

复旦大学《高级软件工程》课程

2017-2018 学年课程期末报告

学号：17212010063

姓名：梁文杰

1. 软件需求

根据自身使用经验并查询相关系统介绍, 分析复旦大学校园一卡通系统并回答以下问题。

1) 简述该系统主要涉及哪几类涉众 (Stakeholder), 每一类涉众针对一卡通系统的主要目标是什么。

1、学生老师等使用一卡通消费的群体

通过对一卡通的充值, 以达到在校园范围内使用一卡通消费的目的。

2、食堂小店等使用一卡通收费的店

通过对一卡通内的余额进行扣费转账, 达到快捷收款的目的。

3、图书馆等使用一卡通出入的场合

规范能够出入特殊场合的人群。

4、学校等推广一卡通规范的部门

推广规定校园内使用一卡通消费, 规定可以获得一卡通的群体, 并对不同的群体制定收费规范, 使达到合理的资源分配, 不让学校资源外流的目的。并且规范一卡通的充值补办挂失业务, 对一些场合的出入进行规范、记录使用者的使用记录, 达到一些特殊的目的。

5、一卡通技术提供商

提供一卡通技术的支持, 并通过自己的技术赚取报酬。

2) 选择某一类涉众的一个目标，围绕该目标介绍一个相关的用况（Use Case），
具体包括：目标名称、用况名称、相关的参与者和交互方、用况与所选择的目标
的关系。

