

2019-2020-2 学期
《面向对象分析与设计》
课程报告

姓名：张世琛

学号：1804030401

班级：计算机 2018-2 班

项目名：网上订餐信息管理系统

2020 年 6 月 10 日

目 录

1	远景.....	1
2	业务建模.....	1
	2.1 业务用例图.....	1
	2.2 业务用例的活动图.....	1
	2.3 业务对象模型.....	2
3	用例建模.....	2
	3.1 用例图.....	2
	3.2 用例文档.....	3
	3.3 概念类图.....	4
4	用例分析.....	4
5	设计模式.....	7
6	架构设计.....	9
	6.1 系统架构.....	9
	7.2 系统部署.....	10
7	构件设计.....	10
	7.1 用例设计.....	10
	7.2 状态图.....	12
	9.3 类图	12
8	个人总结.....	13

1 远景

我打算打造一个网上订餐系统, 主要目的是为了给大学生的提供便利的订餐, 让大学生可以方便地订餐, 还可以对商家的餐饮进行评价, 由系统生成评价档案以供其他人参考等, 而商家后期只负责做饭菜并安排人配送。此外, 需要定期对商家进行卫生安全监察, 生成商家监察档案, 并以此为依据来决定商家的去留等。

2 业务建模

2.1 业务用例图

本系统使用对象为大学生和餐饮商家, 初步设计本系统针对大学生的功能有订餐、餐饮评价、登录系统, 以后逐步完善。

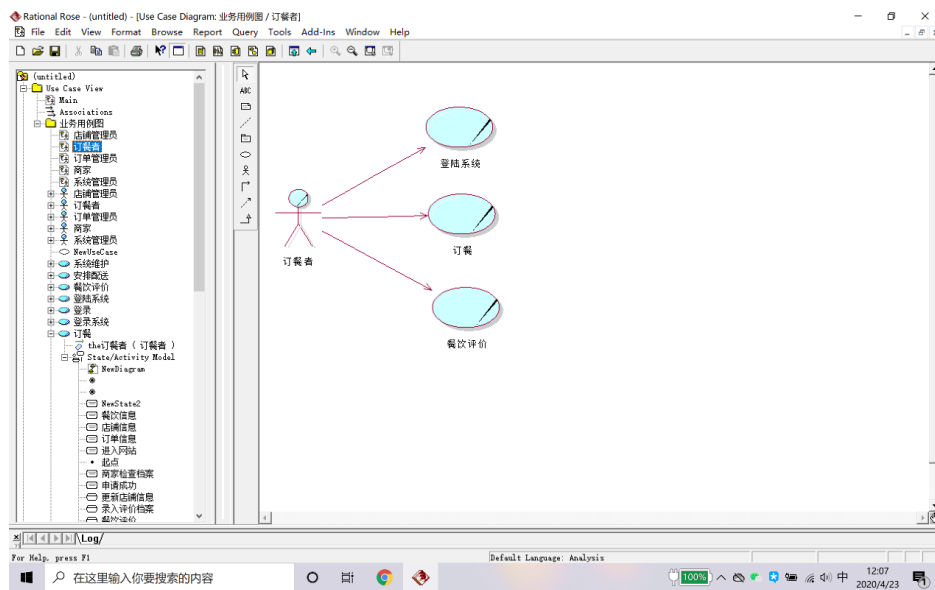


图 2-1 业务用例图

2.2 业务用例的活动图

订餐的流程如下: 订餐者登录系统并查询餐饮信息—>订购餐饮—>对餐饮进行评价—>订单处理—>更新店铺信息。

