钱在路上跑APP

测试计划

文档属性

文档属性

| 项目 | 明细 |
| --- | --- |
| 文档标题 | 钱在路上跑APP测试计划 |
| 整理人 | 肖志强 |
| 创建日期 | 2020413 |
| 最后修改日期 | 2020426 |

修改记录

| 日期 | 整理人 | 版本 | 修改说明 |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

审核

| 姓名 | 认可版本 | 职位 | 日期 |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 4](#_Toc38701532)

[1.1 编写目的 4](#_Toc38701533)

[1.2 背景 4](#_Toc38701534)

[1.3 参考资料 4](#_Toc38701535)

[1.4 测试提交成果 4](#_Toc38701536)

[2. 测试需求 5](#_Toc38701537)

[2.1.1 测试功能模块： 5](#_Toc38701538)

[3. 测试策略 5](#_Toc38701539)

[3.1 测试类型 5](#_Toc38701540)

[3.1.1 功能测试 5](#_Toc38701541)

[3.1.2 界面/易用性测试 6](#_Toc38701542)

[3.1.3 BVT/回归测试 7](#_Toc38701543)

[3.2 版本发布策略 7](#_Toc38701544)

[3.3 阶段测试策略 7](#_Toc38701545)

[3.4 Bug管理策略 8](#_Toc38701546)

[3.4.1 缺陷管理工具 8](#_Toc38701547)

[3.4.2 错误优先级 8](#_Toc38701548)

[3.4.3 错误识别依据 8](#_Toc38701549)

[3.4.4 严重程度分类 8](#_Toc38701550)

[3.5 进度反馈策略： 9](#_Toc38701551)

[3.6 内部例会 9](#_Toc38701552)

[4. 测试环境 9](#_Toc38701553)

[5. 测试工具 9](#_Toc38701554)

[6. 通过准则 10](#_Toc38701555)

[7. 里程碑及人员分配 10](#_Toc38701556)

[细化测试任务 10](#_Toc38701557)

[8. 测试风险分析 11](#_Toc38701558)

[8.1 计划风险 11](#_Toc38701559)

# 引言

## 编写目的

针对钱在路上跑APP编写本次测试计划，本文档对具体后续测试工作安排进行规划，一方面使整个项目组明确测试进度、人员分配及主要职责等；另一方面该文档定义软件测试策略、方法、范围、进度、资源等，指导测试活动的进行，使测试组成员对具体工作有更清晰的了解，按照测试计划进行后期测试工作的开展。从根本上保证系统的切实可行性。

该计划阅读对象包括：测试人员、开发人员、项目其他人员。

## 背景

项目名称：钱在路上跑

项目提出者：河北师大软件学院市场部

开发人员：six项目组

测试人员：six测试组

项目目标：随着人们理财意识的觉醒，越来越多人重视自己的财务管理，钱在路上跑app为人们提供了便利的记账理财服务，记录人们的收入支出轨迹，使人们对自己各方面的收入支出一目了然，对自己的财产规划更条理清晰。

# 测试需求

本项目假设客户使用钱在路上跑app，重点进行功能、界面、易用性、兼容性及性能测试。

本阶段测试重点针对移动设备客户端进行测试。主要测试范围如下：

移动设备客户端功能：登录、用户主页、首页、搜索理财产品、测一测、查看测试结果、查看推荐产品、查看记账信息、记账、记账详情分析、查看我的所有记账情况；

功能性测试范围

### 测试功能模块：

|  |  |  |  |
| --- | --- | --- | --- |
| 移动设备客户端 | 模块 | 优先级 | 说明 |
| 登录 | 高 | 用户登录 |
| 用户主页 | 高 | 查看、编辑个人资料 |
| 首页 | 高 | 按模块分类进行模块跳转 |
| 搜索理财产品 | 中 | 搜索理财产品 |
| 测一测 | 中 | 根据自身支出习惯分析理财状况 |
| 查看测试结果 | 高 | 查看测试结果 |
| 查看推荐产品 | 低 | 查看推荐产品 |
| 查看记账信息 | 低 | 查看记账信息 |
| 记账 | 高 | 记账 |
| 记账详情分析 | 低 | 根据记账信息分析财务情况 |
| 查看我的所有记账情况 | 低 | 查看用户所有记账情况 |

