**<ATM模拟软件>**

**测试计划**

**版本 <1.0>**

**修订历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **描述** | **作者** |
| <2023/10/20> | <发布版1.0> | <演示版> | <贾壹壹> |
| <2023/10/24> | <发布版2.0> | < Beta此版更新了里程碑计划> | <贾壹壹> |
|  |  |  |  |
|  |  |  |  |

**目录**

**[1. 简介](#_Toc149057433)** [4](#_Toc149057433)

[1.1 目的 4](#_Toc149057434)

[1.2 背景 4](#_Toc149057435)

[1.3 范围 4](#_Toc149057436)

[1.4 项目识别 5](#_Toc149057462)

**[2. 测试需求](#_Toc149057463)** [6](#_Toc149057463)

**[3. 测试策略](#_Toc149057464)** [7](#_Toc149057464)

**[3.1 测试类型](#_Toc149057465)** [7](#_Toc149057465)

[3.1.1 数据和数据库完整性测试 7](#_Toc149057466)

[3.1.2 功能测试 8](#_Toc149057468)

[3.1.3 业务周期测试 9](#_Toc149057470)

[3.1.4 用户界面测试 9](#_Toc149057472)

[3.1.5 性能测评 10](#_Toc149057474)

[3.1.6 访问控制测试 11](#_Toc149057475)

**[3.2 工具](#_Toc149057481)** [13](#_Toc149057481)

**[4. 资源](#_Toc149057482)** [13](#_Toc149057482)

**[4.1 角色](#_Toc149057483)** [13](#_Toc149057483)

**[4.2 系统](#_Toc149057484)** [15](#_Toc149057484)

**[5. 项目 里程碑](#_Toc149057485)** [15](#_Toc149057485)

**[6. 可交付成果](#_Toc149057486)** [16](#_Toc149057486)

**[6.1 测试模型](#_Toc149057488)** [16](#_Toc149057488)

**[6.2 测试记录](#_Toc149057489)** [16](#_Toc149057489)

**[6.3 缺陷报告](#_Toc149057490)** [17](#_Toc149057490)

**[7. 附录 A：项目任务](#_Toc149057491)** [17](#_Toc149057491)

**1. 简介**

**1.1 目的**

这个测试计划的主要目的是确保ATM模拟软件在各种情况下正常运行，满足功能和性能要求，以提供高质量的用户体验。

**1.2 背景**

**1. 测试目标**

本测试计划的目标是测试我们的ATM模拟软件。该软件是一款用于模拟自动取款机（ATM）交易的应用程序，旨在提供用户一个实际ATM操作的仿真体验。我们的测试目标是确保该软件在各种条件下正常运行，提供一致的用户体验，满足其功能和性能要求。

**2. 主要功能**

（1）用户能够进行取款、存款、查询余额、转账等基本操作。

（2）系统能够按照规定的操作流程进行交易确认、错误处理和用户验证。

**3. 软件背景**

ATM模拟软件的开发旨在提供用户一个模拟ATM操作的工具。

这样的排版更清晰地突出了各部分的标题和内容，以满足你的排版要求。如果需要继续排版其他部分，请继续提供相关内容。

**1.3 范围**

**测试阶段：** 本测试计划将包括以下测试阶段：

* **单元测试**：对软件的各个独立模块进行测试，以验证它们的功能是否按照要求正常工作。
* **集成测试**：将各个模块组合起来，测试它们之间的协作和数据传递，确保整体功能正常。
* **系统测试**：在系统级别对软件进行测试，验证整个系统是否满足用户需求。
* **性能测试**：评估软件的性能，包括响应时间、资源利用率和负载能力。

**测试类型：** 我们将执行以下类型的测试：

* **功能测试**：验证各个模块和系统是否按照要求执行基本功能，如取款、存款、查询余额、转账等。
* **用户接口测试**：评估用户界面的友好程度和一致性，包括菜单导航、按钮点击、交易流程等。
* **性能测试**：评估软件的性能，包括响应时间、资源利用率和稳定性。
* **安全和访问控制测试**：验证软件的安全性和防止未经授权访问。