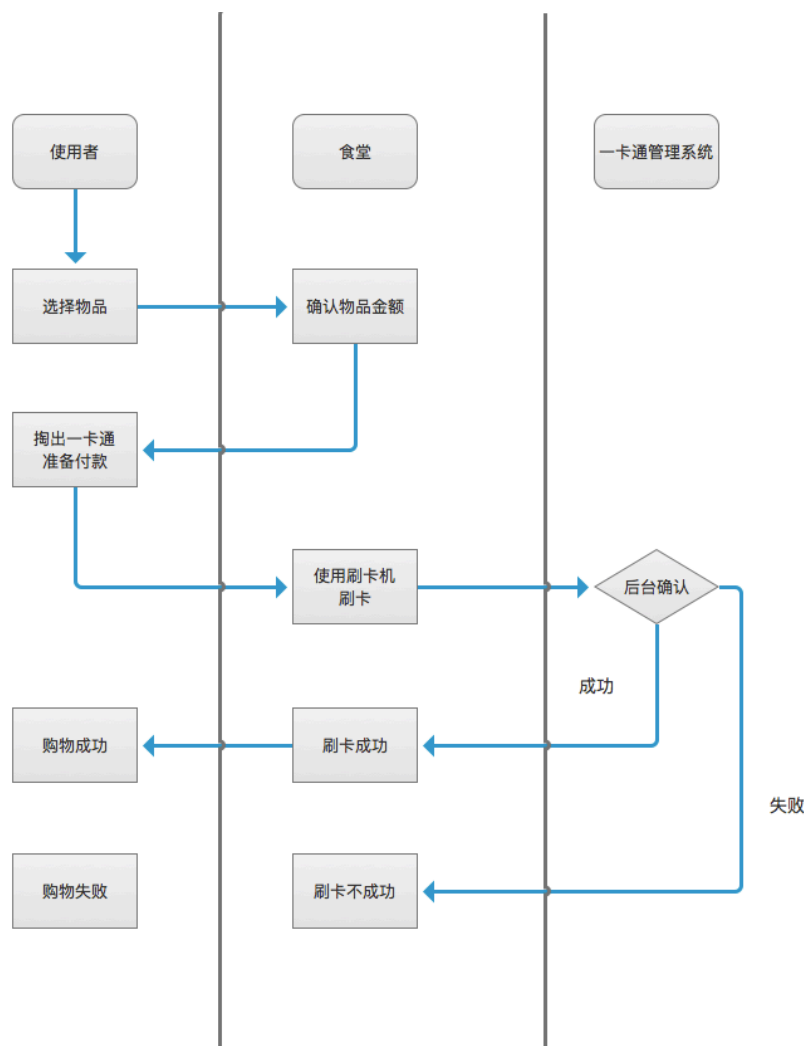
目标名称：使用者使用一卡通在食堂消费

用况名称：一卡通消费

相关的参与者和交互方：一卡通使用者、食堂、一卡通管理系统（学校）

用况与所选择的目标的关系：通过一卡通可以在食堂进行正常消费。

3) 针对 2) 中所介绍的用况，使用 UML 泳道图对其进行描述。



2 . 软件设计

针对某种广泛使用的软件体系结构风格或模式, 选择一个熟悉的复杂软件系统(如大型复杂在线交易系统、网上购物系统、智能家居系统等分布式软件系统)回答以下问题。

1) 概述该软件体系结构风格或模式的内容和内涵。

为解决高并发, 高可用的大型电商网站的架构设计方案, 主要采用了分布式、集群、负载均衡、反向代理、消息队列及多级缓存技术。

架构按层次结构罗列组织, 共分为四层, 分别为负载均衡代理层、应用集群系统层、分布式服务层及数据资源层

在前端返回数据之后, 通过对应的负载均衡服务器把请求发送到对应最快的服务器上。然后对于海量的数据采取分布式冗余存储, 解决空间和安全的问题。采用两级缓存, 一级缓存为本地缓存, 二级缓存为分布式缓存。

2) 概述该软件体系结构风格或模式在该系统中是如何应用的以及发挥了什么样的作用。

借助互联网环境连接不同服务器, 并各个连接的服务器之间通信交互, 提供服务异步调用和返回的通信机制。

数据库集群, 可以实现主备数据库, 做到读写分离以及高可用的实现。冗余方式设计数据库集群, 最为常用的方式为: 读写分离和分库分表了。主数据库服务器只负责写入数据, 而备用服务器数据库只负责读取数据, 可以做到降低数据库的 IO 压力; 另外, 如果业务系统比较庞大, 可以进一步根据业务的关系度及

增长频率分库，若库中的但表数据量比较大，可进一步分表，具体的分库分表可[查看我的博客文章数据库的分库分表](#)。

消息队列，是分布式系统的常用组合，其可以解决子系统或模块间的异步通信，实现高可用，高性能的通信系统。

对于缓存，应用程序首先访问一级缓存；如果一级缓存没有需要的信息，那么取访问分布式缓存，如果分布式缓存也没找到需要的信息，最后去访问数据库获得数据，平均情况下加快了数据的读取速度

3) 进一步分析和总结该软件体系结构风格或模式的通用性以及适合于什么样的软件系统。

该软件体系结构风格适用于需要高并发、高可信、高效率、拥有较大资金支持的大型系统，可以解决那些小系统解决不了的海量数据访问问题。对于不需要解决这些问题的小系统，不适合用如此高成本的系统。

3 . 软件构造

从 Github、Bitbucket 等开源社区中选择一个采用了 Pull Request 开发的开源软件项目,通过分析该项目的一个最终被采纳的 Pull Request 实例回答以下问题。

1) 该 Pull Request 针对的是什么问题？

Github 上的 bootstrap 项目：

最新一次 pull request 为距现在 5 小时前。

针对的问题为：对于标签 position：stick，更新浏览器的兼容性（edge 现在支持它了）。Edge 的最新发货版本支持 position：sticky，并且请求者已经用 Bootstrap

4 成功地测试过了，它可以正常工作。在那个标签中，请求者添加了 IE10 来代替 Edge，因为他认为 IE10 也有这个 position: relative 备用功能。

2) 该 Pull Request 经过了什么样的审核过程（包括讨论、评论、测试、评审等）之后被接受？

- 1、该 pull request 首先被 block，因为需要先进行一次自动审查。
- 2、自动审查通过
- 3、据需被组织，因为该请求需要至少一个具有写入权限的审查者审核一次。
- 4、具有写入权限的审查者 pull 更改后的代码并测试通过之后，同意该 pull request 请求，代码被 merge 到开发库中。

