<Memory>

测试计划

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2016/8 | 0.5 | 初始版本 | 许泽资 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 5

1.1 目的 5

1.2 背景 5

1.3 范围 5

1.4 项目标识 6

2. 测试需求 7

类型 7

描述 7

内容 7

功能性需求 7

登入与注册流程 7

资源创建浏览流程 7

图像上传浏览流程 7

推荐服务获取流程 7

音乐播放流程 7

非功能性需求 7

服务测试 7

模块测试 7

工具类测试 7

数据库完整性验证 7

图片匹配验证 7

音乐图书获取测试 7

3. 测试策略 8

3.1 测试类型 8

3.1.1 数据和数据库完整性测试 8

3.1.2 功能测试 8

3.1.3 用户界面测试 8

3.1.4 性能评价 9

3.1.5 安全性和访问控制测试 10

3.1.6 配置测试 11

3.1.7 安装测试 12

4. 项目里程碑 13

5. 可交付工件 14

3.1 测试日志 14

3.2 缺陷报告 14

测试计划

# 简介

## 目的

<Memory> 的这一“测试计划”文档有助于实现以下目标：

1. 随着项目的不断开发，一部分构件开发已经进入尾声。除了开发人员自测除外，还要进行有序的统一测试。包括登录、注册、资源、好友、图片等服务器端模块以及界面、服务获取等手机客户端模块都需要进行各种测试，来保证其可用性、可靠性，进而提出相应的解决方案，重新用于代码，保证在一定压力下的运行能力。
2. 为了保证快速有效统一的进行整个测试过程，推荐但不限于使用等价类的方式，以及强制产生所有错误信息的方法，来对每一个功能性函数进行测试，保证其可用性，并及时汇报所遇到的问题。对需求性测试，将整个过程进行模拟测试，检查整个系统是否按照预期产生结果，并给出相关的建议。
3. 压力测试可以采用ab等工具进行测试，模拟多用户同时使用操控某个资源，在保证资源正确的情况下，检查响应时间。
4. 安装测试将在远程服务器上进行简单测试，并给出说明文档与相关报告。
5. 整个测试工作大约需要2人 \* 5天

## 背景

1. 数据库的检验中，采用MySQL，以及附带的Workbench工具。

2. 服务器测试环境在tomcat8,windows10上，部署安装测试在linux上。

3. 以及其他常见的测试环境。

## 范围

1. 单元测试。所有的单元模块都要接受相关测试。

2. 集成测试。对服务器端，每个service进行测试；对客户端，每个功能模块接受测试。

3. 系统测试。对系统进行压力测试，和模块之间交叉进行的可用性测试；体验性测试。

4. 开发过程落后于计划，导致测试时间不充足，可能导致体验性测试时间缩短或取消。

5. 测试后的bug修复用时过久，导致落后于计划安排，可能影响体验性测试。

## 项目标识

下表列出了制定*测试计划*所用的文档，并标明了文档的可用性：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **文档 （版本/日期）** | **已创建或可用** | **已被接受或已经过复审** | **作者或来源** | **备注** |
| 需求规约 | 是 | 是 | memory项目组 |  |
| 用例报告 | 是 | 否 | memory项目组 |  |
| 项目计划 | 是 | 是 | memory项目组 |  |
| 设计规约 | 是 | 是 | memory项目组 |  |
| 业务模型或业务流程 | 是 | 是 | memory项目组 |  |
| 数据模型或数据流 | 是 | 是 | memory项目组 |  |

# 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

|  |  |  |
| --- | --- | --- |
| 类型 | 描述 | 内容 |
| 功能性需求 | 登入与注册流程 |  |
| 资源创建浏览流程 |  |
| 图像上传浏览流程 |  |
| 推荐服务获取流程 |  |
| 音乐播放流程 |  |
| 非功能性需求 | 服务测试 |  |
| 模块测试 |  |
| 工具类测试 |  |
| 数据库完整性验证 |  |
| 图片匹配验证 |  |
| 音乐图书获取测试 |  |

# 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。 |
| 方法： | • [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。  • 检查数据库，确保数据已按预期的方式填充，并且所有 数据库事件都按正常方式出现；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： | • 测试可能需要 DBMS 开发环境或驱动程序以便在数据库中直接 输入或修改数据。  • 进程应该以手工方式调用。  • 应使用小型或最小的数据库（其中的记录数很有限）来 使所有无法接受的事件具有更大的可见性。 |

### 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。 |
| 方法： | 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： | 暂无 |

### 用户界面测试

|  |  |
| --- | --- |
| 测试目标： | 核实以下内容：  • 通过浏览测试对象可正确反映业务的功能和需求，这种浏览包括 窗口与窗口之间的浏览，以及各种访问方法  • 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。 |
| 方法： | 为每个窗口进行细节性检测式体验，对每一个流程所感受到的问题，记录并进行汇报。保证风格一致。 |
| 完成标准： | 证实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： | 暂无 |

### 性能评价

|  |  |
| --- | --- |
| 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 |
| 方法： | • 使用为功能或业务周期测试制定的测试过程。  • 通过测试工具来增加事务数量，或通过修改脚本来增加每项 事务的并发程度。  • 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），并在多台客户机（虚拟的即可）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： | 事务预期的完成时间和相应时间在需求文档中可以找到 |

### 

### 

### 安全性和访问控制测试

|  |  |
| --- | --- |
| 测试目标： | 应用程序级别的安全性：核实主角只能访问其所属用户类型已被授权使用的那些功能或数据。  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主角才能访问系统和应用程序。 |
| 方法： | 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使用的功能或数据。  • 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： | 各种已知的主角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： | 暂无 |

### 

### 配置测试

|  |  |
| --- | --- |
| 测试目标： | 核实测试对象可在要求的硬件和软件配置中正常运行。 |
| 方法： | • 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，以模拟主角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： | 可以采用打开所有项目工程文档的方式给与压力。 |

### 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。以前从未安装的新计算机，包括linux，与windows系统下的安装 |
| 方法： | 按照安装文档给出的操作流程对服务器端进行加载和安装。之后进行功能性测试 |
| 完成标准： | 事务成功执行，没有出现任何故障。 |
| 需考虑的特殊事项： | 以一次注册登录，资源添加为一套流程检测为宜。 |

## 4. 项目里程碑

|  |  |  |  |
| --- | --- | --- | --- |
| **里程碑任务** | **工作量** | **开始日期** | **结束日期** |
| 制定测试计划 | 1-2天 | 9/1 | 9/3 |
| 设计测试 | 3-4 | 9/3 | 9/6 |
| 实施测试 | 3-4 | 9/6 | 9/9 |
| 评估测试 | 1-2 | 9/9 | 9/11 |

# 5. 可交付工件

## 测试日志

excel 文档

## 测试以及缺陷报告

缺陷报告文档

3.3 测试用例

测试用例文档

3.4 API相关文档

6. 附录 A：项目任务

以下是一些与测试有关的任务：

• 制定测试计划

* + 确定测试需求
  + 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

- 准备工作量分析文档

- 确定并说明测试用例

- 确定并结构化测试过程

- 复审和评估测试覆盖

• 实施测试

* + 记录或通过编程创建测试脚本
  + 确定设计与实施模型中的测试专用功能
  + 建立外部数据集

• 执行测试

- 执行测试过程

- 评估测试的执行情况

- 恢复暂停的测试

- 核实结果

- 调查意外结果

- 记录缺陷

• 评估测试

- 评估测试用例覆盖

- 评估代码覆盖

- 分析缺陷

- 确定是否达到了测试完成标准与成功标准