

EZ-Shopping APP for Android

测试计划

项目实训 第 5 组

2017/7/11

目 录

1. 引言	3
1.1 产品背景	3
1.2 文件专业术语和缩略词	3
1.3 参考文档	5
2. 目标测试对象	6
2.1 测试包括	6
2.2 测试排除	6
3. 测试策略	6
3.1 测试说明	6
3.1.1 测试描述	6
3.1.2 测试完成标准	7
3.1.3 影响分析结果	7
3.2 测试安排	8
3.2.1 安装测试	8
3.2.2 功能测试	8
3.2.3 回归测试	9
3.2.4 性能测试	10
3.2.5 集成测试	10
3.3 缺陷管理	11
3.3.1 缺陷描述	11
3.3.2 缺陷管理工具	11
3.4 测试度量	11
3.4.1 测试用例设计阶段的度量	11
3.4.2 测试执行阶段的度量	12
3.5 测试报告	12
4. 进入和退出条件	12
4.1 测试准入标准	12
4.2 测试准出标准	13

4.3 软件测试暂停和恢复标准	13
4.3.1 软件测试暂停标准	13
4.3.2 软件测试恢复标准	13
5. 环境需求	14
5.1 系统硬件最低要求	14
5.2 软件测试环境	14
6. 职责、人员和必需的培训	14
6.1 小组成员角色	14
6.2 人员和培训	15
7. 关键项目/ 阶段性里程碑	15
8. 风险、依赖、假设和限制	16
8.1 风险	16
8.2 假设和依赖	16
8.3 限制	17

1. 引言

1.1 产品背景

本次所要测试的产品为安卓平台的 **EZ-Shopping APP**。这是一个基于物联网（IoT）的微支付方案的移动端实现。核心产品为小型硬件 **EZS(EZ Shopping)**，外形和 Amazon 的 **AKS** 类似，是一款带着网络模块的按钮，在使用之前需要通过手机对它进行配置，使之与某件固定商品进行绑定。

在手机端 **APP** 登录之后，进行银行卡绑定和送货地址确定的操作，这是按钮绑定的前提操作。完成之后就可以点击绑定 **EZS** 按钮，手机会显示扫描条形码界面，用户即可对想绑定的商品进行条形码扫描，获取商品信息并确认后将 **EZS** 靠近手机背壳通过 **NFC** 技术进行绑定，**EZS** 中会存储商品的信息和用户的地址信息。在需要购买时，点击按钮即可，并且通过按钮边上的指示灯反馈购买是否成功。下单成功后买方和卖方都可以在自己的平台上查询到该订单并了解订单状态；若下单失败买方可以在手机 **APP** 上查询到下单失败原因。卖家在收到订单后可以发货，并在卖家客户端对订单进行已发货标记，在货物送到后买家可进行确认收货操作，在规定时间内未进行操作会由系统自动确认，确认收货后本订单状态即为已完成。

1.2 文件专业术语和缩略词

术语/英文缩写	中文解释
Code Inspection	代码审查
Review	技术评审
Execution-Based Testing	动态测试
White-Box Testing	白盒测试
Black-Box Testing	黑盒测试
Gray-Box Testing	灰盒测试

Software Quality Assurance	软件质量保证 SQA
Software Development Life Cycle	软件开发生命周期
Smoke Test	冒烟测试
Regression Test	回归测试
Function Testing	功能测试
Performance Testing	性能测试
Stress Testing	压力测试
Volume Testing	负载测试
Usability Testing	易用性测试
Installation Testing	安装测试
UI Testing	界面测试
Configuration Testing	配置测试
Documentation Testing	文档测试
Compatibility Testing	兼容性测试
Security Testing	安全性测试
Recovery Testing	恢复测试
Unit Test	单元测试
Integration Test	集成测试
System Test	系统测试
Acceptance Test	验收测试
The Test Objectives	测试对象
The Test Scope	测试范围
The Test Strategy	测试策略
The Test Approach	测试方法
The test procedures	测试过程
The Test Environment	测试环境
The test Completion criteria	测试完成标准
The Test Cases	测试用例

The Test Schedules	测试进度表
The End User	最终用户
IT (integration testing)	集成测试
ST (system testing)	系统测试
KPA (key process area)	关键过程域
PR (peer review)	同行评审
UAT (user acceptance testing)	用户验收测试
V&V (verification & validation)	验证和确认
CCB (change control board)	控制变更委员会
GUI (graphic user interface)	图形用户界面
CMO (configuration management officer)	配置管理员
MTBF (mean time between failures)	平均失效间隔时间
MTTR (mean time to restoration)	平均修复时间
MTTF (mean time to failure)	平均失效时间