3) 该 Pull Request 被接受后对于项目整体的更新（如模块的变动等）包括哪些？

更新了 docs/4.0/utilities/position.md 文件，更新了其中的一个标题，把

****Microsoft Edge and IE11 will render `position: sticky` as `position: relative`.****
修改成了 ****IE11 and IE10 will render `position: sticky` as `position: relative`.****。

4 . 问题论述

群智化软件开发是指利用基于互联网平台的群体智慧所进行的软件开发。开源软件开发是群智化软件开发的一种典型方式。除此之外还包括互联网众包开发（如 TopCoder）等其他模式。

请围绕“开源和群智化软件开发”这一问题进行论述（1500-2000 字，不含参考文献），系统化地回答以下几个问题：该问题产生的背景是什么，即该问题在软件开发实践中的意义和作用；该问题含义是什么，即该问题包括哪些子问题；该问题所面临的主要困难和挑战是什么；围绕该问题目前有哪些研究工作或商业

化工具，这些工作或工具取得了哪些进展或解决了哪些问题？经过总结后，你觉得该问题目前的总体研究和实践应用现状是什么，今后会如何发展？

注意：请根据文献阅读和自己的理解，用自己的语言进行论述。论述篇幅较短，请突出重点、条理清晰。可以根据需要列举 3-5 篇参考文献，直接引自文献的内容请按照论文引文的方式规范标注。

开源和群智化软件开发是在一定的背景条件下产生的。开源软件被定义为描述其源码可以被公众使用的软件[1]，并且此软件的使用，修改和分发也不受许可证的限制。但它件通常是有 copyright 的。开源软件的开发很好的体现了软件的群智化开发。开源源自黑客对智慧成果共享、自由的追求。开源运动发展到现在，这种能够积极促进人类文明发展的文化已经渗透到信息、教育、健康等领域，融入了哲学范畴。

开源运动的史前史包括了整个 Unix，自由软件和黑客文化的历史。“开放源码”一词来源于 1997 年春天在加州的 Palo Alto 召开的一个所谓“纯粹程序员”参与的战略研讨会。参加会议的有 Todd Anderson，来自 Foresight 研究所的 Chris Peterson，来自 Linux 国际协会的 John Hall 和 Larry Augustin，有硅谷 Linux 用户协会的 Sam Ockman，以及 Eric Raymond[2]。它们关心的是寻找一种方式，来像以前躲避自由软件的人们来推广这种思想，自由软件和自由软件基金会的反商业信条让很多人对自由软件敬而远之。在 Eric Raymond 的坚持下，他们一致通过了用新的术语：OpenSource（开源软件）来描述他们所推进的软件。1998 年 2 月 23 日 Netscope 宣布它将发布 Navigator 浏览器的源代码成为开源软件发展历史的转折点，经过一番激烈争辩，“开放源码”取代“自由软件”成为黑客们对

开放原始码软件的代名词。一些在开源中的重要人物：Richard Stallman——自由软件运动的精神领袖；林纳斯·托瓦兹——Linux 操作系统的创造者。

同时，除了以上自由软件的原因之外，还有其他一些重要的原因。传统的商业软件需要巨大的广告预算，这些在背后需要太多的关注，已经远远超出了它的重要性。但传统的商业软件事实上只是所有软件的很小的一部分，或者说某些需要高成本长时间开发的技术可能在软件中占的比重并不大，但又不可或缺。这时候，顺势采用开源让广大开发者参与进来，既加快了开发进度，又减低了开发成本，对于商业软件公司来说也是有极大帮助的。特别是经济危机以来不断的成本削减，让开源也成为了大公司们的一种策略。同时对于广大普通开发者，他们一些好想法得以通过开源的形式，把一个人难以完成的项目完成，从而不但让自己的智慧得以流传，也让自己得到了个人难以完成的软件。

