华东师范大学软件学院实验报告

实验课程：OOAD	**年级：**2018/19 级	**实验成绩：**
实验名称：OOA	**姓名：**陈俊潼 / 谢嘉东	**实验日期：**2020.12.24
实验编号：No.3	**学号：**10185101210 / 10185101247	**实验时间：**2 学时

一、实验目的

1/ 练习使用分析模型

2/ 掌握 OOA 的分析方法，体会 OOA 的思想

3/ 体会分析模型间的关系,包括 Domain model, use case, System Sequence Diagram (SSD),operation contract(OC)等.

4/ 理解模型的输入与输出之间的关系

二、实验内容与实验步骤

1. **问题：飞机票预订系统**

   假设：自己能实现的飞机票预定系统，不需要跟外部接触。

   要求：

   - 领域模型
   - 用例模型（给出 2 个用例描述即可）
   - 系统顺序图和操作契约（至少 2 个）

三、实验环境

1. 操作系统：macOS Big Sur 11.0.1
2. StarUML 3.2.2

四、说明

本报告作为第三次实验的基础分析设计报告，我们根据第三次实验相关问题进行了一定的修改。因第三次实验为组队实验，故本报告也为两人合作完成。

五、实验过程与分析

1. 领域模型

   a) 首先使用名词短语识别法找到飞机票预订系统中出现的概念：

      i. 用户 User，系统的用户，可以预定机票。用户需要登录进系统才可以进行机票预订。

ii. 乘客 Passenger，飞机票的使用者，和乘客相关的属性有姓名、身份证号和电话号码。

iii. 支付 Payment，飞机票订单的支付方式，例如微信 Wechat、支付宝 Alipay 等。同时还有一个 Other Payment 的实体用来表示未定义的其他支付等。

iv. 订单 Reservation，飞机票订单，里面存储了和一个飞机票订单相关联的所有信息，包括订票时间、订票乘客等、和预定相关联的乘客、以及和本预定相关联的系统用户（User）