1.3 参考文档

1. 《项目实训案例 170704》
2. 《G05-PC 项目章程-20170704》
3. 《G05-PP 项目总体计划-20170704》
4. 《G05-SRS 软件需求规格说明书-20170707》
5. 《G05-SDD 软件设计说明书-20170709》

2. 目标测试对象

2.1 测试包括

目标测试对象	通过标准	备注
需求覆盖	100%	详见《G05-SRS 软件需求规格说明书-20170707》
测试用例的执行	100%	包括安装测试、功能测试、回归测试、性能测试和集成测试五个部分的用例
测试用例的通过率	100%	所有用例都符合完成标准
漏洞修复率	95%	在之后的版本中会得到修复

2.2 测试排除

压力测试：在计算机数量较少或系统资源匮乏的条件下运行测试。

3. 测试策略

3.1 测试说明

3.1.1 测试描述

本次测试分为安装测试、功能测试、回归测试、性能测试和集成测试五个部分，相关描述及优先级如下：

测试策略	描述	优先级
安装测试	检查 APP 是否能够正常安装，运行，卸载	高
功能测试	检查各功能模块是否符合需求，有无遗漏功能，实现正常	高

回归测试	检查修改了旧代码后是否引入新的错误或导致其他代码产生错误	低
性能测试	检查是否能承载一定的压力，且响应及时	中
集成测试	检查各个模块组合起来，能否达到预期要求的功能； 检查一个模块的功能是否会对另一个模块的功能产生不利的影响； 检查全局数据结构是否有问题，会不会被异常修改	高

3.1.2 测试完成标准

最终通过所有的安装测试、功能测试、回归测试、性能测试和集成测试，系统无业务逻辑错误和二级 BUG。经确定的所有缺陷都已得到了商定的解决结果。所设计的测试用例已全部重新执行，已知的所有缺陷都已按照商定的方式进行了处理，而且没有发现新的缺陷。

3.1.3 影响分析结果

影 响 程 度	描 述	响应时间
高	影响正常的用户操作，导致系统崩溃； 严重影响系统流程； 数据库链接异常中断； 故意留有程序后门； 可能有灾难性后果； 产生错误结果，导致系统不稳定； 主要功能未能实现	缺陷提交到缺陷管理系统后，立即修改
中	功能实现但与需求不一致影响到其他模块； 数据库建库脚本错误，数据遗漏或丢失影响软件正常运行；	优先修改

	按钮设计遗漏，包括不同条件下的显示内容； 日期或时间初始值设置错误； 特殊字符处理错误； 页面输入限制错误，包括长度，字符限制	
低	界面设计风格不统一，相同的问题提示信息不一致； 文字不对齐，页面排列项不一致； 错误定位及信息提示不准确； UI 错误，包括字体错误，模板显示错误； 按钮或标签上有拼写错误，大小写错误	尽快在其 他空余时 间修改

3.2 测试安排

3.2.1 安装测试

测试目标	测试该 APP 在不同的安卓系统手机上是否能够正常运行
测试范围	安卓系统手机
技术	在不同安卓系统的手机上安装，运行，卸载
开始标准	集成测试完成后进行
完成标准	在不同安卓系统的手机上都能正常安装使用
测试重点	安卓手机，安装，运行
优先级	优先级高的，如：与其他 APP 不兼容
需考虑的特殊事项	确保安装有所需要的软件，确保人员到位，确保没有硬件故障
暂停原则	版本更新或升级

3.2.2 功能测试

测试目标	确保 APP 各模块功能正常，其中包括买家注册登录、基本信
------	-------------------------------

	息、管理订单、管理按钮，卖家发货，管理订单与商品
测试范围	全部功能都进行测试
技术	设计数据来执行各个用例、用例流或功能，以确保使用有效数据时得到预期的结果，使用无效数据时显示相应的错误或警告消息，各业务规则都得到了正确的应用
开始标准	编码完成后进行
完成标准	测试用例执行完毕，bug 数量减小，趋于稳定，所有功能都能得到正确验证
测试重点	用户下单模块
优先级	优先级高的，如：APP 崩溃、闪退，核心功能没有实现
需考虑的特殊事项	确保测试环境搭建正确，确保人员到位，规范执行测试用例
暂停原则	模块未完成开发

3.2.3 回归测试

测试目标	检查修改了旧代码后是否引入新的错误或导致其他代码产生错误
测试范围	整个安卓 APP
技术	GTRT、BRRT、BORT、BIRT
开始标准	每当被测试的软件或其环境改变时进行回归测试
完成标准：	95%的测试用例执行通过并通过系统测试
测试重点	用户下单模块
优先级	优先级高的，如：APP 崩溃、闪退，核心功能没有实现
需考虑的特殊事项	确保测试环境搭建正确，确保人员到位，规范执行测试用例
暂停原则	一轮测试完成

