |
|  |  |  |  |

1. **简介**
   1. **目的**

旅游信息管理系统软件测试计划文档有助于实现以下目标：

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

旅游信息管理系统的角色分为两个：管理员和导游。

登录功能：用户通过登录界面，选择身份，输入用户名密码，验证成功转到对应界面。

管理员功能：

1. 管理导游信息：查看、添加、删除导游信息，退出系统；
2. 管理城市景点：查看、添加、删除城市景点，退出系统；
3. 管理旅游线路：查看景点，查看、添加、删除旅游路线，退出系统；
4. 管理旅游方案：查看导游和旅游线路，查看、添加、删除旅游方案，退出系统；
5. 查看报名情况，退出系统；
6. 管理论坛：查看、添加、删除论坛，退出系统。

导游功能：

1. 管理个人信息：查看、修改个人信息，退出系统；
2. 管理旅游线路：查看景点，查看、添加、删除旅游路线，退出系统；
3. 管理旅游方案：查看导游和旅游线路，查看、添加、删除旅游方案，退出系统；
4. 查看报名情况，退出系统；
5. 管理论坛：查看、添加、删除论坛，退出系统。
   1. **范围**

测试计划主要针对功能测试。

测试计划分为三阶段：单元测试，集成测试，系统测试。

单元测试：所需测试的功能包括：登录功能，管理员用户的管理导游信息功能、管理城市景点功能、管理旅游线路功能、管理旅游方案功能、查看报名情况功能、管理论坛功能，导游用户的管理个人信息功能、管理旅游线路功能、管理旅游方案功能、查看报名情况功能、管理论坛功能，退出系统功能。对这些功能进行数据和数据库完整性测试、功能测试、用户界面测试。

集成测试：主要分为两个模块：管理员模块和导游模块。针对模块进行数据库完整性测试、功能测试、用户界面测试、安全性和访问控制测试。

系统测试：针对整个系统进行测试。对系统进行数据和数据库完整性测试、功能测试、用户界面测试、性能评测、安全性和访问控制测试、故障转移和恢复测试、安装测试。

* 1. **项目核实**

下表列出了制定测试计划时所使用的文档，并标明了各文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接收或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | 🗹 是  否 | 🗹 是  否 | 张仕易 |  |
| 功能性规约 | 🗹 是  否 | 🗹 是  否 | 张仕易 |  |
| 项目初始计划 | 🗹 是  否 | 🗹 是  否 | 沈晨琳 |  |
| 项目详细计划 | 🗹 是  否 |  是 🗹 否 | 沈晨琳 |  |
| 设计规约 | 🗹 是  否 |  是 🗹 否 | 孟娇 |  |
| 项目原型 | 🗹 是  否 | 🗹 是  否 | 殷志谦 |  |
| 用户手册 |  是 🗹 否 |  是 🗹 否 | 殷志谦 |  |
| 业务模型或业务流程 |  是 🗹 否 |  是 🗹 否 | 孟娇 |  |
| 数据模型或数据流 |  是 🗹 否 |  是 🗹 否 | 孟娇 |  |
| 业务功能和业务规则 |  是 🗹 否 |  是 🗹 否 | 孟娇 |  |
| 项目或业务风险评估 |  是 🗹 否 |  是 🗹 否 | 沈晨琳 |  |

1. **测试需求**

|  |  |  |  |
| --- | --- | --- | --- |
| 测试对象 | 功能性需求 | 详细功能需求 | 非功能性需求 |
| 管  理  员  用  户 | 登录功能 |  | 安全性，界面跳转正确 |
| 管理导游信息 | 查看、添加、删除导游信息，退出系统 | 数据一致性，界面跳转正确 |
| 管理城市和景点 | 查看、添加、删除景点信息，退出系统 | 数据一致性，界面跳转正常 |
| 管理旅游线路 | 查看、添加、删除旅游线路，退出系统 | 数据一致性，界面跳转正确 |
| 管理旅游方案 | 查看导游和旅游线路，查看、添加、删除旅游方案，退出系统 | 数据一致性，界面跳转正确 |
| 查看报名情况 |  | 数据一致性，界面跳转正确 |
| 管理论坛 | 查看、添加、删除论坛，退出系统 | 数据一致性，界面跳转正确 |
| 导  游  用  户 | 登录功能 |  | 安全性，界面跳转正确 |
| 管理个人信息 | 查看、修改个人信息，退出系统 | 数据一致性，界面跳转正确 |
| 管理旅游路线 | 查看景点，查看、添加、删除旅游路线，退出系统 | 数据一致性，界面跳转正确 |
| 管理旅游方案 | 查看导游和旅游线路，查看、添加、删除旅游方案，退出系统 | 数据一致性，界面跳转正确 |
| 查看报名情况 |  | 数据一致性，界面跳转正确 |
| 管理论坛 | 查看、添加、删除论坛，退出系统 | 数据一致性，界面跳转正确 |

