

EZ-Shopping APP for Android

项目总体计划

项目实训 第 5 组

2017/7/4

目 录

第 1 章 引言	1
1.1 编写目的	1
1.2 范围	1
1.3 项目背景	1
1.4 预期读者	2
1.5 参考资料	2
第 2 章 项目概述	3
2.1 工作内容	3
2.2 项目产品	3
2.2.1 程序	3
2.2.2 文档	3
2.3 验收标准	4
2.4 约束限制	4
第 3 章 项目组织	5
第 4 章 项目管理	6
4.1 阶段划分与交付产品	6
4.2 项目进度安排	6
4.3 工作环境与工具使用	7
4.4 配置管理计划	7
4.4.1 配置标志	7
4.4.2 版本管理	7
4.4.3 变更控制	8
4.4.4 配置状态报告	9
4.4.5 配置审核	9
4.5 风险管理计划	9
4.5.1 风险评估	9
4.5.2 风险控制	10
4.6 成本管理计划	12
4.7 沟通管理计划	13
4.7.1 开发者与客户沟通	13
4.7.2 开发者内部沟通	13

第 1 章 引言

1.1 编写目的

本文档是项目实训课程的组成部分，编写项目计划文档的目的是：明确该项目的范围、工作内容、工作方法、时间安排、管理与控制方法、资源情况等，使项目的实施在本计划的基础上得到实施与控制。

1.2 范围

本计划主要描述了本项目的工作内容、项目组织、项目的管理办法与过程要求、项目采用的技术、度量办法与相关的管理、控制要求。

在本计划的基础上还将形成项目的进度计划、配置管理计划、质量保证计划、总体测试计划，这些计划作为本计划的补充与具体说明，受本计划影响。

1.3 项目背景

移动支付定义：在商务处理流程中，基于移动网络平台，随时随地的利用移动智能设备，如手机、PDA、笔记本电脑等工具，为服务于商务交易而进行的有目的的资金流流动。

微支付定义：是用款额特别小的电子商务交易，类似零钱应用的网络支付方式。

物联网（IoT）：顾名思义就是物物相连的互联网，以实现环境以及状态信息实时的共享以及智能化的收集、传递、处理、执行。

场景：生活中，我们司空见惯的很多日常生活产品，其实在用户使用场景的考虑上，是有蛮多讲究的，产品经理需要学会从日常生活体验中去发现、理解产品的用户场景，对提升我们的需求分析和设计能力，是蛮有帮助的。

微移动支付应用平台开发：亚马逊 2015 年发布了一款名为 Dash Button 的小硬件产品。它是一个小型、专一、快捷的按钮，可以将按钮贴在对应的地方，并使用一些标识以便区分。比如按一下贴在洗衣机上的按钮，就可以自动下单购买

洗衣粉，一键支付，这是物联网应用的一大突破。2016 年针对 Dash Button 过于专一的缺点进行了升级，不再是一个按钮对应一个产品，而是一对多的模式，不用在家中各处贴满按钮，从某种意义上实现了“万物互联”。但是一大局限是需要用户具备一定的编程知识，显然这样的人群规模不大。综上，使用更为平凡的条形码扫描具有更大的市场，针对即将用完想要续购的商品即可进行扫描一键购买，解决了购买单一、适用人群不大的问题。

1.4 预期读者

本文档的读者包括需求开发人员、系统设计人员、编码开发人员、测试人员。

1.5 参考资料

1. 《项目实训案例 170704》
2. 2017 项目实训 参考资料

第2章 项目概述

2.1 工作内容

1. 理论知识的学习，其中包括所用软件和编程语言的学习和熟悉；
2. 该项目的需求分析，包括功能需求、系统需求、性能需求。其中功能点要与项目需求方进行沟通交流，并形成软件需求规格说明书；
3. 该项目的概要设计和详细设计，并形成软件设计说明书；
4. 编码的完善及相关注释，并形成源程序清单；
5. 项目测试及维护，并形成软件测试计划和软件测试报告；
6. 相关文档的完成。