**测试对象的接受和不接受的性能和功能：**

* 我们将接受并测试以下功能和性能：取款、存款、查询余额、转账、用户登录、错误处理、用户友好性、响应时间、资源利用率。
* 我们将不接受测试的功能和性能：故障转移和恢复测试（由于在受控环境中进行安全测试，无法模拟真正的故障情况，此测试不适用）、配置测试（由于这是模拟软件在不同操作系统上的安装和卸载过程，此测试不适用）。

**假设：** 在编写此测试计划的过程中，我们假设以下情况：

* 开发团队已按照需求文档开发了软件，且软件没有明显的功能缺陷。
* 测试数据将准备好并符合需求。
* 测试环境将模拟真实的ATM操作，并且与最终部署环境相似。

**风险和意外事件：** 在测试过程中，可能会出现以下风险和意外事件：

* 软件的性能问题可能导致延迟或系统崩溃。
* 安全漏洞可能会暴露用户数据。
* 由于环境或硬件故障，测试可能会受到干扰。

**约束：** 在测试过程中，存在以下约束：

* 测试团队必须在规定的时间内完成测试，以配合软件的发布时间表。
* 资源限制可能影响测试的完整性和深度。

范围部分提供了有关测试的阶段、类型、接受和不接受的功能和性能、假设、风险和约束的详细信息，以帮助确保测试计划的全面性和准确性。

**1.4 项目识别**

测试计划的可用性表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （和版本/日期）** | **已创建或可用** | **已收到或已审核** | **作者或资源** | 备注 |
| 软件需求规范 | 🗹 是 o 否 | 🗹 是 o 否 | 尹洁 |  |
| 详细软件项目计划 | 🗹 是 o 否 | 🗹 是 o 否 | 张庆霄 |  |
| 用户手册 | o 是 🗹 否 | o 是 🗹 否 | 杨铃兰 |  |
| 设计文档 | o 是 🗹 否 | o 是 🗹 否 | 徐冰妮 |  |

**2. 测试需求**

**1功能性需求：**

**1.1取款功能**

测试目标： 验证取款功能是否按照要求正常工作，包括取款金额的正确性、账户余额的扣除、交易记录的生成等。

测试用例：

* 正常取款：用户输入合法的取款金额，验证账户余额减少，交易记录生成。
* 超过余额的取款：用户输入超过账户余额的取款金额，验证系统拒绝交易。
* 取款金额为零：用户输入零金额的取款，验证系统拒绝交易。
* 非法输入：用户输入无效的取款金额（非数字字符等），验证系统拒绝交易。

**1.2 存款功能**

测试目标： 验证存款功能是否按照要求正常工作，包括存款金额的正确性、账户余额的更新、交易记录的生成等。

测试用例：

* 正常存款：用户输入合法的存款金额，验证账户余额增加，交易记录生成。
* 非法输入：用户输入无效的存款金额（非数字字符等），验证系统拒绝交易。
* 存款金额为零：用户输入零金额的存款，验证系统拒绝交易。

**2 非功能性需求：**

**2.1 性能需求**

测试目标： 验证系统在不同负载条件下的性能。

测试用例：

* 正常负载测试：模拟多个用户执行标准操作，验证系统的响应时间和资源利用率。
* 峰值负载测试：模拟大量用户同时执行操作，验证系统的承载能力。
* 长时间运行测试：持续执行操作，验证系统的稳定性和资源泄漏。

**2.2 安全性需求**

测试目标： 验证系统的安全性和访问控制机制。

测试用例：

* SQL注入攻击：尝试通过输入恶意SQL语句进行攻击，验证系统是否防止了这些攻击。
* 跨站脚本攻击：尝试通过注入恶意脚本进行攻击，验证系统是否防止了这些攻击。
* 访问控制测试：尝试访问未授权的功能，验证系统是否正确限制了访问。