针对以上测试需求重点进行用户界面测试、性能评测、访问控制测试、安装测试：

|  |  |
| --- | --- |
| 测试名称 | 测试工作目标 |
| 用户界面测试 | 核实以下内容：  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。  窗口的对象和特征（例如，菜单、大小、位置、状态和中心）都符合界面设计。 |
| 性能评测 | 核实所指定的事务或业务功能在以下情况下的性能行为：  正常的预期工作量  预期的最繁重工作量 |
| 安全性和访问控制测试 | 应用程序级别的安全性：核实用户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。 |
| 安装测试 | 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：  首次安装。以前从未安装过旅游信息管理系统的新计算机  　　更新。以前安装过相同版本的旅游信息管理系统的计算机 |

1. **测试策略**

**3.1 测试类型**

**3.1.1 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。满足数据库表中的所有约束。 |
| 技术： | 调用Oracle数据库中的访问方法和进程，并在其中填充有效的和无效的数据。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。 |
| 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，所有表中的约束都已经满足。 |
| 需考虑的特殊事项： | 在测试这些子系统时，不应将测试对象的用户界面用作数据的接口进程而应该在数据库中以手工方式输入数据并调用相关方法和进程。 |

**3.1.2 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：  在使用有效数据时得到预期的结果。  在使用无效数据时显示相应的错误消息或警告消息。  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： | 确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）。 |

**3.1.3 业务周期测试**

|  |  |
| --- | --- |
| 测试目标 | 确保测试对象及背景的进程都按照所要求的业务模型和时间表正确运行。 |
| 技术： | 通过执行以下活动，测试将模拟若干个业务周期：  1. 将修改或改进对测试对象进行的功能测试，以增加每项功能的执行次数，从而在指定的时间段内模拟若干个不同的用户。  2. 将使用有效的和无效的数据或时间段来执行所有与时间或数据相关的功能。  3. 将在适当的时间执行或启用所有周期性出现的功能。  在测试中还将使用有效的和无效的数据，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： | 系统日期和事件可能需要特殊的支持活动；  需要通过业务模型来确定相应的测试需求和测试过程。 |

**3.1.4 用户界面测试**

|  |  |
| --- | --- |
| 测试目标： | 核实以下内容：  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。  窗口的对象和特征（例如，菜单、大小、位置、状态和中心）都符合界面设计。 |
| 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。 |

**3.1.5 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：  正常的预期工作量  预期的最繁重工作量 |
| 技术： | 使用为功能或业务周期测试制定的测试过程。  通过修改数据文件来增加事务数量，或通过修改脚本来增加每项事务的迭代数量。  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（虚拟的或实际的客户机，请参见下面的“需要考虑的特殊事项”）上重复。 |
| 完成标准： | 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。  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： | 综合的性能测试还包括在服务器上添加后台工作量。  可采用多种方法来执行此操作，其中包括：   * 直接将“事务强行分配到”服务器上，这通常以“结构化查询语言”(SQL)调用的形式来实现。 * 通过创建“虚拟的”用户负载来模拟许多个（通常为数百个）客户机。此负载可通过“远程终端仿真”(Remote Terminal Emulation)工具来实现。此技术还可用于在网络中加载“流量”。 * 使用多台实际客户机（每台客户机都运行测试脚本）在系统上添加负载。 * 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。 * 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。 |

**3.1.6 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： | 使用为功能或业务周期测试制定的测试。  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： | 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。  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。 |

**3.1.7 强度测试**

不进行该种测试

**3.1.8 容量测试**

不进行该种测试

**3.1.9 安全性和访问控制测试**

|  |  |
| --- | --- |
| 测试目标： | 应用程序级别的安全性：核实用户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。 |
| 技术： | 应用程序级别的安全性：确定并列出各用户类型及其被授权访问的功能或数据。  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。  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。 |
| 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： | 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。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。 |

**3.1.10 故障转移和恢复测试**

|  |  |
| --- | --- |
| 测试目标： | 确保恢复进程（手工或自动）将数据库、应用程序和系统正确地恢复到了预期的已知状态。测试中将包括以下各种情况：  客户机断电  通过网络服务器产生的通信中断  数据库指针或关键字无效  数据库中的数据元素无效或遭到破坏 |
| 技术： | 应该使用为功能和业务周期测试创建的测试来创建一系列的事务。一旦达到预期的测试起点，就应该分别执行或模拟以下操作：  客户机断电：关闭PC机的电源。  通过网络服务器产生的中断：模拟或启动网络的通信中断（实际断开通信线路的连接或关闭网络服务器或路由器的电源）。  一旦实现了上述情况（或模拟情况），就应该执行其他事务。而且一旦达到第二个测试点状态，就应调用恢复过程。  在测试不完整的周期时，所使用的技术与上述技术相同，只不过应异常终止或提前终止数据库进程本身。  对以下情况的测试需要达到一个已知的数据库状态。当破坏若干个数据库字段、指针和关键字时，应该以手工方式在数据库中（通过数据库工具）直接进行。其他事务应该通过使用“应用程序功能测试”和“业务周期测试”中的测试来执行，并且应执行完整的周期。 |
| 完成标准： | 在所有上述情况中，应用程序、数据库和系统应该在恢复过程完成时立即返回到一个已知的预期状态。此状态包括仅限于已知损坏的字段、指针或关键字范围内的数据损坏，以及表明进程或事务因中断而未被完成的报表。 |
| 需考虑的特殊事项： | 恢复测试会给其他操作带来许多的麻烦。断开缆线连接的方法可能并不可取或不可行。所以，可能会需要采用其他方法，例如诊断性软件工具。需要系统（或计算机操作）、数据库和网络组中的资源。 |