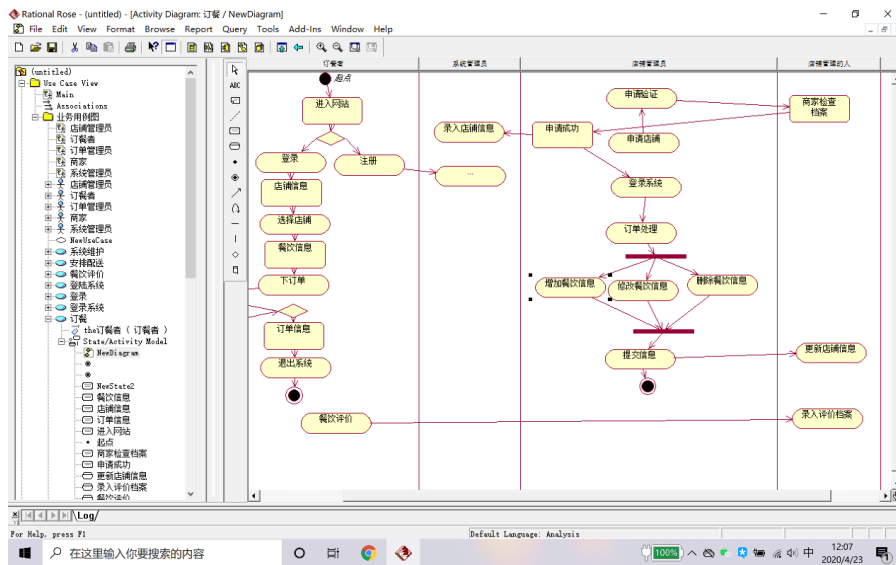


图 2-2 业务用例的活动图

2.3 业务对象模型

雇员分为三部分：

- 订单管理员：管理订单
- 系统管理员：管理店铺信息
- 店铺管理员：管理餐饮信息

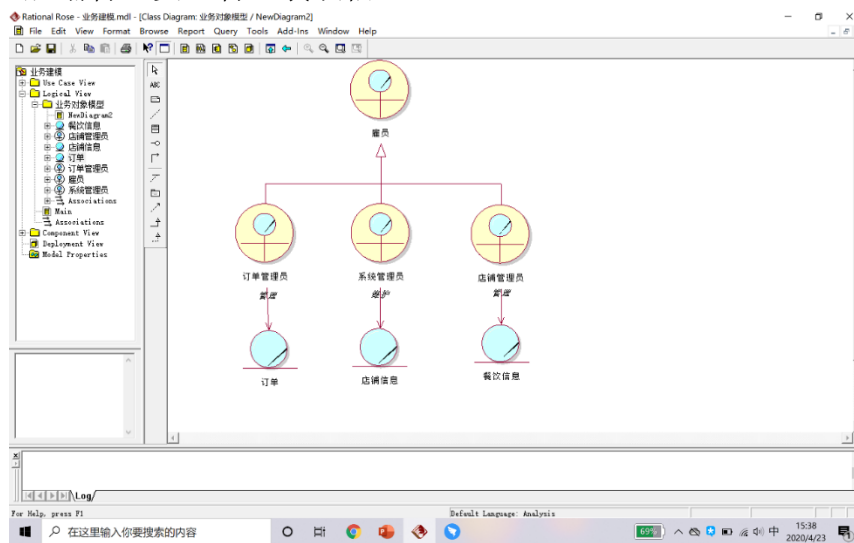


图 2-3 业务对象模型

3 用例建模

3.1 用例图

本系统参与者为订餐者、财务管理员、订单管理员、店面管理员，有各自的

用例，住户登录系统进行操作，物业工作人员处理住户发出的请求。

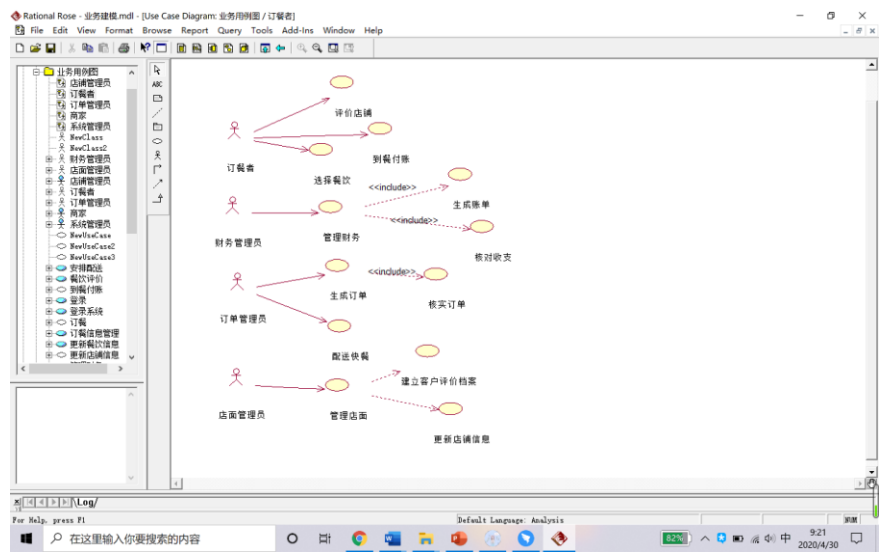


图 3-1 用例图

3.2 用例文档

用例名称	订餐
简要描述	订餐者登网站，选择餐饮，并下单，完成订餐
参与者	订餐者
涉众	订餐者：订购自己心爱的餐饮
相关用例	订单、餐饮
前置条件	订单者正确登录到该系统
后置条件	如果预购成功，则系统记录预定信息，相关的餐饮被正确标识
基本事件流 (1) 用例起始于订餐者需要在网上订购餐饮 (2) 订餐者按照要求设定查询条件（D-1）来查询可订餐的餐饮信息 (3) 系统显示所有可预定的餐饮信息(A-1) (4) 订餐者选择种类餐饮(A-2),并输入餐饮的个数 (5) 系统计算所需的总费用（B-2） (6) 订餐者网上支付所需的金额（A-3） (7) 系统保存订购信息，显示订购成功的信息(A-4),用例结束	
备选事件流 A-*系统在保存订购信息前，随时都可以终止改用例 (1) 系统提醒订餐者当前所有操作都会被取消 A-1 没有找到可预定的餐饮 (1) 系统显示没有可订购的餐饮信息 (2) 订餐者可以重新设定查询条件，也可以选择结束改用例	

