第一章引言

1.1项目背景

自2007年，苹果推出了第一代iPhone，智能手机开始真正走向市场。并凭借其人性化，功能强，速度快的特点飞速发展和普及。随之而来的是移动应用出现在我们生活的方方面面，从日常社交，新闻浏览，运动健身到购物付款，移动应用保罗万象，甚至南大宿舍洗衣机也需要U净这款app使用。现代人一机在手，行走天下，越来越离不开手机，离不开里面覆盖生活方法面面的应用。据工信部发布2017年上半年我国互联网业务运行情况报告显示国内移动应用程序市场持续活跃且移动互联网应用数量已超402万款，其中游戏达116万款。

然而在这高速发展的移动应用市场却也存在着诸多隐患，2016年CNCERT/CC捕获及通过厂商交换获得的移动互联网恶意程序样本数量为2,053,501个。而Android平台用户成为最主要的攻击对象。恶意程序针对Android平台的恶意程序共有2,053,450个，占99.9%以上，位居第一。应用安全正成为用户越来越关注的点。另外移动应用的使用场景比传统pc端应用或网页应用更加复杂，同一个移动应用的使用环境包括不同版本的不同操作系统，不同的硬件设备，不同的网络环境。

所以对于开发者而言，确保移动应用在复杂环境中有效运行无疑是一个艰巨的任务。而测试无疑是质量安全的必要保证。但移动应用环境的复杂带来了移动应用测试的复杂，对于开发者而已，保证测试设备完备，测试人员专业必然是困难的事情。面对这样的难题，一个提供大量只有多样手机用户的专业的测试平台则是帮助开发人员的法宝。这就是群智测试工具兴起的必然。

1.2群智测试工具以及现状

1.2.1群智测试工具

1.2.1bugtags

1.2.2班墨云测试

1.2.2群智测试现状

1.3项目功能概述

1.4论文的主要工作和组织结构

第二章技术概述

第三章系统需求分析与概要设计

3.1BugHunter项目整体概述

群智安卓测试BugHunter系统是一个面向安卓测试的移动应用群智测试平台，它的主要目标是为安卓测试经理提供bug生命周期的管理功能，为测试人员提交bug高效准确的上报功能。

BugHunter系统通过web前端安卓测试经理不仅提供app和bug的管理功能，还提供了丰富的bug信息统计图表的展示，更便于分析app的开发情况。

通过开发安卓测试SDK，应用开发人员可以方便的将我们提供的静态库添加到其app中添加bug的反馈和提交功能，之后测试人员可以很方便的使用我们提供的悬浮按钮查看当前页面存在的bug和提交当前页面bug，当app崩溃时将自动上传bug信息。

本人则负责用spring-boot+mysql为基础开发server端，为app提供bug上报途径，为web端提供app和bug的信息筛选和管理功能。

3.2系统需求分析

3.2.1用例图

为了更形象的阐述需求，将系统的功能以用例图和用例描述的方式展示如下图3.1所示：

用例图由角色，系统边界和用例组成，它可以帮助我们更好地描述系统内外交互。系统的涉众有测试人员和测试经理两类。测试人员借助系统完成测试任务上报bug。测试经理则研究系统收集到的数据，分析应用开发状况。

