华东师范大学软件学院实验报告

实验课程： OOAD	**年级：** 2018 级	**实验成绩：**
实验名称： OOA	**姓名：** 陈俊潼	**实验日期：**
实验编号： No.1	**学号：** 10185101210	**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：面向对象分析：**飞机票预订系统**。要求：

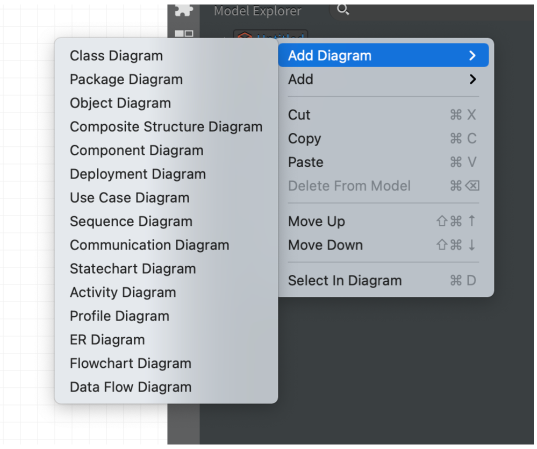
- 领域模型
- 用例模型（给出 2 个用例描述即可）
- 系统顺序图和操作契约（至少 2 个）

三、实验环境

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

四、实验过程与分析

启动 StarUML 中，可以看到在 StarUML 中支持绘制的模型有如下几种：



首先对模型进行分析。本实验中，考虑到现在的大多数订票系统已经是在线的订票系

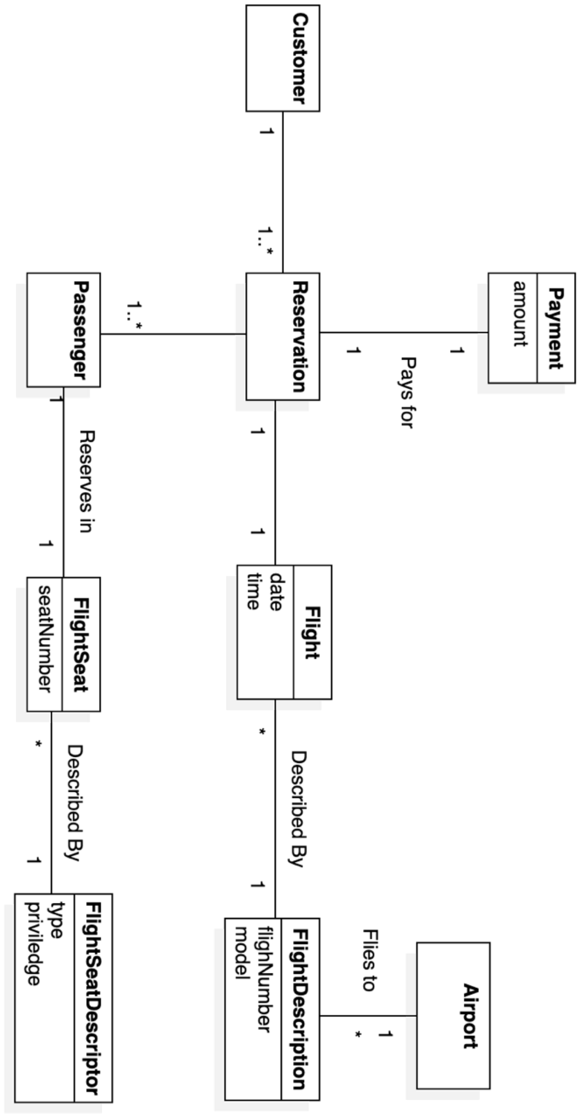
统，故在实验中没有引入前台收银员的概念，而使得用户能够通过网站交互界面直接和订票系统 Flight System 进行交互。此外，没有考虑现金交易的情况，所有的交易都通过网上在线支付的方式，在用户选定了航班和座位号之后以预付费的方式支付。