2.2 项目产品

2.2.1 程序

应交付项目名称：EZ-Shopping APP for Android

2.2.2 文档

1. 软件需求规格说明书（SRS）：清楚、准确地描述软件的每一个基本需求（功能、性能、设计约束和属性）和外部界面，并把每一个需求规定成能够通过预先定义的方法（例如检查、分析、演示或测试等）被客观地验证与确认的形式；
2. 软件设计说明书（SDD）：对一些规模较大或复杂性较高的项目，应该把本文档分成概要设计说明书（PDD）与详细设计说明书（DDD）两个文档。其概要设计部分描述所设计软件的总体结构、外部接口、各个主要部件的功能与数据结构以及各主要部件之间的接口，还要对主要部件的每一个子部件进行描述。其详细设计部分必须给出每一个基本部件的功能、算法和过程描述等；
3. 软件测试计划（STP）：描述所采用的软件测试方法，以用来验证软件需求规格说明书中的需求是否已由软件设计说明书描述的设计实现，软件设计说明书表达的设计是否已由编码实现，还可用来确认编码的执行是否与软件需求规格

说明书中所规定的需求相一致；

4. 软件测试报告（STR）：描述软件测试计划的执行结果，需包括软件质量保证计划所需要的所有评审、检查和测试的结果；

5. 用户手册

6. 项目开发总结（PDS）。

2.3 验收标准

程序部分能够顺利、准确地完成相应需求：功能完整、准确；

文档部分达到格式标准、陈述简明准确。

2.4 约束限制

1. 7.4 日完成分组与项目计划制定
2. 7.5-7.7 日完成需求分析阶段
3. 7.8-7.10 完成项目的概要设计与详细设计
4. 7.11-7.13 完成编程实施与部署，形成可以运行的程序
5. 7.14-7.17 完成软件的测试，完成个人论文与测试报告
6. 7.18 项目评价

第3章 项目组织

姓名	工作角色	身份	工作内容
毛一鸣	组长	在校学生	给组员分配任务、制定开发计划并参与实际开发
陈晨	组员	在校学生	参与实际开发
金嘉诚	组员	在校学生	参与实际开发
蔡跃区	组员	在校学生	参与实际开发
邢卫	客户	老师	协助本小组进行需求开发并验收项目成果
金波	客户	老师	协助本小组进行需求维护并验收项目成果

第 4 章 项目管理

4.1 阶段划分与交付产品

阶段名称	交付产品名称	产品形式	交付时间
启动阶段	项目总体计划	PDF 文档	7.4
设计阶段一	软件需求规格说明书	PDF 文档	7.7
设计阶段二	详细设计报告	PDF 文档	7.10
进行阶段	可执行的项目软件	程序	7.13
测试阶段	测试报告	PDF 文档	7.17
总结阶段	个人课程论文	PDF 文档	7.17
	小组互评	EXCEL 文档	7.18

4.2 项目进度安排

时间	工作内容	所需资源	标志性事件
7.4-7.4	项目计划	项目总体计划模板	项目总体计划书完成
7.5-7.7	需求分析	参考资料	完成需求规约书
7.8-7.10	详细设计，概要设计	教师咨询，参考资料	完成概要设计说明书
7.11-7.13	编程实施部署	数据库	代码、注释全部完成
7.13-7.17	项目测试	数据库，源程序	测试结果正确
7.18	评分与评价	全小组程序	完成互评表

4.3 工作环境与工具使用

本网站要求提供对外服务的能力,保证至少 300 名用户同时请求服务的要求。包括数据存储能力,网络服务吞吐能力,数据安全特性等。

数据库选用 MySQL, 客户端选择 Android 平台。

开发平台可以选择 Eclipse 或者 Android Studio 等 ide。

4.4 配置管理计划

4.4.1 配置标志

软件项的标识基本按照《软件配置标识命名规则》进行。要通过标识能够确定软件项之间的相互联系。

4.4.2 版本管理

1. 首先在服务器上建立一个目录,作为项目配置数据库。在此目录下按照每个项目组建一个分目录,项目组代码及项目组名构成目录名,然后在此项目组目录下按照所属每个项目建一个子目录,同一项目的开发文档存放在一个目录下,项目编号紧跟项目名就是目录名。在一个项目分目录下可按非受控文档与受控文档建立一级次目录,然后在一级次目录下按文档的不同类型建立二级次目录,使得所有开发文档能分门别类的组织存放,便于查询。
2. 项目子目录的受控文档一般只有项目经理和属于该项目的开发人员和配置管理员能够访问到。配置管理员负责分配访问权限,一般项目经理对该目录具有较大的权限——读取、添加和更改;一般开发人员只有读取的权限。
3. 在项目开发的某一阶段结束时,通过了该阶段评审的这些开发文档交配置管理员保存到项目数据库,做为正式版本的第一版——1.0 版本。
4. 在以后的开发中,如果软件需要修改,那修改后的软件可用多级编号来表示新版本——1.1、1.2 等加以区别标识。
5. 在各个评审阶段产生的所有评审报告和修改报告都要进行编号保存,编号与