v. 航班 Flight，预定中的航班，有具体的执飞时间

vi. 座位 Seat，航班中的座位，有座位的类型（例如头等舱、商务舱、经济舱）以及对应的剩余数量和售价

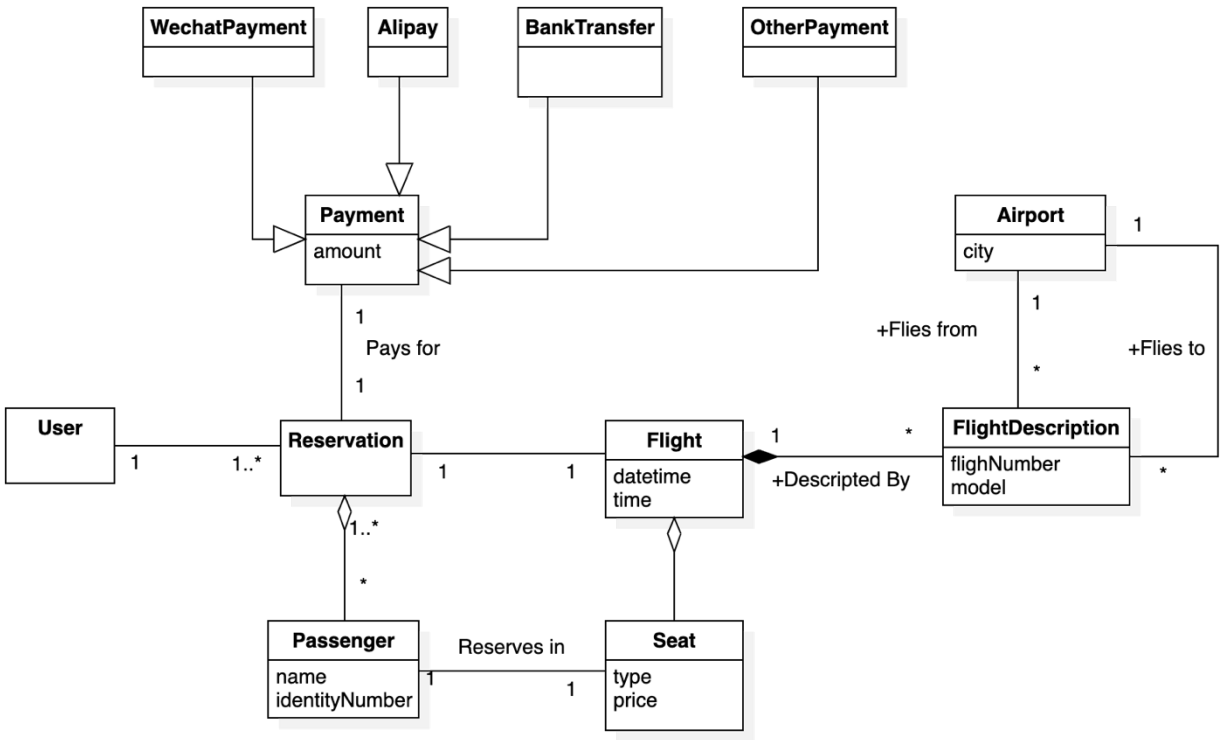
vii. 机场 Airport，给机起飞和降落的地方

viii. 航线 FlightDescription，对固定航班的一个描述，包含航班编号和执飞飞机

b) 然后我们使用概念类名列表查找是否存在遗漏的概念类。

i. 考虑到不需要跟外界接触，而实际上的飞机票预订系统不可能没有支付系统，故我们只是简单实现了接口，而没有实际的作用，即只会反馈相关的信息

c) 这之后我们便可以画出初步的领域模型，如下图所示：

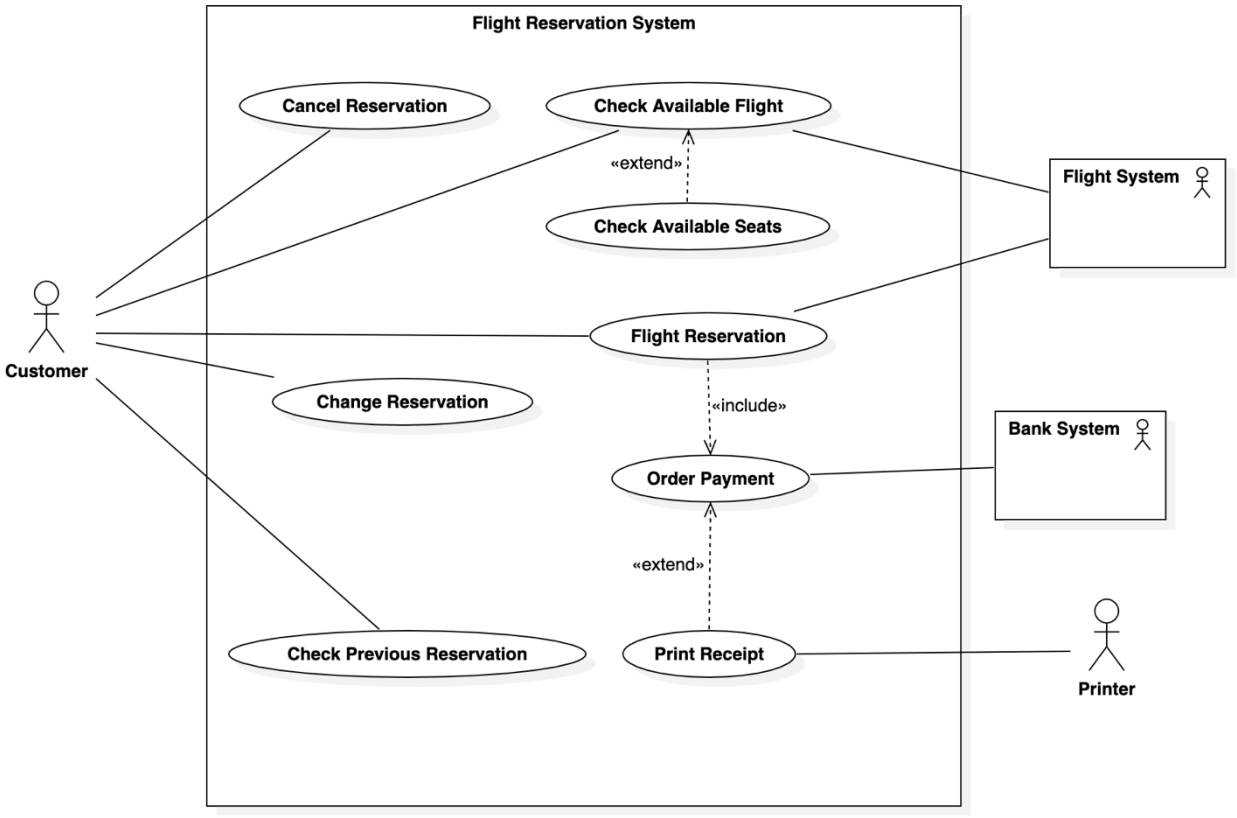


2. 用例模型

飞机票预订系统中，基础的用例有：查找可预定的航班 Check Avaliable Flight、航班预订 Flight Reservation、取消订单 Cancel Reservatiion、改签 Change Reservation、支付 Order Payment；基础的参与者有：顾客 Customer、银行系统 Bank System、机票系统 Flight System、打印机 Printer

考虑到现实系统中用户的实际常用需求，引入查询历史订单 Check Previous Reservation、打印凭条 Print Receipet、查询剩余座位 Check Avaliable Seat 常用用例。

于是可以得到用例模型如下如所示：



a) 用例描述

针对其中的两个用例，需要给出用例描述，处于方便表述与交流的目的，以下描述使用英文表达，相关领域术语、参与者和用例的名称，此前已有中英文对照列出。