以上是测试需求部分的示例，其中包括功能性需求和非功能性需求，以及相应的测试用例。这有助于确保测试计划覆盖了软件的所有重要方面。根据项目的需求和规模，可以进一步细化和扩展测试需求。

**3. 测试策略**

**3.1 测试类型**

3.1.1 数据和数据库完整性测试

在 <ATM模拟软件> 中，数据库和数据库进程应作为一个子系统来进行测试。 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。 对于数据库管理系统 (DBMS)，还需要进行深入的研究，以确定可以支持以下测试的工具和技术。

|  |  |
| --- | --- |
| 测试目的： | 确保数据库访问方法和进程正常运行，没有数据损坏。 |
| 技术： | ·调用每个数据库访问方法和过程，为每个种子设定有效的种子 以及无效数据或数据请求。  ·检查数据库以确保数据具有已按预期填充，所有数据库事件正确发生，或查看返回的数据以确保检索到正确的数据出于正确的原因 |
| 完成 标准： | 所有数据库 访问方法和流程按设计运行，无需任何数据 腐败。 |
| 特殊 考虑： | ·测试可能需要DBMS开发 要进入的环境或驱动程序或 直接在数据库中修改数据。  ·应手动调用进程。  ·应使用最小大小的数据库（有限的记录数）来提高任何不可接受的事件。 |

3.1.2 功能测试

对测试对象的功能测试应侧重于所有可直接追踪到用例或业务功能和业务规则的测试需求。 这种测试的目标是核实数据的接受、处理和检索是否正确，以及业务规则的实施是否恰当。 此类测试基于黑盒技术，该技术通过图形用户界面 (GUI) 与应用程序进行交互，并对交互的输出或结果进行分析，以此来核实应用程序及其内部进程。

|  |  |
| --- | --- |
| 测试目标： | 确保ATM模拟软件的各个功能正常工作，包括导航、数据输入、处理和检索 |
| 技术： | 执行每个用例、用例流或函数，使用有效和无效数据验证以下内容：   * 使用有效数据时会出现预期结果。 * 使用无效数据时，将显示相应的错误或警告消息。 * 每个业务规则已正确应用。 |
| 完成标准： | * 所有已执行计划的测试。 * 所有已识别的缺陷均已解决。 |
| 特殊注意事项： | 确保以下项目或问题不会影响功能测试的实施和执行：   * 所有测试数据应仔细准备，包括模拟各种交易和金额。 * 在测试期间，要确保测试环境模拟了真实的ATM操作，包括交易确认、卡片读取等。 * 安全性测试将在另一个测试类型中执行，不包括在此功能测试中。 |

3.1.3 业务周期测试

业务周期测试应模拟在一段时间内对 <ATM模拟软件> 执行的活动。 应先确定一个时间段（例如一年），然后执行将在该时间段（一年内）发生的事务和活动。 这种测试包括所有的日、周和月周期，以及所有与日期相关的事件（如备忘录）。

|  |  |
| --- | --- |
| 测试目标 | 确保正确的测试目标和后台进程 根据所需的业务模型和时间表运行。 |
| 技术： | 测试将模拟几个商业周期执行以下操作：  ·用于测试目标功能测试的测试将被修改或增强，以增加次数 执行每个函数以模拟 指定时间段内的多个不同用户。  ·时间或日期敏感函数将使用有效和无效的日期或时间段执行。  ·定期发生的功能将在适当的时间执行或启动。  ·测试将包括使用有效和无效数据来验证以下内容：  ·这使用有效数据时会出现预期结果。  ·相应的错误或警告消息 在无效数据出现时显示 使用。  ·正确应用每个业务规则。 |
| 完成标准： | ·已执行所有计划的测试。  ·已识别的缺陷已得到解决。 |
| 特殊注意事项： | ·系统日期和活动可能需要特殊支持活动  ·需要模型来确定适当的测试要求和程序。 |