# 测试策略

## 测试类型

主要进行功能测试，功能测试之后进行界面/易用性测试，测试的过程中迭代使用BVT/回归测试。

### 功能测试

**概述：**主要验证移动设备客户端的功能是否满足《音缘APP需求确认书.docx》中所规定的的功能性需求。

* **目标：**确保软件需求说明书中要求的各个功能模块全部按需求实现。

|  |  |
| --- | --- |
| **测试目标** | 所有功能测试需求项的功能实现。 |
| **测试方法和技术** | 按照测试需求、通过准则、测试用例，采用黑盒测试法，核实以下内容：  在使用合法数据时得到正确的结果，在使用非法数据时显示相应的错误信息或警告信息。  各个功能模块的功能都得到了正确的应用。 |
| **完成标准** | 计划的测试已全部执行。  发现的缺陷修复率达到通过准则要求。  不能实现的功能测试需求项开发组给出了合理的说明或作了需求变更。  所作修改是否已达到需求所定义的。 |
| **需考虑的特殊事项** | 客户端、中间件及数据库三开发组开发进度不同导致功能模块不完整，延缓测试进度。 |

### 界面/易用性测试

* **概述：**主要是使用户与软件之间的交互能够正常且简易地进行。
* **目标：**达到以下目标：
  + 确保各种浏览及各种访问方法都使用正常；
  + 确保各年龄段人群使用，尤其是要兼顾年龄大的用户；
  + 以符合标准和规范、直观性、一致性、灵活性、舒适性、正确性、使用性7要素为基础，作为界面及易用性测试的标准。

|  |  |
| --- | --- |
| **测试目标** | 查看界面排版、文字格式等美观性元素，简单快捷的支持用户操作等。  中间件使用WEB界面：简洁、操作流程清晰。  客户端JAVA编写：简单、易用、字体清晰。 |
| **测试方法和技术** | 根据整体架构设计及界面原型检验页面元素 |
| **完成标准** | 计划的测试已全部执行。  发现的缺陷修复率达到通过准则要求。 |
| **需考虑的特殊事项** | 兼顾年龄较大用户的使用。 |

### BVT/回归测试

* **概述：**每当软件发生变化时，必须重新测试原来已经通过测试的区域，验证修改的正确性及其影响。
* **目标：**验证修改后的BUG，查看是否影响了其他的功能。

|  |  |
| --- | --- |
| **测试目标** | 验证修改后的BUG,原先的BUG是否已经修复,并且查看是否影响其他的功能流程。 |
| **测试方法和技术** | 主要是验证前一版本提交的BUG,按照提交BUG时给定的数据在最新版本上进行操作验证。 |
| **完成标准** | 修复的BUG得到预先需求的确认。 |
| **需考虑的特殊事项** |  |

## 版本发布策略

原则1：当进行首轮测试时，若系统主干功能不能通过冒烟测试，则需要开发组重新发布版本，再对新版本进行首轮测试

原则2：遵循每日构建原则。每日构建工作由测试团队负责，每日发布新的测试版本（Build）并对其进行BVT测试，BVT测试通过后针对该Build进行细测。要求：每个成功的Build都应该通过BVT测试。

原则3：每当完成修复就建立新版本，并进行BVT测试，如果未通过BVT测试（仍存在bug过多或bug级别严重），则可要求重新发布版本，进行第二次BVT测试。

## 阶段测试策略

针对实际项目情况，测试阶段分为：

1、**单元测试阶段：**由开发人员针对个人负责的单元或模块进行单元测试。通过本阶段后进行下一阶段。

2、**冒烟测试阶段：**验证系统第一版本的关键功能，目的验证该系统版本是否可用，是否能进行具体功能细测，若出现过多限制后续测试的阻塞级别bug，则需要请开发组发布新版本。

3、**初测/细测阶段：**重点验证软件功能是否满足需求，该阶段由测试人员完成。测试成员对功能点进行独立测试，并编写简易的测试用例。同时，在该用例执行中记录该用例执行的状态（通过 / 未执行 / bug ID）