**3.1.11 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： | 使用功能测试脚本。  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（例如Microsoft应用程序：Excel和 Word），然后将其关闭。  执行所选的事务，以模拟用户与测试对象软件和 非测试对象软件之间的交互。  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： | 需要、可以使用并可以通过桌面访问哪种非测试对象软件。   通常使用的是哪些应用程序？   应用程序正在运行什么数据？例如，在Excel中打开的大型电子表格，或是在Word中打开的 100页文档。   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。 |

**3.1.12 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：  首次安装。以前从未安装过旅游信息管理系统的新计算机  更新。以前安装过相同版本的旅游信息管理系统的计算机 |
| 技术： | 手工开发脚本，以验证目标计算机的状况：首次安装-旅游信息管理系统从未安装过；旅游信息管理系统安装过相同或较早的版本。  启动或执行安装。 |
| 完成标准： | 旅游信息管理系统事务成功执行，没有出现任何故障。 |
| 需考虑的特殊事项： | 旅游信息管理系统应用程序是否已经成功安装，而且没有遗漏主要的软件构件。 |

**3.2 工具**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 产商/自产 | 版本 |
| 测试管理 | Github | 微软公司 |  |
| 缺陷跟踪 | Github | 微软公司 |  |
| 用于功能性测试的ASQ 工具 | TestNG |  | v7.6.0 |
| 用于性能测试的ASQ工具 | TestNG |  | v7.6.0 |
| 测试覆盖监测器或评测器 | EclEmma |  | v3.1.6 |
| 项目管理 | Microsoft Project | 微软公司 |  |
| DBMS工具 | PL/SQL Developer | Allround Automations公司 | 14.0.6 |

1. **资源**

**4.1 角色**

|  |  |  |  |
| --- | --- | --- | --- |
| **人力资源** | | | |
| 角色 | 具体人员 | 所推荐的最少资源  （所分配的专职角色数量） | 具体职责或注释 |
| 测试经理 | 陈明惠 | 1 | 1. 获取适当的资源，提供管理报告 2. 生成测试计划，生成测试模型，评估测试工作的有效性 3. 分配测试任务至具体成员 |
| 测试员 | 陈明惠及参与人员 | 均参与 | 执行测试，记录结果，从错误中恢复，记录变更请求 |
| 设计员 | 孟娇 | 1 | 确定并定义测试类的操作、属性和关联。 |
| 配置员 | 王景 | 1 | 提供测试所需的配置库 |

**4.2 系统**

|  |  |
| --- | --- |
| **系统资源** | |
| **资源** | **名称/类型** |
| 数据库服务器 | Oracle服务器 |
| 网络或子网 | 校园网 |
| 数据库名称 | Oracle |
| 数据库开发管理工具 | PL/SQL Developer |
| 客户端测试PC | Windows10/11 |
| 包括特殊的配置需求 | 安装有Eclipse |
| 测试存储库 | Oracle数据库 |
| 网络或子网 | 校园网 |
| 测试开发PC | Windows10/11 |

1. **项目里程碑**

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作** | **开始日期** | **结束日期** |
| 制定测试计划 | 制定软件测试计划 | 2022/10/13 | 2022/10/20 |
| 设计测试 | 设计软件测试用例 | 2022/11/10 | 2022/11/17 |
| 实施测试 | 按照测试计划实施测试 | 2022/11/18 | 2022/12/1 |
| 执行测试 | 按照测试计划执行测试 | 2022/11/22 | 2022/12/1 |
| 对测试进行评估 | 对测试结果进行评估 | 2022/11/27 | 2022/12/1 |

1. **可交付工件**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **所属类别** | **文档/工具/报告** | **创建人员** | **交付对象** | **交付时间** |
| 测试模型 | 测试设计文档 | 陈明惠 | 沈晨琳 | 2022/11/17 |
| 测试记录 | 使用Github来记录和报告测试结果和测试状态 | 陈明惠 | 沈晨琳 | 2022/12/1 |
| 缺陷报告 | 使用Github来记录、跟踪和报告测试中发生的意外情况及其状态 | 陈明惠 | 沈晨琳 | 2022/12/1 |