3.1.4 用户界面测试

用户界面 (UI) 测试用于核实用户与软件之间的交互。 UI 测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的访问或浏览功能。 另外，UI 测试还可确保 UI 中的对象按照预期的方式运行，并符合公司或行业的标准。

|  |  |
| --- | --- |
| 测试目标： | ·导航通过测试目标正确反映业务功能和需求，包括窗口到窗口，字段到字段和访问的使用方法（Tab 键、鼠标移动、快捷键）键）  ·窗口对象和特征（如菜单、大小、位置、状态和焦点）符合标准。 |
| 技术： | 为每个窗口创建或修改测试以 验证每个应用程序窗口的正确导航和对象状态，以及对象。 |
| 完成标准： | 每个窗口都已成功验证到与基准版本保持一致或在可接受的标准范围内 |
| 特殊注意事项： | 并非自定义和第三个属性的所有属性可以访问参与方对象。 |

3.1.5 性能测评

 性能评测是一种性能测试，它对响应时间、事务处理速率和其他与时间相关的需求进行评测和评估。 性能评测的目标是核实性能需求是否都已满足。

实施和执行性能评测的目的是将测试对象的性能行为当作条件（例如工作量或硬件配置）的一种函数来进行评测和微调。

[注： 以下所说的事务是指“逻辑业务事务”。 这种事务被定义为将由系统的某个 Actor 通过使用测试对象来执行的特定用例，例如，添加或修改给定的合同。]

|  |  |
| --- | --- |
| 测试目标： | ·正常预期工作负载  ·预期的最坏情况工作负载] |
| 技术： | ·使用为开发的测试程序功能或商业周期测试。  ·修改数据文件以增加数量的事务或脚本增加每个事务发生的迭代次数。  ·脚本应该在一台机器上运行 |
| 完成标准： | ·单个事务或单个用户：成功完成测试脚本，没有任何失败，并且在预期或所需的时间内完成每笔交易。  ·多笔交易或多笔交易用户：成功完成测试脚本，无失败，并在可接受的时间内。 |
| 特殊注意事项： | 全面的性能测试包括具有服务器上的后台工作负荷。  有几种方法可用于执行此操作，包括：  ·“推动交易”到服务器，通常采用结构化的形式查询语句（SQL）调用。  ·创建“虚拟”用户负载以模拟许多客户，通常为几百个。使用远程终端仿真工具 以完成此负载。此技术也可用于加载具有“流量”的网络。  ·使用多个物理客户端，每个客户端运行测试脚本以施加负载系统。  性能测试应在专用上执行机器或在专用时间。这允许完全控制和精确测量。  用于性能测试的数据库应为任何大小或比例相等。 |

3.1.6 访问控制测试

安全性和访问控制测试侧重于安全性的两个关键方面：

应用程序级别的安全性，包括对数据或业务功能的访问

系统级别的安全性，包括对系统的登录或远程访问。

应用程序级别的安全性可确保：在预期的安全性情况下，Actor只能访问特定的功能或用例，或者只能访问有限的数据。 例如，可能会允许所有人输入数据、创建新账户，但只有管理员才能删除这些数据或账户。如果具有数据级别的安全性，测试就可确保“用户类型一”能够看到所有客户消息（包括财务数据），而“用户二”只能看见同一客户的统计数据。