开源软件并不是一个问题，我们应该正确的看待它。表面上它好像并不看重知识产权，把自己的工作无偿贡献给了他人。但是这背后是对人类发展的巨大促进。如果一些关键技术永远掌握在一些少数人的手中形成垄断，那社会的发展将会无比的缓慢。那些公司们也会凭借着这些已有技术高枕无忧不思进取，严重拖缓了历史的发展进程。

从另一面看，开源群智化的开发似乎也是对软件质量的更好保证。一些研究已经显示了 open-source 软件与别的商业软件相比在具有可靠性上有更大优势。更加有效的开发模式，更多的独立同行对代码和设计的双重审查，以及大部分作者对自己作品的极大荣誉感，都对其优良的质量有所贡献。一些公司甚至给予发现 Bug 者以物质奖励。私有软件有很多“死角”，隐藏着许多 Bug。源码对于查错和理解产品工作原理来说很重要。在大的软件公司，只有少数人能接触到源

码，而这些能接触源码的人通常用户都无法直接接触到。同时能接触源码对于修补安全漏洞来说，也是非常重要的[3]。

开放源码给用户极大自由，使他们能够按照自己的业务需求定制软件。大型组织能从即使很小的定制行为中削减大量开支和人力成本。用户的挑错和改进反过来对标准开放源码软件包也是个贡献。这在传统的私有软件开发中是不可能得到的。

但是，开源软件目前也面临着一些困难。一方面是一些闭源公司任然在一些方面混的风生水起。他们深知如果开源的话他们将面临巨大的市场挑战。但这对开源软件开发来说并不是一个好消息，对于用户来说也必须忍受公司的低效与可能的不便利改版，还有隐藏的收费陷阱。

在过去，大多数开源代码都是由志愿者编写的。如今，绝大多数像 Linux 和 OpenStack 的项目代码贡献来自于红帽和 Intel 等的程序员。公司在开源开发上投入这么多钱是件好事，但这种变化确实反映了企业对开源代码的控制程度要高得多。如果导致紧张关系，开源社区必须学会应对。这是来自支持开源企业的挑战。

开源社区自己也存在着一套工具和规范，来支撑整个开源社区的正常运作。

首先就是一系列的开源协议。比如：LGPL 许可证、MPL 许可证、BSD 许可证、GPL 许可证、QNCL 许可证、IBM 许可证、Apache Licence、MIT。在这些协议下，作者可以充分保护一些自己最基本的知识产权，也限定了一些使用者的行为。者对于社区来说是有利的。接下来，就是一套开发写作工具与平台。一些简单地交互工具如 Email 和 IM 等都是很有用的，还有就是项目管理类工具 jira 等、文档管理类 confluence 等、代码托管平台 github 等，还有 git 这样的分布式

版本控制系统给予了开源软件的协作开发强有力的支持,通过它开发者们可以轻松地进行版本管理。

开源总体来说在目前属于欣欣向荣的状态。从十几年前没有几家公司从事开源的项目,也没有多少人使用开源的软件,到目前开源项目遍地开花,开源软件走进了千家万户,变化可以说是翻天覆地的,未来也是非常光明的。而且,我觉得开源运动不应该止步于软件领域。实际上,越来越多的制造企业正在为开源做准备,而且已经涌现出了一批前景看好的案例。随着开源运动进一步在医药、制造等硬件领域普及,人类有望形成一种新型的经济模式。它建立在相互的信任关系之上,能为参与各方都创造价值。

[1] Raymond E. The cathedral and the bazaar[J]. Knowledge Technology & Policy, 1999, 12(3):23-49.

[2] baike.baidu.com/item/开放源代码/114160?fromtitle=开放源码&fromid=7176422

[3] 王怀民, 基于群体智慧的群体化软件开发方法, CNNC2014, 2014,10