Use case description for *Flight Reservation*

Summary	Flight reservation. User can reserve a flight with a given list of passengers and certain flight and seats.

Actor List	Customer, Flight System, Bank System
Precondition	1. The customer is logged in; 2. The flight and seat is available and not locked by other passengers 3. The system is not in maintenance.
Description	The user should input information about passengers of this reservation, and specify the flight number and seat number or level, then the reservation order would be handled by the flight system.
Post Condition	The seat is locked; The reservation will then be finished after the customer pays for the order.
Exception	1. If user not logged in: Request user to log in 2. If the seat is not available: Notify user to change to another seat; 3. If the system is in maintenance: Tell the customer to try again later and save current order in the cache.

Use case description for *Change Reservation*

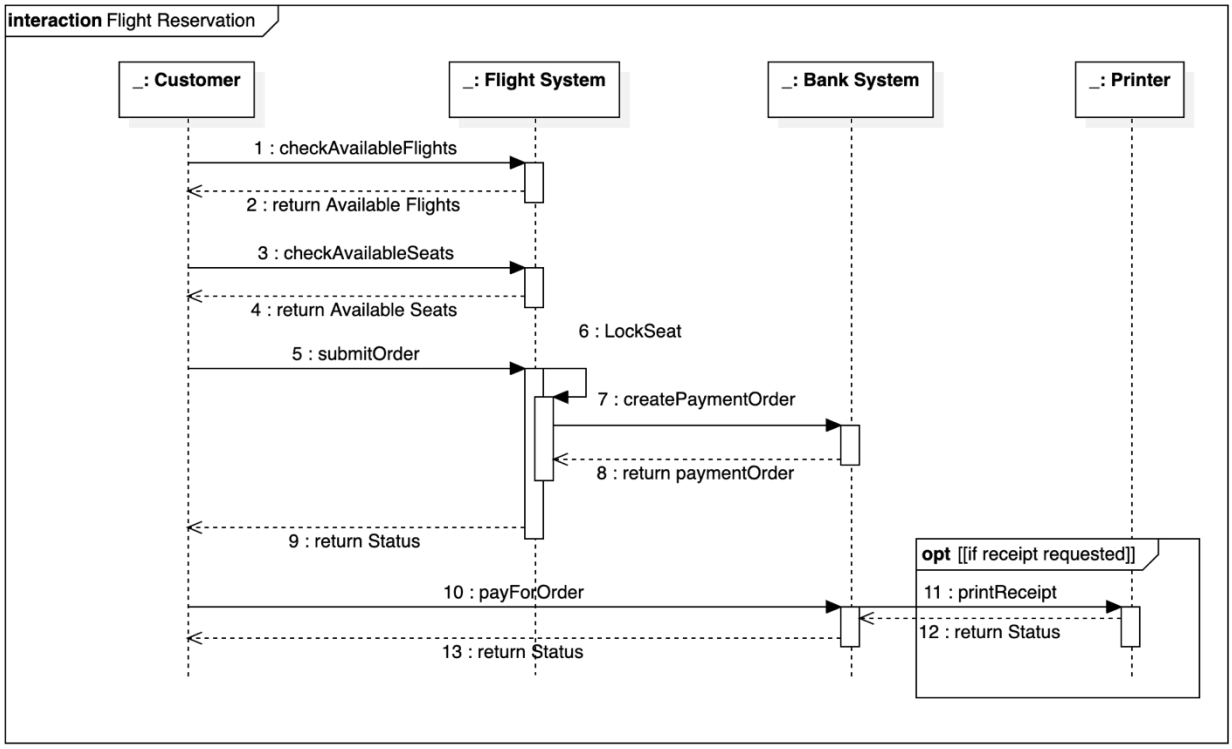
Summary	User can change seats or seat level of previous order.
Actor List	Customer, Flight System, Bank System
Precondition	1. The customer is logged in; 2. There are existing orders of the same customer; 3. The seat that the customer is willing to change is available and not locked by other unpaid customers. 4. The system is not in maintenance.
Description	After confirming there is an existing order and the new seat is available, the reservation order will be updated and a compensation payment order is created based on the new seat price and change fee.
Post Condition	The reservation order is updated. A compensation payment order is