系统级别的安全性可确保只有具备系统访问权限的用户才能访问应用程序，而且只能通过相应的网关来访问。

|  |  |
| --- | --- |
| 测试目标： | ·         应用程序级安全性：[验证参与者是否可以访问仅提供其用户类型的函数或数据权限。  ·         系统级安全性：验证只有那些具有允许访问系统和应用程序以访问它们。 |
| 技术： | ·         应用程序级安全性：[标识并列出每个用户类型 以及每种类型具有权限的函数或数据。  ·为每种用户类型创建测试，并 通过创建来验证每个权限 特定于每种用户类型的事务。  ·修改用户类型并重新运行测试 用户。在每种情况下，请验证这些 附加功能或数据正确可用或被拒绝。  ·         系统级访问 |
| 完成标准： | 对于每个已知的参与者类型都有相应的功能或数据可用，并且所有事务按预期运行，并在先前的应用程序功能中运行测试。 |
| 特殊注意事项： | 访问必须与适当的网络或 系统管理员。此测试 可能不需要，因为它可能是网络或系统的功能管理。 |

3.1.7 安装测试

安装测试有两个目的。第一个目的是确保该软件在正常情况和异常情况的不同条件下——例如，进行首次安装、升级、完整的或自定义的安装——都能进行安装。异常情况包括磁盘空间不足、缺少目录创建权限等。第二个目的是核实软件在安装后可立即正常运行。这通常是指运行大量为功能测试制定的测试。

|  |  |
| --- | --- |
| 测试目标： | 验证测试目标是否正确安装到每个在以下条件下所需的硬件配置：  ·新安装，新机器，从不以前使用 <ATM模拟软件> 安装的  ·更新以前安装的机器<ATM模拟软件>，同版本  ·更新以前安装的机器<ATM模拟软件>，旧版本 |
| 技术： | ·手动或开发自动脚本，以验证条件目标计算机新建 - <ATM模拟软件>从未安装；<ATM模拟软件> 相同版本或旧版本已安装）。  ·启动或执行安装。  ·使用预定的函数子集 测试脚本，运行事务。 |
| 完成标准： | <ATM模拟软件>事务成功执行，无故障。 |
| 特殊注意事项： | 应该选择 <ATM模拟软件> 的哪些事务才能准确地测试出 <ATM模拟软件> 应用程序已经成功安装，而且没有遗漏主要的软件构件 |

**3.2 工具**

此项目将使用以下工具

|  |  |  |  |
| --- | --- | --- | --- |
|  | 工具 | 供应商/内部 | 版本 |
| 测试管理 | teambition | 阿里巴巴 |  |
| 缺陷跟踪 | BugZilla | Mozilla |  |
| 项目管理 | teambition | 阿里巴巴 |  |
| 数据库管理系统工具 | MySQL |  |  |

**4. 资源**

**4.1 角色**

此表显示了在此项目的人员配置方面所作的各种假设。

|  |  |  |
| --- | --- | --- |
| 人力资源 | | |
| 工人 | 最少资源 推荐  （分配的全职员工人数） | 具体职责 或评论 |
| 测试经理，  测试项目经理 | 1 | 提供管理监督。  责任：  ·         提供技术指导  ·         获取适当的资源  ·         提供管理报告 |
| 测试设计员 | 1 | 识别测试用例、确定测试用例的优先级并实现测试用例。  责任：  ·         生成测试计划  ·         生成测试模型  ·         评估测试的有效性 努力 |
| 测试员 |  | 执行测试。  责任：  ·         执行测试  ·         日志结果  ·         从错误中恢复  ·         文档更改请求 |
| 测试系统管理员 | 1 | 确保管理和维护测试环境和资产。  责任：  ·         管理测试管理系统  ·         安装和管理工作人员对 测试系统 |
| 数据库管理员、数据库管理器 | 2 | 确保管理测试数据（数据库）环境和资产 并维护。  责任：  ·         管理测试数据（数据库） |
| 设计员 | 2 | 标识和定义操作、属性和 测试类的关联。  责任：  ·         识别和定义测试 类  ·         识别和定义测试 包 |
| 实施员 | 2 | 实现和单元测试测试类和测试包。  责任：  ·         创建测试类并 在测试模型中实现的包 |

