

项目开发总结报告（GB8567——88）

1 引言

1.1 写作目的

本文将回顾该项目的开发过程，检讨此次软件工程实践项目给我们带来的经验与教训

1.2 项目背景

项目名称 在线浏览附近商家商品 app(EasyPick)
软件系统名称 EasyPick
任务制定人 丁鑫棚
开发团队 丁鑫棚 董自陈 汪哲汇 杨瑞 殷容碑
用户 全体消费者

1.3 定义

1.4 参考资料

2 构件简述

2.1 构件

客户端运行于具有 Android2.2 以上系统上，服务器端运行于 Flask/Tomcat/MySQL 环境

2.2 主要功能与表现

系统的主要特性：

商家：

- a) 可以添加/删除商品
- b) 可以添加商品信息，修改商品价格
- C) 可以修改商店名称，营业时间
- D) 不能修改商家地址

用户

- c) 可以根据浏览、查找商品，商家
- d) 可以查看商品，商家信息
- e) 可以对商品/商家进行评论和打分
- f) 可以收藏商品/商家
- g) 可以查看其他用户的信息
- f) 可以修改自己兴趣
- g) 可以修改除了账号外其他基本资料

系统设计过程中考虑到了大部分功能需求和非功能性需求，具有良好的可扩展性。

2.3 基本 workflow

系统以三层结构组织， workflow 如下：

用户请求->客户端 UI->请求->转发 Controller->子系统 Controller->业务逻辑
->持久化映射层->数据库

3 项目分析

3.1 工作效率分析

项目组很好的进行了分工与协作，这主要依赖于小组成员紧密的互相联系。通过 IM 工具和版本控制工具，系统所有的信息在小组成员之间进行了充分流通，这也提高了工作效率。

在项目开发过程中，也出现了一些拖延现象，具体分析见第四部分，问题分析。

3.2 产品质量分析

项目组完成了所有规定的任务阶段(可行性分析\需求获取\需求分析\系统设计\对象设计\测试\总结)。在每个阶段都生成了对应的主文档(可行性分析报告\系统需求文档\系统架构文档\测试计划\项目总结)。符合项目要求。

3.3 使用技术评估

所选择的所有开发和运行环境均为开源软件。

3.3.1 项目管理技术

项目组使用 qq 工具和邮件进行沟通。项目文档使用 git 源代码跟踪技术进行管理。这保证了项目组文档的一致性和有效性。在实践中，版本管理系统很好的实现了同步和交流的功能。

3.3.2 系统架构评估

3.3.2.1 系统整体架构

系统使用了轻量级 python Flask 模型。我们选择了这样的平台:Apache/Jpython/Tomcat/MySQL, 这样的选型主要有两大优势:

1. 系统技术成熟，版本稳定, 基本无 bug

2. 容易扩展，可以很容易进行不同节点的拆分

3.3.2.2 系统设计

系统选用了传统的三层结构。清晰，明了，子系统耦合较小，内聚性高。

系统的MVC框架选用 Struts,持久化存储使用 Hibernate, 依赖注入使用 Spring, 这一组合可靠，稳定，很好的满足了要求。

3.3.3 项目设计技术评估

项目设计中大量使用了设计模式来降低耦合，包括观察者模式、桥接模式、代理模式等。

3.4 失误分析

I. 项目同步不足

由于项目由多人协作构成，不同人对相同问题理解有差异。在开发伊始，每个

人都维护着自己的事情，使得项目内部出现了较严重的不一致现象。

II. 拖延问题严重

拖延问题主要是由项目组成员协作不足所导致的，由于对彼此的分工不够清晰，遇到问题缺乏有效的交流机制，导致很多问题进度缓慢。解决该问题的方法主要是定期的项目组会议，在每个迭代周期明确每个人的任务。

III. 频繁变更需求

4 经验总结

4.1 团队需要有效的协作手段

几乎所有的问题都出在项目组缺乏有效沟通上。我们使用 IM 工具来缓解这一问题，或许事实上更好的解决方法是定期的例会和有效的文档(代码)。

4.2 仔细设计，然后敏捷

在分析-设计-编码迭代过程中，使用敏捷方法可以避免项目组一上来陷入需求分析的海洋，选择最重要的一组用例进行分析和实现，有助于理解系统，更好的设计需求。同时，这也使得项目组更加可控

4.3 享受设计和编码

5 尚未解决的问题

5.1 推荐算法的完整实现

5.2 商家客户端还未实现，目前商家信息为自己添加

5.3 允许高并发