	created.
Exception	1. If user not logged in: Ask user to log in 2. No previous order found: Notify user to check verification info or order number 3. If system is in maintainance: Ask user to try again later.

3. **系统顺序图及操作契约**

选择上述用例中的，机票预订 Flight Reservation 和改签 Change Reservation 两个用例，完成系统顺序图；并分别就系统顺序中展示的各一个系统事件给出操作契约。

a) 系统顺序图一



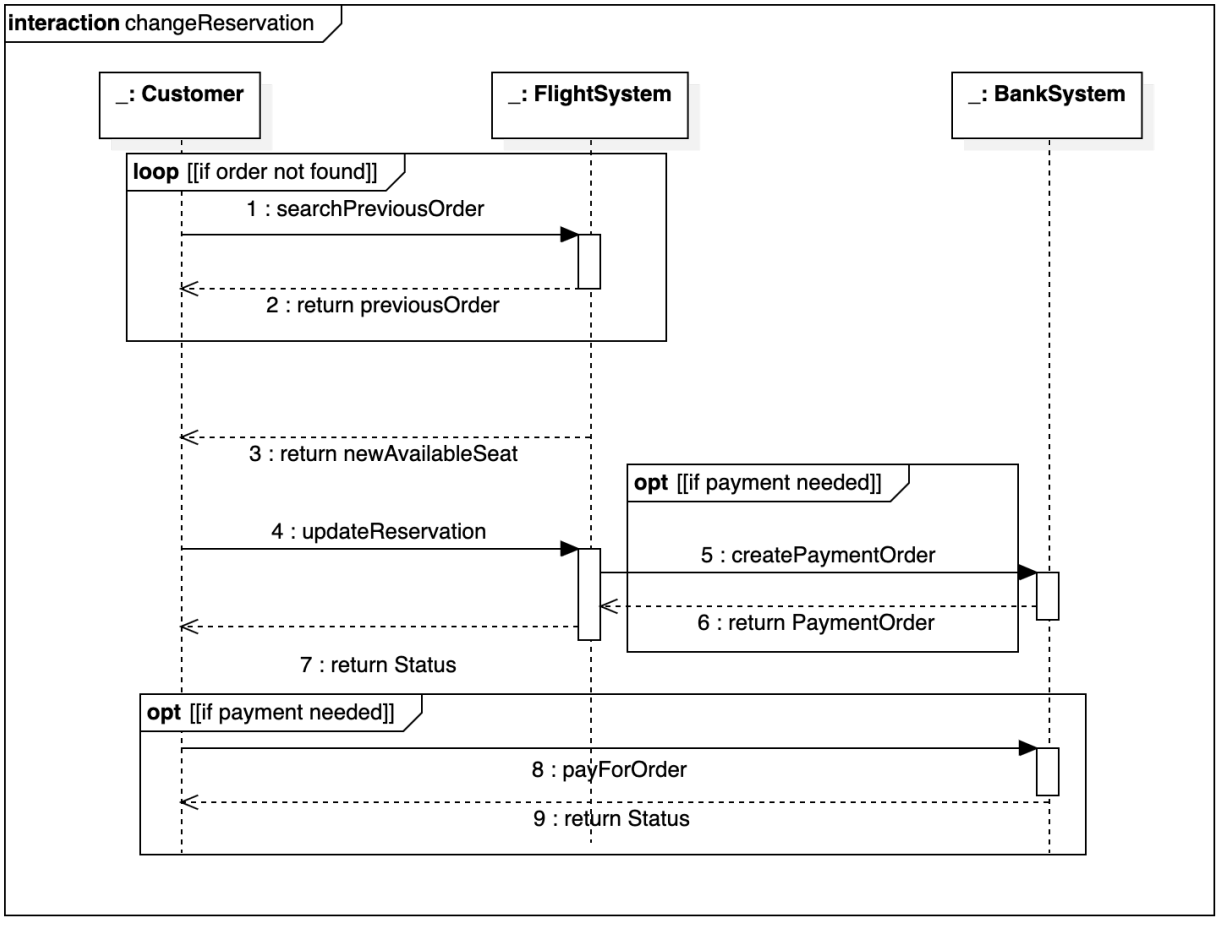
b) 操作契约一

Operation Contrast for *Flight Reservation*

Operation	reserveFlight(flightId: FlightID, seatId: SeatID, customerId: CustomerID)
Cross References	Use Cases: Flight Reservation, Check Available Seats, Check Available Flights, Order Payment

Preconditions	1. The customer is logged in; 2. The flight and seat is available and not locked by other passengers 3. The system is not in maintenance.
Postconditions	1. A new order instance **paymentOrder** is created, associated with the customer. It's price is the amount of the reservation order. 2. A new reservation instance **flightReservation** is created, associated with the customer, the flight, the passengers and corrresponding seats.

c) 系统顺序图二



d) 操作契约二

Operation Contrast for *Change Reservation*

Operation	rchangeReservationFlight(oldReservationId: ReservationID, flightId: FlightID, seatId: SeatID, customerId: CustomerID)
Cross References	Use Cases: Change Reservation, Check Available Seats, Check Available Flights, Order Payment
Preconditions	1. The customer is logged in; 2. There are existing orders of the same customer; 3. The seat that the customer is willing to change is available and not locked by other unpaid customers. The system is not in maintenance.
Postconditions	1. Previous reservation instance **flightReservation** is updated with new seat and flight, associated with the customer: a. The seat is the new seat that the user offered b. The flight is the new flight that the user offered c. The amount equals to the new price of the reservation 2. If change fee is required, a new order instance **paymentOrder** is created, associated with the customer. It's price is the amount needed to compensate for the flight changing.

五、实验结果总结

经过本次实验，我从需求分析开始，对机票预订系统进行了面向对象分析，并使用 StarUML 绘制了领域模型、用例模型、和顺序图。在面向对象的分析过程中，了解了面向对象建模过程中的过程，也尝试遵循面向对象分析设计的种种规范来得到更加合理、易于实现、具备可拓展性和维护性的软件模型。

同时，本次实验也让我能够更加熟练地使用 StarUML 进行建模，了解了软件的使用方法和操作细节，为今后绘制更加复杂的 UML 图和软件模型做了准备。

除此以外，详细的模型分析为我们之后的程序编写提供了极大的便利，也让我们对面向对象的分析设计有了更加深刻的体会。

六、附录

本次实验中绘制的图源文件 FlightReservation.mdj:

https://billc.oss-cn-shanghai.aliyuncs.com/file/2020-12-26-FlightReservation.mdj