相应文档的编号要对应。

4.4.3 变更控制

（一）微小改正时的变更控制

1. 在评审或测试后发现的问题由评审组组长或项目经理形成《软件问题报告单》或《源代码修改记录单》，并通知配置管理员。
2. 由配置管理员将需要修改的软件的备份从项目配置数据库中检出，开发人员执行修改。
3. 修改完毕后进行修改测试，编程错误累计到了一定的量或者测试时间已满一个月（从上一次入配置库后算起），凭《源代码修改记录单》及修改后的源代码，通知配置管理员，配置管理员确定测试报告的完备性，并在核对软件修改内容和修改人员填写的《软件修改报告单》或《源代码修改记录单》中的修改描述一致后，将文件登入项目配置数据库中，生成新版本。
4. 配置管理员修改《软件配置状态表》和《软件变更记录表》，以使其他相关开发人员及时了解软件变化情况。

（二）较大变动时的变更控制

1. 开发人员或用户提出影响较大的修改要求。（这是指要增加或删除某些功能或者是发现错误的阶段在造成错误的阶段的后面等。）
2. 配置管理员在收到这类修改要求时，必须组织有项目经理以及开发人员参加的修改评审会，讨论修改的影响范围，修改的必要性、可行性以及修改方法、步骤和实施计划。
3. 在修改方案通过并经项目经理审核后，要由产品开发部经理签字批准。涉及重大技术方案的修改时，修改方案必须由总工程师或技术总监签字批准。以决断修改工作中各项活动的先后顺序及各自的完成日期，以保证整个开发工作按原定计划日期完成。
4. 配置管理员在接到修改批准——由项目经理或产品开发部经理或总工程师或技术总监签字同意的《软件问题报告单》后才可将需修改的软件的备份从项目数据库中检出，开发人员执行修改。
5. 修改完毕后，交客户服务部进行测试和评审，测试和评审都通过后，交配置

管理员处。

6. 配置管理员检查测试报告和评审报告是否完备，核对《软件修改报告单》中的修改描述和修改后的软件是否相符。核查结果符合要求，配置管理员将修改后的软件登入项目数据库中，生成新版本。
7. 配置管理员修改《软件配置状态表》和《软件变更记录表》，以使其他相关开发人员及时了解软件变化情况对受影响的软件做出相应的修改。

4.4.4 配置状态报告

1. 两份配置状态报告——《软件配置状态表》和《软件变更记录表》分别以电子表格的形式存放在项目分目录下，以便项目开发人员随时查询，了解软件的修改变化情况。
2. 《软件配置状态表》由配置管理员负责填写，主要反映项目中各软件项的配置情况。开发人员通过查阅该表可及时全面的了解项目中软件项的配置使用情况。
3. 《软件变更记录表》由配置管理员负责填写，主要记录软件开发过程中所有的修改情况，该表以修改时间排序，以便开发人员及时了解软件项最新的变化。

4.4.5 配置审核

为保证各项产品在技术上和管理上的完整性，总经理室在软件开发过程中的详细设计阶段和测试阶段完成时，对配置情况进行抽查。总经理室先提出要审核的内容和各项指标，逐项审核完成后要作好记录，形成《配置审核报告》。

4.5 风险管理计划

4.5.1 风险评估

（一）需求获取方面的风险

1. 产品前景和项目范围没有达成明确共识引发的风险
2. 需求开发所需的时间分配不合理引发的风险
3. 需求规格说明的不完整性和不正确性引发的风险
4. 创新产品的需求不完全引发的风险
5. 忽视非功能需求引发的风险
6. 客户对产品需求意见不一致引发的风险
7. 未加说明的需求引发的风险
8. 对已有的产品作为需求基线来源引发的风险
9. 根据用户提议的解决方案引发的风险