3.2.4 性能测试

测试目标	检查是否能承载一定的压力，且响应及时
测试范围	核实 APP 的安全、启动时间，CPU 占用，内存占用，流量耗用，电量耗用等
技术	使用自动化云测试工具，如 testin，爱内测
开始标准	自动化测试脚本设计并评审通过且项目组移交系统测试
完成标准	系统满足用户需求中所要求的性能要求
测试重点	APP 启动时间，CPU 占用
优先级	优先级高的，如：APP 崩溃、闪退，核心功能没有实现
需考虑的特殊事项	确保测试环境搭建正确，确保人员到位，规范执行测试用例
暂停原则	系统满足用户需求中所要求的性能要求

3.2.5 集成测试

测试目标	验证各个模块组合起来，能否达到预期要求的功能
	验证一个模块的功能是否会对另一个模块的功能产生不利的影响
	查看全局数据结构是否有问题，会不会被异常修改
测试范围	整个安卓 APP
技术	等价类划分，边界值分析，因果图，正交试验法，场景法，状态图法
开始标准	编码完成后进行
完成标准	测试用例执行完毕，bug 数量减小，趋于稳定
测试重点	用户下单模块
优先级	优先级高的，如：APP 崩溃、闪退，核心功能没有实现
需考虑的特殊事项	确保测试环境搭建正确，确保人员到位，规范执行测试用例

暂停原则	模块未开发完成
------	---------

3.3 缺陷管理

3.3.1 缺陷描述

对缺陷的描述应该包含：

- （1）缺陷 ID。唯一的缺陷 ID，可以根据 ID 追踪缺陷；
- （2）缺陷状态。常见状态有新建、待解决、已解决和已修复；
- （3）缺陷标题；
- （4）缺陷的详细描述。对缺陷的详细描述，包括缺陷如何复现的步骤等；
- （5）缺陷的严重程度。一般分为致命、严重、一般、细微四中；
- （6）缺陷的紧急程度。从 1-4，1 是优先级最高的等级，4 是优先级最低的等级；
- （7）缺陷提交人；
- （8）缺陷提交时间；
- （9）缺陷所属项目/模块；
- （10）缺陷解决人；
- （11）缺陷处理时间。

3.3.2 缺陷管理工具

本次测试选取的缺陷管理工具为 Bugzilla，它是一个开源的缺陷跟踪系统，可以管理软件开发中缺陷的提交、修复、关闭等整个生命周期。

3.4 测试度量

3.4.1 测试用例设计阶段的度量

- （1）规模：测试方案数量，测试用例数量，测试工具涉及数量，测试用例；

- (2) 工作量：文档的草稿编写工作量、评审前阅读工作量、评审工作量；
- (3) 进度：每件具体工作的计划开始结束时间、实际开始结束时间、计划工时数、实际工时数、计划完成率；
- (4) 缺陷：评审过程中出现的错误数量、缺陷数量，级别。

3.4.2 测试执行阶段的度量

包括测试用例执行率、测试用例通过率、测试用例问题发现率；缺陷数量、缺陷级别统计、缺陷分布统计（模块、阶段）、缺陷密度、人均缺陷发现效率；测试用例执行工作量、回归测试执行工作量。

3.5 测试报告

在测试结束后，把测试的过程和结果写成文档，并对发现的问题和缺陷进行分析。

4. 进入和退出条件

4.1 测试准入标准

- (1) 《需求规格说明书》中规定的优先级大于 4.0 的功能均已实现，对于没有实现的功能已提供测试范围；
- (2) 开发人员编码结束，并在开发环境已完成单元测试；
- (3) 已完成集成测试，被测系统的基本流程可以走通，界面上的功能均实现，经过代码评审并符合软件编码规范；
- (4) 测试计划已经通过评审并设置基线；

- (5) 测试用例已经通过评审并设置基线；
- (6) QA 环境要求已经被清楚地定义了；
- (7) 成功完成构建验证测试和冒烟测试。

4.2 测试准出标准

- (1) 所有测试用例都已通过评审；
- (2) 所有测试用例都已成功执行；
- (3) 达到测试计划中所规定的覆盖率的要求；
- (4) 所有发现的 bug 都已记录在案。

4.3 软件测试暂停和恢复标准

4.3.1 软件测试暂停标准

- (1) 在进行软件系统测试时，发现程序存在重大 bug（影响基本功能性的）或 bug 过多时，或者流程卡壳导致基本流程无法走通，测试无法正常进行，可申请暂停测试；
- (2) 存在其他优先级更高任务时，可向领导申请暂停测试；
- (3) 软件项目需暂停以进行调整时，测试应随之暂停，并备份暂停点数据；
- (4) 软件项目在其开发生命周期内出现重大估算、进度偏差，需暂停或终止时，测试应随之暂停或终止，并备份暂停或终止点数据。