A-2 所有可订购的餐饮没有让订餐者满意的餐饮

订餐者可以重新设定查询条件,也可以选择结束改用例

A-3 订餐者没有足够的金额来支付

订餐者可以选择其他种类的餐饮,或者修改餐饮个数,也可以选择结束该用例

A-4 系统保存失败

(1) 系统显示保存失败信息,并提醒订餐者可以重新提交

(2) 订餐者可以重新提交本次订购信息,也可以选择结束改用例

补充约束-数据需求

D-1 查询条件包括餐饮是否有剩余以及能提供多少份餐饮

补充约束-业务规则

无

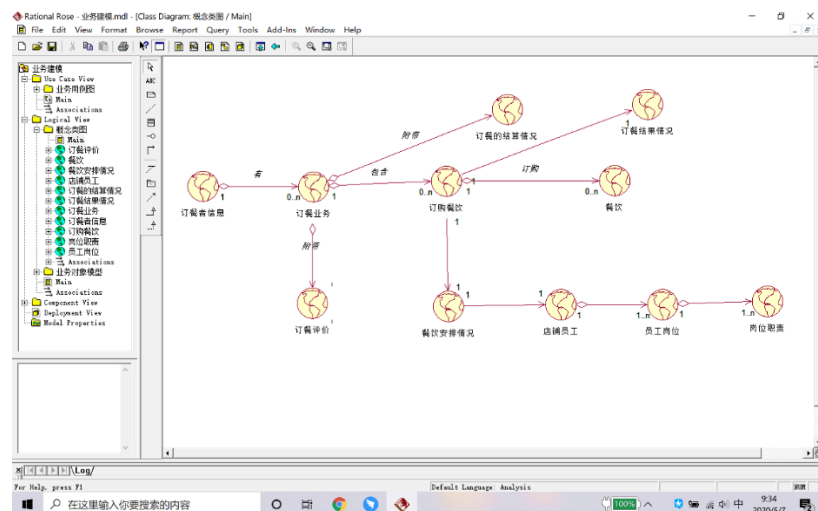
补充约束-非功能需求

无

可支持性

目前只考虑网上支付订餐的情况,应该还要为货到付款留下接口

3.3 概念类图

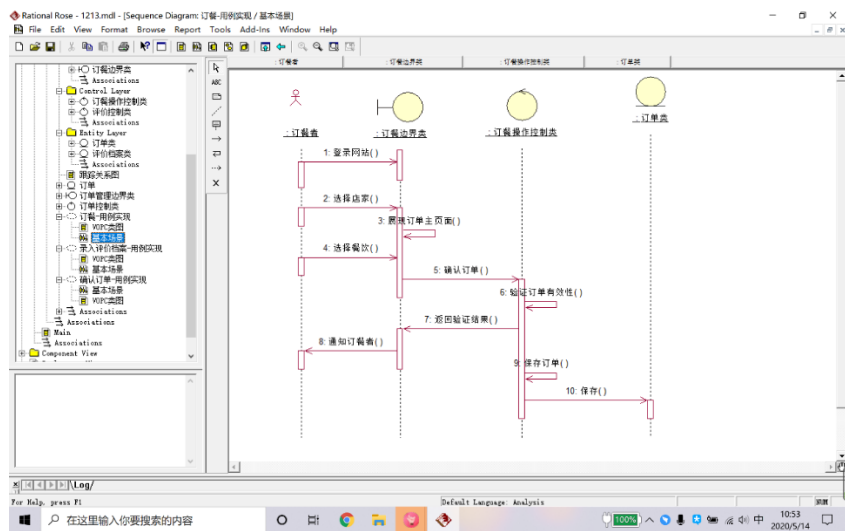