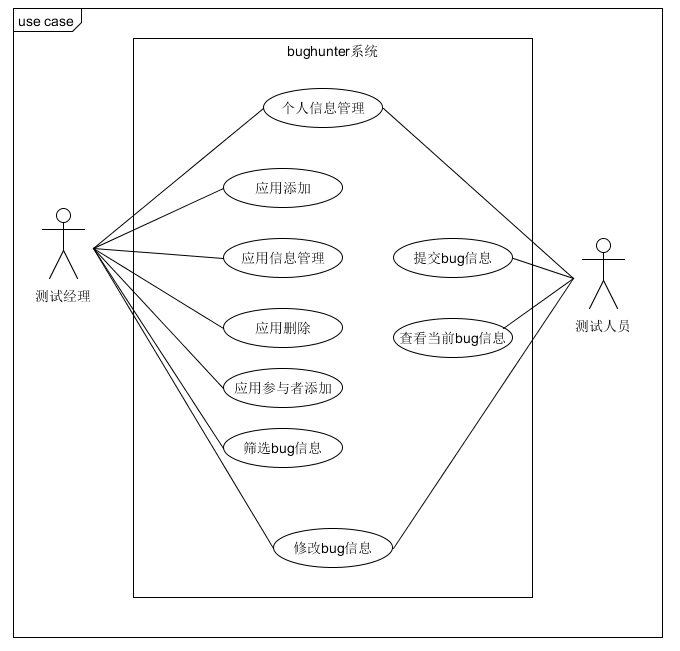


表3.1 用例图

3.2.2用例描述

基于以上用例图编写用例描述如下列表所示：

1）个人信息管理

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc01 |
| 名称 | 个人信息管理 |
| 参与者 | 用户 |
| 触发条件 | 用户需求管理个人信息 |
| 前置条件 | 用户已登陆成功 |
| 后置条件 | 用户信息按用户的需要做出修改 |
| 正常流程 | 1.用户进入个人信息页面  2.系统返回用户个人信息  3.用户修改个人信息，并点击确定  4.系统将用户个人信息修改情况反馈用户 |
| 扩展流程 | 无 |
| 特殊需求 | 无 |

表3.2 个人信息管理用例

2）应用添加

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc02 |
| 名称 | 应用添加 |
| 参与者 | 测试经理 |
| 触发条件 | 测试经理添加应用到账户 |
| 前置条件 | 测试经理已经登陆成功 |
| 后置条件 | 系统将应用信息添加到数据库 |
| 正常流程 | 1.测试经理进入应用添加页面，并且输入完整应用信息，点击添加  2.系统自动生成AppKey和AppSecret，将应用信息添加到数据库，返回AppKey和AppSecret |
| 扩展流程 | 无 |
| 特殊需求 | 1.AppKey应为app主键，确保唯一性  2.AppSecret为app验证信息，需难以破解，确保安全性 |

表3.3 应用添加用例

3）应用信息管理

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc03 |
| 名称 | 应用信息管理 |
| 参与者 | 测试经理 |
| 触发条件 | 应用信息需要被查看或修改 |
| 前置条件 | 测试经理已经登陆成功 |
| 后置条件 | 系统将应用信息修改提交到数据库 |
| 正常流程 | 1.测试经理进入应用信息页面  2.系统返回应用信息  3.测试经理修改应用信息，点击确定  4.系统按需要修改应用信息，返回修改结果 |
| 扩展流程 | 无 |
| 特殊需求 | 无 |

表3.4 应用信息修改用例

4）应用删除

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc04 |
| 名称 | 应用删除 |
| 参与者 | 测试经理 |
| 触发条件 | 应用已经被放弃，需要删除 |
| 前置条件 | 测试经理已经登陆 |
| 后置条件 | 应用被从数据库删除，包括应用所含bug |
| 正常流程 | 1.测试经理进入应用页面，点击删除  2.系统删除数据库中包含的应用及其bug信息，返回结果 |
| 扩展流程 | 无 |
| 特殊需求 | 无 |

表3.5 应用删除用例

5）应用参与者添加

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc05 |
| 名称 | 应用参与者添加 |
| 参与者 | 测试经理 |
| 触发条件 | 有人员需要被添加应用管理权限 |
| 前置条件 | 该应用测试经理已经登陆 |
| 后置条件 | 选择的用户被添加到应用中 |
| 正常流程 | 1.测试经理进入应用成员添加页面，输入添加人员邮箱，成员类型，点击确定  2.系统将该成员添加至应用，返回结果 |
| 扩展流程 | 2a.该成员邮箱不存在  2a1.系统返回该成员不存在  2b.该测试经理无添加成员权限  2b1.系统返回权限不足信息 |
| 特殊需求 | 1.仅应用创建者有权限添加成员 |

