# Canteen Hunter移动应用

# 架构评审报告

## 引言

* 1. 编写目的

本文档的编写目的是对“Canteen Hunter”的概要设计进行说明，为后继的详细设计等工作提供参考和依据，本文档主要描述的内容有：

* 分析
* 测试
* 形成报告
  1. 背景
* 系统名称：Canteen Hunter
* 开发者：FlasHope小组。组长：陈吕劼。

目前在校园内，吃总是会给同学带来困扰：“去哪吃”“哪里人少”“哪里好吃”等。为了满足这方面需求，存在一种便捷的方式，为吃饭的餐厅提供合理的建议，或者详细查看某餐厅信息显得非常有必要。在这样的背景下，Canteen Hunter 应运而生。

* 1. 定义

1. 三层架构

软件开发的三层架构是指：UI（User Interface）用户界面层，BLL（Business Logical Layer）业务逻辑层以及DAL（Data Access Layer）数据访问层，用户界面层设计出友好的用户界面，并做好对后台的接口工作，数据访问层负责对数据进行封装以及其他相关操作，业务逻辑层则介于两层之间，负责前后台的整合、业务的描述以及实现。

1. ATAM

Architecture Tradeoff Analysis Method(构架权衡分析方法）。他是评价软件构架的一种综合全面的方法。这种方法不仅可以揭示出构架满足特定质量目标的情况，而且（因为它认识到了构架决策会影响多个质量属性）可以使我们更清楚地认识到质量目标之间的联系——即如何权衡诸多质量目标。

## 准备工作

对项目的评估是使用的ATAM架构评估综合方法。 有如下两个评估原因：首先，无论如何，如果架构本身在根本上存在缺陷，则早发现比晚发现要好得多；其次，该系统存在同一时段多用户同时使用的可能，因此必须保证该架构具有足够的健壮性、可扩展性以及可修改性。

因此我们由小组中开发人员成立了评估小组，为确保架构质量打下基础。

## 评估

评估期间，小组经过多次讨论，从多个角度出发，表述了该系统的商业目标，对于开发者来说，提出如下业务需求：

|  |
| --- |
| 1）可获得每个餐厅的菜色介绍，包括价格、图片介绍、人员评价、口味分类、适合搭配、忌搭配、卡路里、不适合人群、特别注意等。 |
| 2）可及时了解每个餐厅人数情况。 |
| 3）用户可通过多种途径记录下自己的就餐情况，应用可通过这些数据改进对用户推荐时的准确性。 |
| 4）根据用户个人情况和餐厅情况推荐适合餐厅和菜品。参考的个人情况包括就餐时间、就餐地点、口味偏好、忌口、常去地点、个性需求（减肥、健身）等；餐厅情况包括当日人数、当日特色菜、餐厅评价等。 |
| 5）提供搜索功能，可以根据菜名，口味，偏好等关键内容进行搜索。 |
| 6）提供提前预定服务和外卖服务。其中提前预定服务可以教师优先、活跃用户优先等方式来提供；外卖服务会适当裁剪，避开与专业外卖应用的竞争冲突。并与之配套便捷的在线支付功能。 |
| 7）提供老朋友、同乡人之间的约餐服务，提供陌生人之间的拼餐业务。通过一定的奖惩措施保障用户的正常体验。 |
| 8）用户就餐后，可对当日就餐餐厅或菜品提出评价和意见，评价讲提供给其他用户作为选择餐厅的参考，意见将提供给餐厅以作为改进的参考。 |
| 9）应用能够提供增值服务和植入软性广告等，保证应用的收益。 |
| 10）在用户选定就餐餐厅后能够向其推荐周边商家（超市等） |
| 11）满足用户动手欲，提供某些方便美味的菜品制作过程食谱等。 |

该系统的限制包括：

1. 软件开发在校园内进行，在资源的接收与获取方面，资料的主要来源为网络、学校公共平台、校内专业人员，同外界企业相比较，我们的资料来源相对有限。
2. 开发周期太短，时间不允许开发人员对功能有更细微的完善。

如下质量属性被确认为高优先级的质量属性：

1. 性能。不允许出现过长的响应时间。
2. 功能性。功能要完善。

可用性。需要有美观的界面，以及真正有意义的功能。

### 架构表述

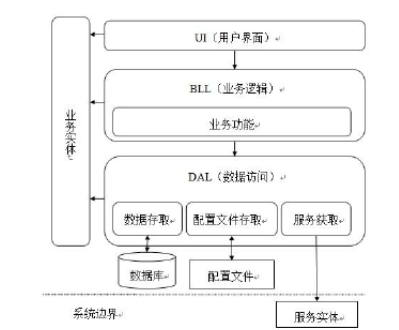
在评估小组与设计师进行交流期间，在评估开始前和评估进行中，需要有几个构架视图和构架方法。如下几点内容非常关键：

(1) 餐厅信息主要由数个数据模块构成。

(2) 要将餐厅信息构建为具有极高的可配置性。

 (3) 平台系统需要进行分层。

(4) 餐厅信息是一个基于存储库的系统，该系统的中心有一个实用的数据库。



每当用户向系统请求查看餐厅信息时，平台会根据用户传来的关键字，在数据库中相应表项进行检索后，将数据回传给用户。

### 

存储 读取

提交 反馈

服务器

DAL

客户端

用户手机

### 生成质量属性效用树

下表给出了在对短信平台评估期间生成的质量属性效用树，有几个质量属性求精没有与之相关的场景。这种情况经常出现，这并不是问题，对于某个质量属性，人们有时能够想出一个合理的求精，但当让他们在自己的系统的上下文中对该质量属性聚义用例进行说明时，却发现该求精实际上并不适用。

|  |  |  |
| --- | --- | --- |
| **质量属性** | **属性求精** | **场景** |
| 性能 | 响应时间 | 系统处于峰值时应在3秒内做出处理 |
| 吞吐量 | 峰期负载系统能满足500人同时请求 |
| 易用性 | 操作无逻辑障碍 |  |
| 操作提示 | 用户初次使用应有相关教程 |
| 可维护性 |  | 维护人员遇到了搜索和响应时间缺陷，修复该bug，并发布新版本 |
| 可靠性 | 机密性 | 不存在用户隐私泄露情况 |
| 意外故障处理 | 在运行过程中遇到意外导致系统停止运行，重启APP，响应时间不超过10S |
| 可扩展性 | 充足接口添加新功能 | 提供在线支付 |

## 结果表述

ATAM评估的一个具体结果就是生成了最终报告，该报告包括一个有风险决策、误风险决策、敏感点和权衡点的列表。他还包括一个涉及如下内容的目录：所使用的架构方法、效用树、经过集体讨论确定的场景以及所选择的每个场景的分析记录。最后，最终报告还包括由该评估小组所确定的风险主题的集合，并指出了每个风险主题所危及的商业动机。