3-3-1 概念类图

4 用例分析

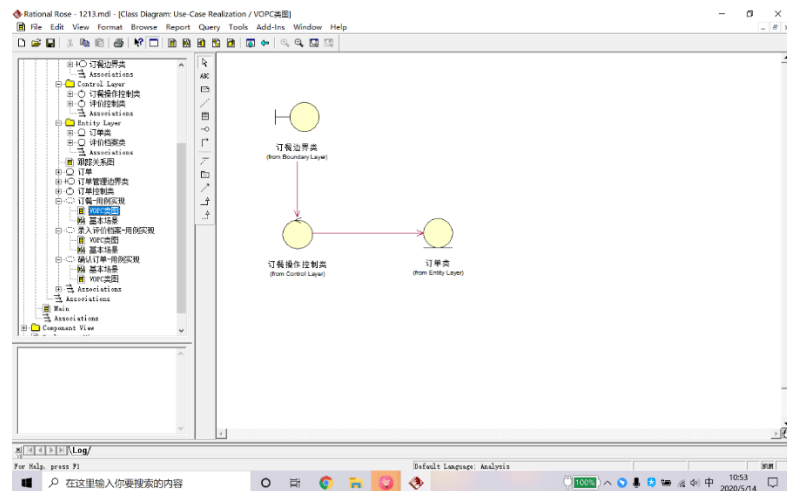
4.1 订餐

顺序图强调消息的时间顺序,用来描述对象间的交互。



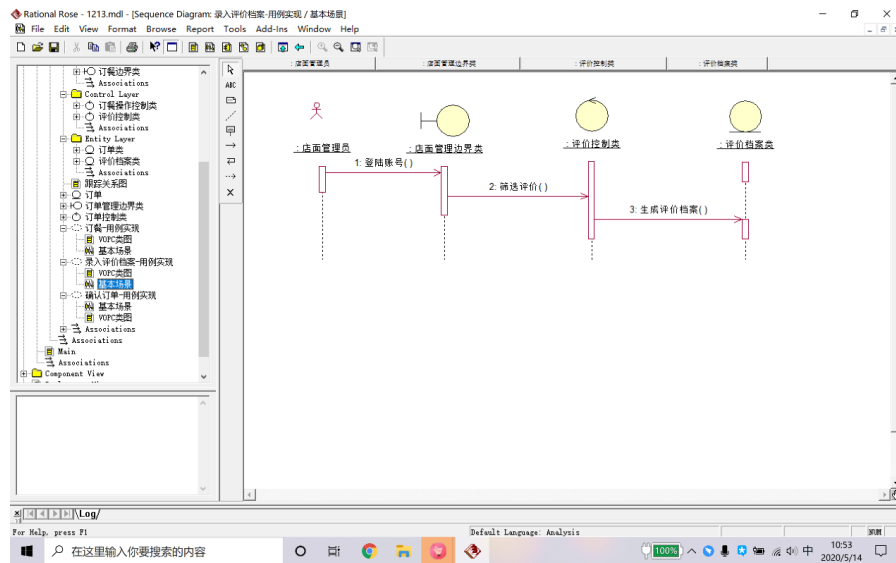
4-1-1 订餐顺序图

VOPC 类图（参与类类图），用来表述参与改用例实现的类及关系

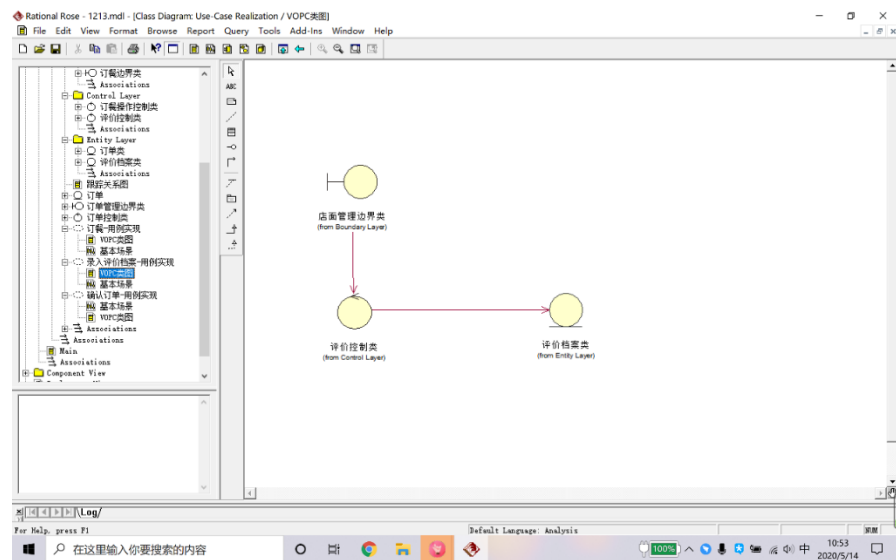


4-1-2 订餐的 VOPC 类图

4.2 评价管理

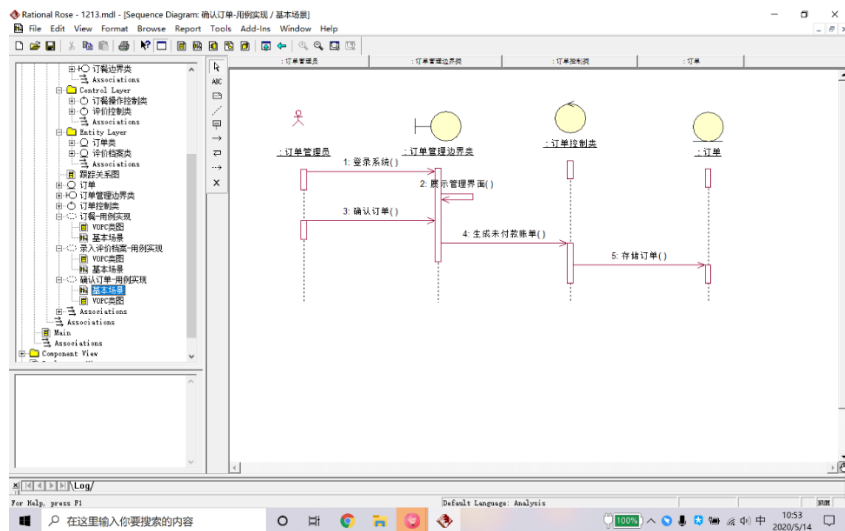


4-2-1 评价管理的顺序图

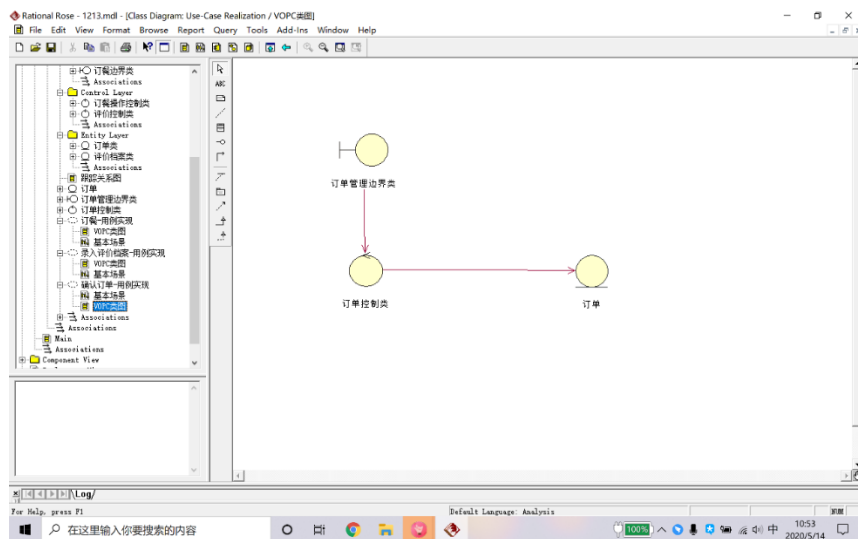