（二）需求分析方面的风险

1. 设定需求优先级引发的风险
2. 技术上难以实现的特性引发的风险
3. 不熟悉的技术、方法、语言、工具或者硬件引发的风险

（三）编写需求规格说明方面的风险

1. 需求理解引发的风险
2. 尽管问题待确定但迫于时间压力而继续向前引发的风险
3. 具有二义性的术语引发的风险
4. 需求中包括设计引发的风险

（四）需求确认方面的风险

1. 未经确认的需求引发的风险
2. 审查熟练程度引发的风险

（五）需求管理方面的风险

1. 变更需求引发的风险
2. 需求变更过程引发的风险
3. 为实现的需求引发的风险
4. 扩大目标范围引发的风险

4.5.2 风险控制

（一）需求获取方面的控制

1. 在项目早期编写一份包括业务需求在内的前景和范围文档，并将它作为添加新需求和修改现有需求的指导。
2. 合理安排需求开发所需的时间，需求开发活动的工作量应占项目总工作量的10%-15%。
3. 在与客户的密切合作下完成对项目需求、实际功能的明确认识。
4. 强调市场调研、构建原型并成立客户小组，负责获取对新产品前景的反馈信息。
5. 向客户询问以获得相应的质量特性需求，例如性能、易使用性、完整性和可靠性需求。尽可能精确的在软件需求规格说明中，对这些非功能性需求及其验收标准编写文档。
6. 确定主要客户，并采用产品代言人的方法，保证有足够的客户代表的积极参与，确保由合适的人对需求做出权威性的决策。
7. 尽量识别客户可能做出的任何假设。提出自由回答的问题来鼓励客户分享更多的想法、期望、主意、信息和关注点，而不是我们以其他方式所听到的。
8. 通过逆向工程发现的需求编写成文档，让客户评审这些需求，以确保其正确性和相关性。
9. 分析人员必须提炼出隐藏在客户提出的解决方案背后的真正意图，综合选择最佳的问题解决方案。

（二）需求分析方面的控制

1. 要确保每个功能需求、特性或用例都设定了优先级，并安排在一个特定的系统版本或迭代中实现它们。
2. 评估每个需求的可行性，确定哪些需求的实现时间可能比预期长，尽早采取措施。
3. 为满足某些需求而采取新技术时，要考虑到学习曲线的问题，只有通过一定的学习时间才能达到适当的熟练程度。要尽早确认那些高风险的需求，并留出足够的时间让用户从错误中学习经验，实验以及制作原型。

（三）编写需求规格说明方面的控制

1. 对需求文档进行正式评审的团队应该包括开发人员、测试人员和客户等，以减小需求的不同理解造成的风险。

2. 应该记录下负责最终解释每个负责人的姓名和解决的截止日期。
3. 创建一个数据字典来定义一些术语的条目和结构，对软件需求说明的评审可以帮助参与者对关键术语和概念达成一致的理解。
4. 对需求的评审，可以确保强调的是需要解决的业务问题是什么，而不是规定如何解决。

（四）需求确认方面的控制

1. 在构造设计开始之前，确认需求的正确性和质量，应该为质量保证活动预留出一定的时间并提供资源，要确保客户参与需求审查活动。
2. 要对参与需求文档审查的所有团队成员进行培训，请组织内部有经验的审查人员或者外界的咨询顾问来评述早先的审查。

（五）需求管理方面的控制

1. 应该推迟实现那些很可能还要发生变更的需求，待确定之后再实现，并在设计时要考虑到应该使系统易于修改。
2. 需求变更过程要包括对提议的变更进行影响分析，组建变更控制委员会作出决策，使用工具支持预定义的过程。
3. 需求跟踪矩阵有助于在设计、构造或者测试期间避免遗漏任何需求。
4. 应该制定分阶段或者增量的交付产品的实现计划。在初始版本中先实现核心功能，在以后的迭代中再逐步增加系统功能。

4.6 成本管理计划

开发者人数：4 人

开发周期：15 天

需求工程经费预算：无

4.7 沟通管理计划

4.7.1 开发者与客户沟通

与项目发起者——老师沟通：在需求开发阶段，与项目发起者即老师进行交流。以小组讨论方式确定本项目的需求。其他沟通可通过邮件或短信电话方式进行。

4.7.2 开发者内部沟通

开发者每天按时上班，出现问题及时组内沟通解决。