4.3.2 软件测试恢复标准

- (1) 重大 bug 被解决或程序通过重新修正；
- (2) 优先级更高的任务已经被完成；
- (3) 软件项目被调整后重新启动，测试任务应随之启动。

5. 环境需求

5.1 系统硬件最低要求

服务器终端：英特尔 P4 CPU，1G 内存，10M 网卡，键盘，鼠标，显示屏，10M 网络带宽

5.2 软件测试环境

操作系统：Win7，Win10，Mac OS

数据库平台：MySQL

Web 服务器：Apache

MySQL 管理软件：PHPMysqlAdmin

手机终端：操作系统 Android 4.0 以上

6. 职责、人员和必需的培训

6.1 小组成员角色

本测试小组由毛一鸣担任组长，组员有金嘉诚、蔡跃区、陈晨。

角色	人员	具体职责
测试组组长	毛一鸣	负责任务分配和管理监督，并控制测试进度
测试设计员	全体	根据需求规格说明书确定测试用例和优先级
测试员	全体	执行测试，记录测试结果，编写测试报告
数据库管理员	蔡跃区	确保测试数据（数据库）环境和资产得到管理和维护

6.2 人员和培训

培训需求	培训内容	培训方式	开始时间	完成时间
业务流程	熟悉该项目的需求，掌握其业务逻辑	学生自行阅读	2017.7.13	2017.7.13
安装配置	在本机安装并配置好待测试的系统	学生自行安装	2017.7.14	2017.7.14
工具使用	学习自动化测试工具的使用	学生自行查找资料并学习	2017.7.14	2017.7.15

7. 关键项目/ 阶段性里程碑

里程碑	计划开始日	实际开始日	计划结束日	实际结束日
项目/阶段开始	2017.7.4	2017.7.4	2017.7.4	2017.7.4
SQA 测试计划同意	2017.7.11	2017.7.11	2017.7.12	2017.7.12
测试资源要求	2017.7.13	2017.7.13	2017.7.13	2017.7.13
测试队伍训练完成	2017.7.13	2017.7.13	2017.7.13	2017.7.13
要求基线	2017.7.14	2017.7.14	2017.7.14	2017.7.14
测试用例设计基线	2017.7.14	2017.7.6	2017.7.14	2017.7.7
QA 周期 1 建立接受测试执行	2017.7.14	2017.7.14	2017.7.15	2017.7.15
QA 周期 1 功能性测试执行	2017.7.14	2017.7.14	2017.7.15	2017.7.15
QA 回归测试执行	2017.7.15	2017.7.15	2017.7.15	2017.7.15
QA 性能/失败测试执行	2017.7.16	2017.7.16	2017.7.16	2017.7.16
QA 最终完整建立测试	2017.7.17	2017.7.17	2017.7.17	2017.7.17

执行				
项目状态评估审查	2017.7.18	2017.7.18	2017.7.18	2017.7.18
项目/阶段性结束	2017.7.18	2017.7.18	2017.7.18	2017.7.18

8. 风险、依赖、假设和限制

8.1 风险

本软件开发项目的全部风险承担者，以及各自在本阶段所需要承担的主要风险，首要风险承担者包括：任务提出者、软件开发者、软件测试者、产品使用者。

8.2 假设和依赖

列举出对软件产品需求分析报告中，影响需求陈述的假设因素（与已知因素相对立）。如果这些假设因素不正确、不一致或者被修改，就会使软件产品开发项目受到影响。这些假设的因素可能包括：

计划使用的商业组件，或者其它软件中的某个部件；

假定产品中某个用户界面将符合一个特殊的设计约定；

有关本软件用户的若干假定（例如：假定用户会熟练使用安卓 APP）；

有关本软件开发工作的若干假定（例如：在有货的情况下商家无条件接受订单）；

有关本软件运行环境的一些问题；

此外，确定本软件开发项目对外部约束因素所存在的依赖。有关的约束可能包括：

工期约束；

经费约束；
人员约束；
设备约束；
地理位置约束；
其它有关项目约束。

8.3 限制

确定影响开发人员自由选择的问题，并且说明这些问题为什么成为一种限制。可能的限制包括下列内容：

必须使用的特定技术、工具、编程语言和数据库；

避免使用的特定技术、工具、编程语言和数据库；

要求遵循的开发规范和标准：例如，如果由客户的公司或者第三方公司负责软件维护，就必须定义转包者所使用的设计符号表示和编码标准；

企业策略的限制；

政府法规的限制；

工业标准的限制；

硬件的限制：例如：定时需求或存储器限制；

数据转换格式标准的限制。