14-2-2 评价管理的 VOPC 类图

4.3 订单管理



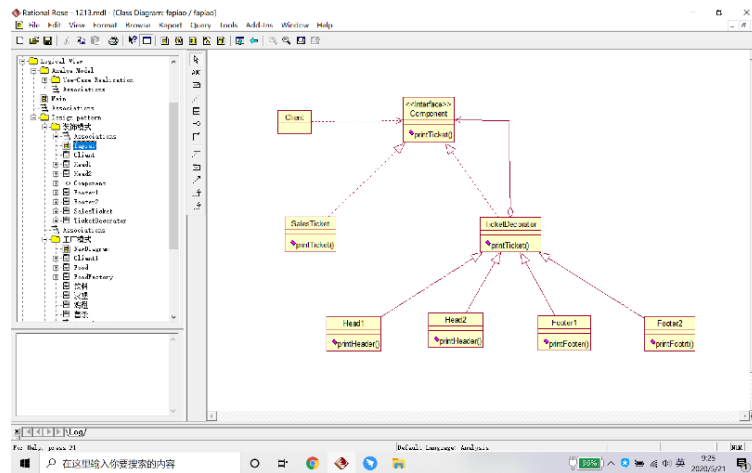
4-3-1 订单管理顺序图



4-3-2 订单管理 VOPC 类图

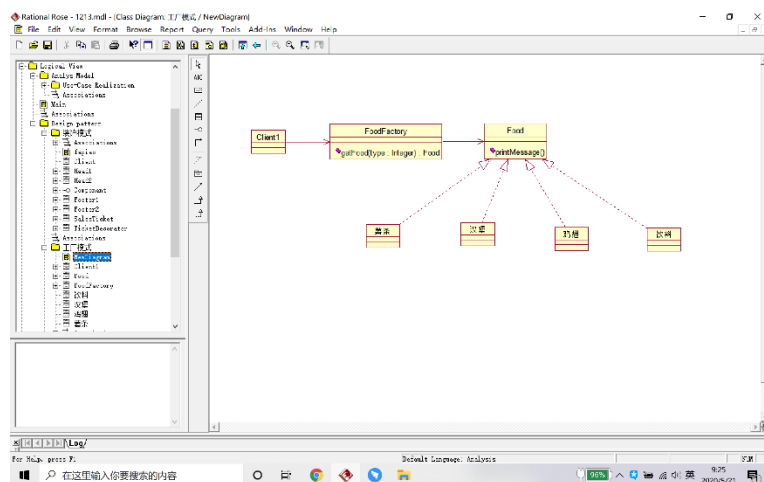
5 设计模式

装饰模式：动态的给一个类增加一些额外的职责
 更灵活的增加扩展对象的功能
 通过一种动态的方式来扩展一个对象的功能
 使用不同装饰类的组合创造不同行为的组合



5-1 装饰模式

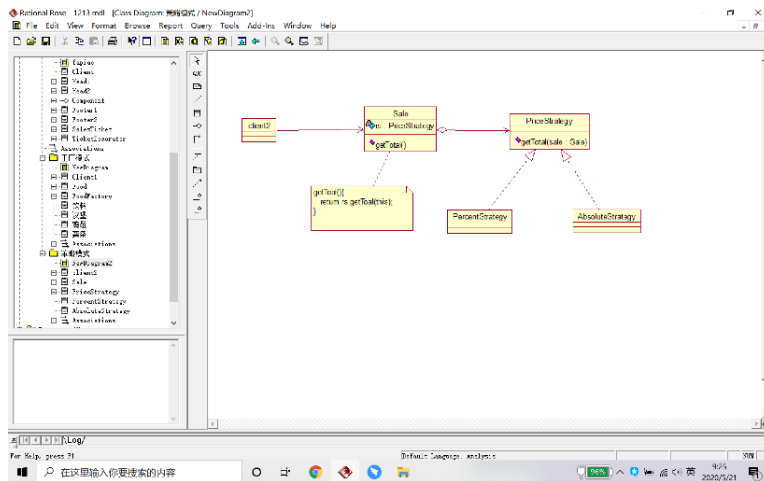
工厂模式：定义一个用于创建对象的接口，让子类决定实例化哪一个类。主要解决接口选择的问题，一个调用者想创建一个对象，只要知道其名称就可以了它的扩展性高。