4、**BVT测试阶段：**针对新Build进行版本功能验证，通过后方可进行新功能点的细测阶段。

5、**回归测试阶段：**重点在于验证bug是否解决及相关功能是否受影响。

## Bug管理策略

### 缺陷管理工具

采用Redmine缺陷管理系统进行缺陷实时提交和跟踪。

Redmine缺陷管理系统地址：搭建未完成，需另行通知

Redmine登录名：个人的姓名全拼 如：刘远铎的用户名为 liuyuanduo

Redmine默认密码：个人邮箱密码

### 错误优先级

立刻级别：必须立即修改；

紧急级别：立即修改（最长时间不得超过2天）；

高级级别：在投入正式运行前修改；

普通级别：在投入正式运行前大部分需修改，未修改的缺陷需进行讨论；

低级别：可延期修改或不修改。

### 错误识别依据

1. 需求文档
2. 技术规格说明

### 严重程度分类

|  |  |
| --- | --- |
| **严重程度** | **现象** |
| 严重 | ①   用户需求未实现（影响到用户完成业务）；  ②   用户需求实现错误（影响到用户完成业务）；  ③   导致被测软件响应明显很慢（假死）、死机、非法退出、崩溃；  ④   导致后台数据受损或丢失 |
| 中等 | ①用户需求实现错误（不影响用户完成业务、用户使用不频繁)；  ② 用户操作过程中系统出现异常报错，但不影响系统功能的使用。  ③用户使用不频繁的功能，响应时间超出忍耐限度；  ④ UI上存在错误引导用户的信息。 |
| 轻微 | ①   UI控件不符合界面规范。  ②   影响UI友好性 |

## 进度反馈策略：

1. 项目组内部每天以日报的形式沟通当日工作内容、进展、计划
2. 测试全部完成后，由测试管理人员向开发组反馈测试整体情况

## 内部例会

1. 执行测试阶段开始后，测试组每天上班前进行测试的作战会议

# 测试环境

|  |
| --- |
| **软件环境（相关软件、操作系统等）** |
| 服务器：Windows Server 2003+SQL Server 2005 +office excel+Excel Server |
| 客户端：Andriod（手机），JDK+Eclipes（pc） |
| 中间件：JDK+Eclipes |
| **硬件环境（网络、设备等）** |
| 测试服务器：尽量模拟真实运行环境 |
| 客户端：单元测试阶段使用PC的手机模拟程序，系统测试阶段使用安卓手机 |

# 测试工具

|  |  |  |  |
| --- | --- | --- | --- |
| **用途** | **工具** | **生产厂商/自产** | **版本** |
| 缺陷跟踪 | Redmine |  |  |
| 测试用例 | Test link |  |  |

# 通过准则

1. 所有的测试类型都执行并达到完成标准。
2. 需求覆盖率达到100%。
3. 立刻\紧急\高级别的错误修复率达到100%。
4. 普通级错误的修复率达到90%以上。
5. 严重、中等缺陷修复率100%，轻微类型缺陷个数控制在5个。

# 里程碑及人员分配

|  |  |  |  |
| --- | --- | --- | --- |
| **阶段** | **区间测试任务** | **阶段完成标志** | **人员** |
| 测试计划  （3月29日-4月1号） | 阅读需求，编写测试计划，分派测试任务。 | 提交《测试计划》并通过评审 | 测试员1 |
| 设计阶段  （4月2日—4月9日） | 熟悉系统需求并设计测试用例；测试中同步细化、更新用例。 | 提交《测试用例设计》 | 测试员2 |
| 执行功能测试  （4月10日—5月21日） | 客户端功能测试，参照开发进度开展；同步进行客户端与中间件交互测试；同步进行BVT及回归测试。 | 达到通过准则 | 测试员1 |
| 测试总结报告编写  （5月22日-5月25日） |  | 提交《测试总结报告》 | 测试员2 |

# 细化测试任务

|  |  |  |
| --- | --- | --- |
| **阶段** | **功能模块分配** | **人员** |
| 设计阶段  （4月2日—5月9日） |  | 测试员1、测试员2 |
| 执行功能测试  （4月10日—4月21日） | 用户模块：测试员1  商家模块：测试员2  管理员模块：测试员1、测试员2 | 测试员1、测试员2 |
| 测试总结报告编写  （4月22日-4月26日） |  | 测试员1、测试员2 |

# 测试风险分析

## 计划风险

1. 项目进度较紧张且项目组各成员工作安排紧密，在任务提交及时性上存在风险
2. 项目组开发进度不吻合，沟通上容易存在问题
3. 项目进度较难，项目组成员任务紧张
4. 项目bug较多，测试任务繁重