<快借宝>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 6/9/2018 | 1.0 | 初稿 | 宋博仪 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 4

4.1 需求覆盖率及缺陷分布 4

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

5.2 非功能性缺陷 6

6. 测试结论与建议 6

测试报告

# 简介

## 目的

测试代码，确保其表现同预期一致

## 范围

1. 个人信息管理模块中，用户可以注册登录，之后可以用户名进行借款、还款或出借，完成业务后可以注销操作。
2. 我要借款模块中，用户可以进行借款申请，查看欠款，还款操作。
3. 我的债权模块中，用户可以进行借出钱款，查看债权，购买理财产品，查看已购产品，转让债权操作。
4. 个人安全管理模块中，用户可以进行绑定手机，绑定邮箱，找回密码操作。

## 定义、首字母缩写词和缩略语

|  |  |  |
| --- | --- | --- |
| Abbreviations缩略语 | Full spelling 英文全名 | Chinese explanation 中文解释 |
| API | Application Programming Interface | 应用程序编程接口 |
| JDK | Java Development Kit | Java 软件开发工具包 |
| ANTD | ANT Design | React UI组件库 |
| P2P | person to person | 单人对单人 |

## 参考资料

《快借宝项目软件需求规约》、《快借宝项目测试用例》。

## 概述

系统测试报告，说明了软件测试的执行情况和软件质量，并分析缺陷原因。

# 测试概要

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 天数 | 开始时间 | 结束时间 | 人员 |
| 个人信息 | 0.16 | 2018-7-14 | 2018-7-14 | 宋博仪 |
| 借款 | 0.17 | 2018-7-14 | 2018-7-14 | 宋博仪 |
| 理财 | 0.17 | 2018-7-14 | 2018-7-14 | 宋博仪 |
| 信用管理 | 0.20 | 2018-7-24 | 2018-7-24 | 宋博仪 |
| 数据统计 | 0.20 | 2018-7-24 | 2018-7-24 | 宋博仪 |
| 在线咨询 | 0.10 | 2018-7-24 | 2018-7-24 | 宋博仪 |

# 测试环境

应用服务器配置：

CPU：Inter Cel430

ROM:1G

OS: Windows XP SP4

DB：SQL Server 2000

客户端：IE7.0

# 测试结果及分析

## 需求覆盖率及缺陷分布

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **用户注册** | 4 | 100% | 0 | 0 |  |
| **用户登录** | 3 | 100% | 0 | 0 |  |
| **银行卡** | 3 | 100% | 0 | 0 |  |
| **个人信息** | 1 | 100% | 0 | 0 |  |
| **信用额度** | 4 | 100% | 0 | 0 |  |
| **借款管理** | 5 | 100% | 0 | 0 |  |
| **数据统计** | 5 | 100% | 0 | 0 |  |
| **理财商城** | 5 | 100% | 0 | 0 |  |
| **帮助** | 2 | 100% | 0 | 0 |  |
| **出借** | 3 | 100% | 0 | 0 |  |
| **债权管理** | 1 | 100% | 0 | 0 |  |
| **理财产品管理** | 1 | 100% | 0 | 0 |  |
| **功能项小计** | 37 | 100% | 0 | 0 |  |
| **非功**  **能项** | **性能** | 1 | 100% | 0 | 0 |  |
| **部署** | 1 | 100% | 0 | 0 |  |
| **可靠性** | 1 | 100% | 0 | 0 |  |
|  | **非功能项小计** | 3 | 100% | 0 | 0 |  |
|  | **总计** | 40 | 100% | 0 | 0 |  |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 0 | 0 | 0 | 0 | 0 |
| **占缺陷百分比** | 0 | 0 | 0 | 0 | 0 |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | NULL |  |  |  |  |  |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

**表4 非功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **测试类型** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | NULL |  | 性能 |  |  |  |

# 测试结论与建议

应当有专门的、不参与代码编写的人员做测试，不然难以客观严密。