5-2 工厂模式

策略模式：定义一系列算法，将每一个算法封装起来，并让它们可以相互替换。

- 提供了对“开闭原则”的完美支持
- 提供了管理相关算法族的办法
- 提供了可以替换继承关系的办法



5-3 策略模式

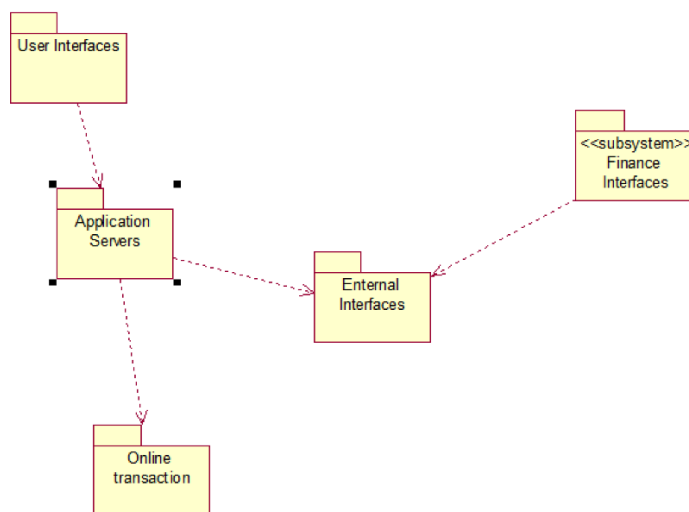
6 架构设计

6.1 系统架构

构架 (Architecture) 设计即在系统的全局范围内, 以分析活动的结果为出发点:

- ◆ 将现有的“分析类”映射成设计模型中的“设计元素”
- ◆ 明确适用于系统的“设计机制”
- ◆ 调整内容逐渐充实为系统“构架”