表3.6 应用参与者添加用例

6）筛选bug信息

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc06 |
| 名称 | 筛选bug信息 |
| 参与者 | 测试经理 |
| 触发条件 | 测试经理筛选bug信息，分析应用开发状况 |
| 前置条件 | 测试经理已经登陆 |
| 后置条件 | 无 |
| 正常流程 | 1.测试经理点击进入bug展示页面  2.系统返回该应用下所有bug用于展示  3.测试经理点选筛选条件  4.系统根据筛选条件返回所需bug信息 |
| 扩展流程 | 无 |
| 特殊需求 | 1.筛选条件为app版本和bug类型，状态，优先级 |

表3.7 筛选bug信息用例

7）修改bug信息

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc07 |
| 名称 | 修改bug信息 |
| 参与者 | 测试经理或测试人员 |
| 触发条件 | bug信息不够完善，或bug状态改变需要修改信息 |
| 前置条件 | 用户已登录 |
| 后置条件 | 系统按需要修改bug信息并记录修改历史 |
| 正常流程 | 1.用户已进入bug信息页面，做出修改，点击确定  2.系统按需要修改bug信息并记录修改历史后返回结果 |
| 扩展流程 | 无 |
| 特殊需求 | 1.bug信息用户仅可以修改描述、状态、优先级、类型 |

表3.8 修改bug信息用例

8）提交bug信息

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc08 |
| 名称 | 提价bug信息 |
| 参与者 | 测试人员 |
| 触发条件 | 测试人员发现bug或应用崩溃 |
| 前置条件 | 测试人员登陆成功或测试人员仅使用而未登陆均可 |
| 后置条件 | 系统将测试人员提交的bug保存到数据库 |
| 正常流程 | 1.测试人员发现bug并且填写提交信息完毕后点击确定，或应用在使用中崩溃自动提交bug。  2.系统记录提交的bug到数据库，返回结果 |
| 扩展流程 | 2a.该系统的版本未被记录  2a1.系统记录该系统版本 |
| 特殊需求 | 1.用户仅可填写bug描述、状态、优先级、类型，其他信息由SDK自动提取 |

表3.9 提交bug信息用例

9）查看当前bug信息

|  |  |
| --- | --- |
| 项目 | 内容描述 |
| ID | uc09 |
| 名称 | 查看当前bug信息 |
| 参与者 | 测试人员 |
| 触发条件 | 测试人员进入app行的activity |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 正常流程 | 1.测试人员点击应用使应用进入新的页面  2.系统返回当前页面已存在的bug信息 |
| 扩展流程 | 无 |
| 特殊需求 | 无 |

表3.9 提交bug信息用例

3.2.3非功能需求描述

系统的非功能需求如表3.10，要求系统具有良好的可靠性、易用性、可维护性和可移植性，同时，系统要求在较短时间内对用户操作做出响应。

|  |  |
| --- | --- |
| 需求类别 | 需求描述 |
| 效率 | 对于所有操作，系统应及时作出响应，响应时间应在0.5秒之内 |
| 可维护性 | 系统设计应尽可能合理，容易维护和扩展新模块 |
| 安全 | 系统需要验证操作是否合法，并拒绝非法操作 |
|  |  |

表3.10 非功能需求

系统所受的约束见表3.11，系统需要在有网络的环境下运行。

|  |  |
| --- | --- |
| 约束类别 | 约束描述 |
| 开发约束 | 系统服务器要求使用spring-boot+mysql的方式开发 |
| 部署环境 | 该系统服务器端将被部署到阿里云服务器ECS上，配置为1核，2GB，系统使用CentOS 7.3 64位 |
| 网络约束 | 系统必须在有网络环境下运行 |
| 用户约束 | 测试人员可不经过登陆使用系统，但须提供AppKey和APPSecret |

表3.11 约束

3.3系统概要设计

3.3.1概念类图

3.3.2体系结构设计-物理视图（Physical View）

3.3.3体系结构设计-开发视图（Development View）

3.4本章小结

第四章BugHunter详细设计与实现

1项目子模块概述

2子模块详细设计

第五章总结与展望