**4.2 系统**

以下表列出了测试项目的系统资源。

|  |  |
| --- | --- |
| 系统 资源 | |
| 资源 | 名字 /类型 |
| 数据库服务器 | MySQL |
| - 网络或子网 | 待定 |
| - 服务器名称 | 待定 |
| - 数据库名称 | 待定 |
| 客户端测试电脑 |  |
| — 包括特殊配置要求 | 待定 |
| 测试存储库 |  |
| - 网络或子网 | 待定 |
| - 服务器名称 | 待定 |
| 测试开发电脑 | 待定 |

**5. 项目里程碑**

 对 <ATM模拟软件> 的测试应包括上面各节所述的各项测试的测试活动。 应该为这些测试确定单独的项目里程碑，以通知项目的状态和成果。

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | 工作量 | **开始日期** | **结束日期** |
| 计划测试 | 7个工作日 | 2023-10-19 | 2023-10-25 |
| 设计测试 | 10个工作日 | 2023-10-25 | 2023-11-5 |
| 完善核心功能 | 2个工作日 | 2023-11-8 | 2023-11-9 |
| 进行单元测试以及集成测试 | 4个工作日 | 2023-11-10 | 2023-11-13 |
| 测试并修复缺陷 | 4个工作日 | 2023-11-17 | 2023-11-20 |
| 进行系统测试 | 3个工作日 | 2023-11-24 | 2023-11-26 |

**6. 可交付成果**

本节列出了将要创建的各种文档、工具和报告，及其创建人员、交付对象和交付时间。

**6.1 测试模型**

**创建人员：** 测试团队

**交付对象：** 项目管理团队、开发团队、相关利益相关者

**交付时间：** 在测试开始前

**描述：** 测试模型将包括测试计划、测试用例、测试数据和测试环境的详细描述。测试计划将定义测试的范围、目标、策略和计划，测试用例将描述各种测试场景和预期结果，测试数据将指定用于测试的数据集，测试环境将描述测试的硬件和软件环境。这些文档将用于指导测试活动和报告测试结果。

**6.2 测试记录**

**创建人员：** 测试团队

**交付对象：** 项目管理团队、开发团队、相关利益相关者

**交付时间：** 每次测试执行后

**描述：** 测试记录将说明用于记录和报告测试结果和测试状态的方法和工具。这包括记录测试用例的执行情况、错误报告、测试通过/失败状态、缺陷状态以及其他与测试相关的信息。测试记录将有助于跟踪测试进展和结果。

**6.3 缺陷报告**

**创建人员：** 测试团队

**交付对象：** 项目管理团队、开发团队、相关利益相关者

**交付时间：** 在发现缺陷后立即

**描述：** 缺陷报告将确定用来记录、跟踪和报告测试中发生的意外情况及其状态的方法和工具。每个缺陷报告将包括问题的描述、复现步骤、优先级、状态和分配给的责任人。缺陷报告将用于跟踪缺陷的解决过程，确保它们得到适当的处理和解决。

以上是有关可交付成果的描述，这些文档、工具和报告对于测试过程的有效管理和沟通至关重要，以确保测试的成功进行和问题的有效处理。

**7. 附录 A：项目任务**

以下是与测试相关的任务：

·计划测试

-          确定测试要求

-          评估风险

-          制定测试策略

-          确定测试资源

-          创建时间表

-          生成测试计划

·设计测试

- 识别和描述测试用例

- 识别和构建测试程序

-回顾 并评估测试覆盖率

·实施测试

- 记录或程序测试脚本

- 确定特定于测试的功能在设计和实现模型中

- 建立外部数据集

·执行测试

-执行测试程序

-评价测试的执行

-恢复从停止的测试

-验证结果

-探讨意外结果

-缺陷报告

·评估测试

-评价测试用例覆盖率

-评价代码覆盖率

-分析缺陷

-确定如果已达到测试完成标准和成功标准