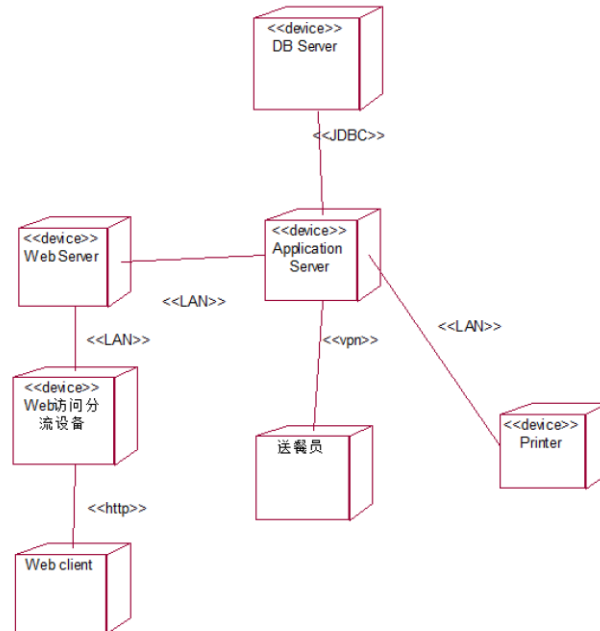
包是一种将模型元素分组的机制, 包主要用于组织模型元素、配置管理单元。



7-1 包图

7.2 系统部署

描述分布(Distribution)活动重点在于开发构架的部署(Deployment)视图。它的目的是说明如何在物理节点间分布系统功能，仅对分布式系统有效。



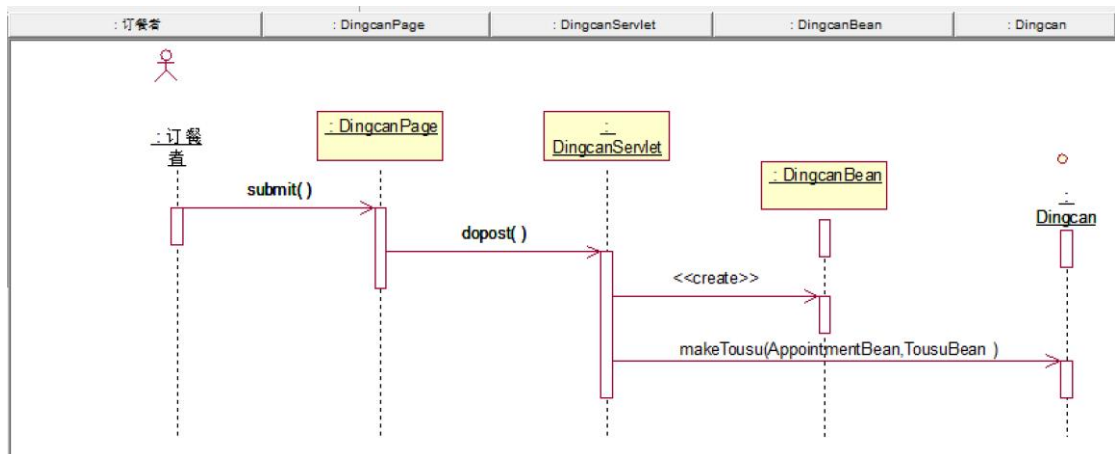
7-2 部署图

7 构件设计

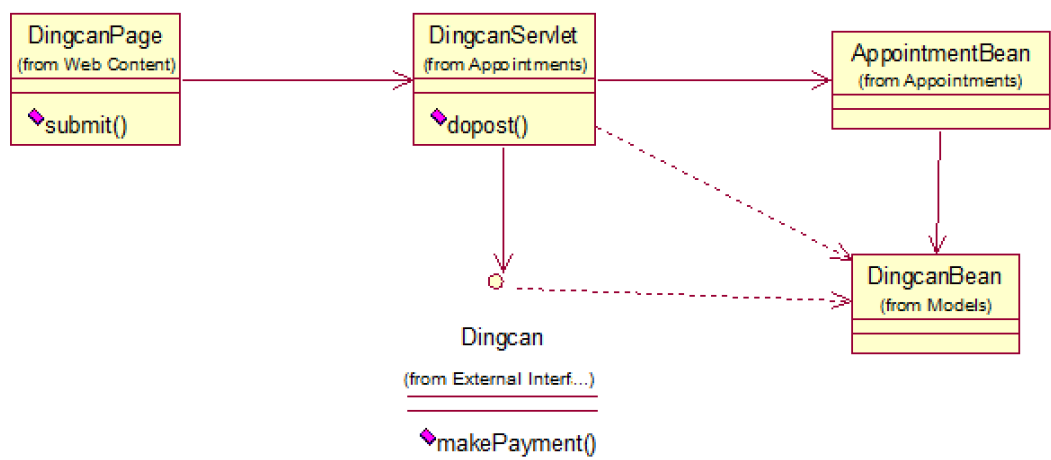
7.1 用例设计

构件设计的任务是基于“构架分析”和“用例分析”的框架，利用“构架设计”提供的素材，在不同的局部，将分析的结果用“设计元素”加以“替换”和“实现”。

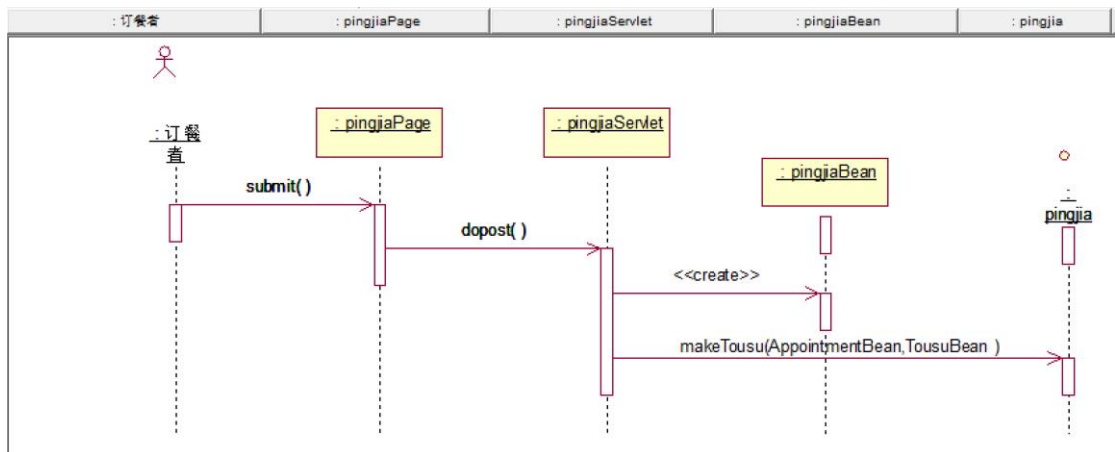
用例设计：利用交互图改进用例实现
改进对设计类的操作需求
改进对子系统和它们的接口的操作需求



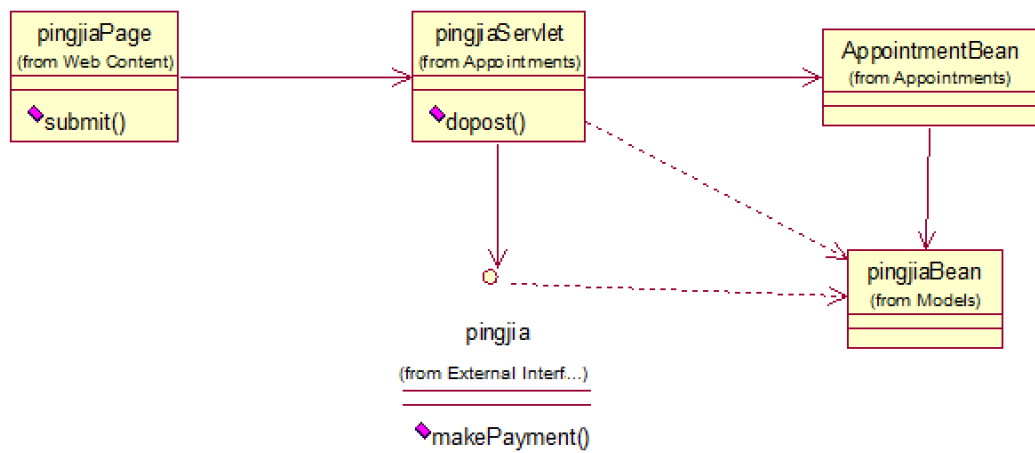
7-1-1 订餐的顺序图



7-1-2 订餐的类图



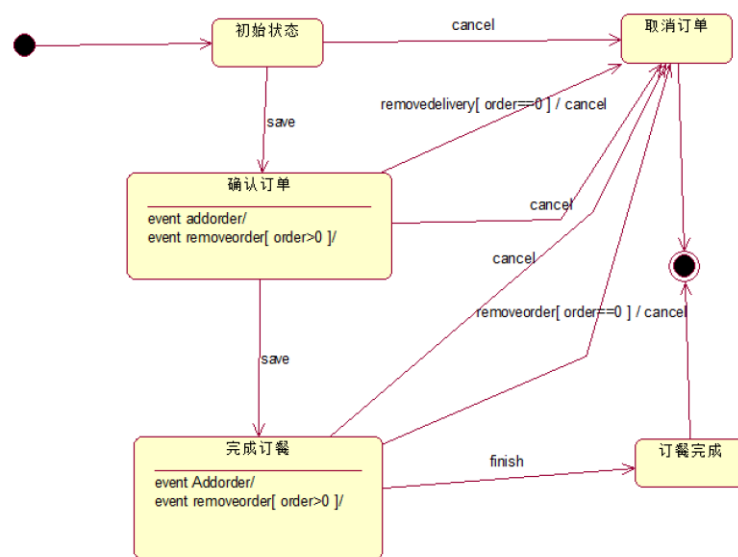
7-1-3 评价的顺序图



7-1-4 评价的类图

7.2 状态图

选择至少一个其实例化对象具有多个状态的类，用 UML 状态图描述其状态变化。



7-2 状态图

7.3 类图

设计类：

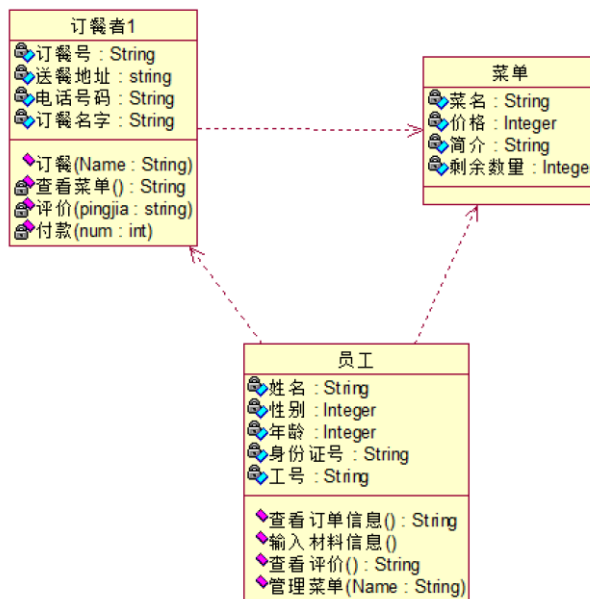
设计模型的构造块

设计类是已经完成了规格说明并且达到能够被实现程度的类

来源于问题域和解域

通过分析类的精化得到的问题域—添加实现细节

解域提供了能够实现系统的技术工具



7-3 设计类图

8 个人总结

这学期的面向对象课程对我来说是收获匪浅的一门课。通过老师课件的讲解，自己一些相关书籍的阅读和实践作业的完成，逐步对课程有了由浅及深的认识。

在修这门课之前，我头脑和实操中写的编代码都还停留在一个小的工程中，或许简单的一个 **project** 就能解决绝大部分问题。但是这一个学期的学习之后，我对软件的开发有了一个新的认识。当我认为有些问题可以通过编程解决的时候，那就不能只放眼于具体的代码实现阶段，而是需要用一个类似于软件构架师的思维和角度去仔细思考和斟酌如何才能设计出一个体系结构相对合理，存储方式比较简洁的产品项目。起初我认为这并不难，我一直将代码的编写放在一个软件的重中之重上，因为我觉着它最费脑筋；但是学习完这门课程之后，我不这么想了。我认为完善的需求分析和设计尽可能少 **bug** 的用例图、UML 类图才是最为关键的。

首先用例图，如果你的需求能够定好，那用例图的设计自然不会太难。我设计时就能根据当时对需求的构想写出用户的操作流，**include** 和 **extend** 也是能够很好地帮助我完成对用例图的设计，表明用户接下来需要做的步骤。最后就是时序图和类图，其实从 **ER** 图开始，我认为接下来的时序图和 UML 类图也同样需要对敲和斟酌，一个小的疏忽可能在你设计的某个时刻被发现，从而带来“翻天覆地”的变化，所以需要不断地返工和完善这两项设计。

直至现在，我也不敢说自己的设计是正确的，因为是第一次进行软件体系的架构，有很多地方一开始不太懂，感觉软件架构的意识不强，导致时序图不知道怎么画，如何入手；不懂 UML 里控制类如何串联起边界类和实体类。不过这正

是学习的过程，如今我感觉对体系架构有了初步的认识，希望自己在以后碰到类似的设计工作时，能够合理的运用学过的东西，逐渐培养起软件架构的意识。