遵循领域模型的设计准则，使用 FlightDescription 和 SeatDescription 两个类来表示复杂的概念类，来抽象出一个长时运行的航班的目的地、航班号等信息，再使用具体的 Flight 类来和 FlightDescription 和 SeatDescription 相关联来阐释航班信息。

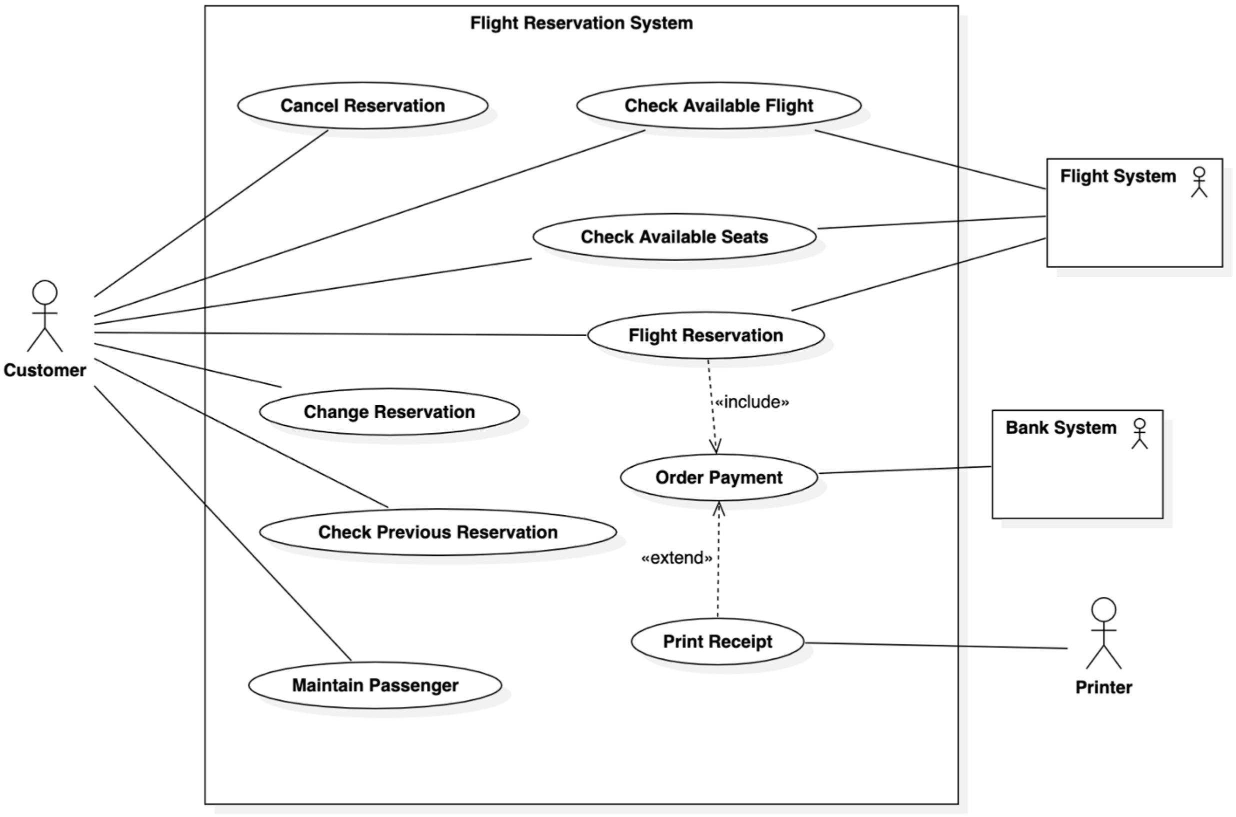
考虑到现实系统中用户的实际常用需求，引入改签、查询历史订单、取消订单、打印凭条等常用用例。

接下来使用 StarUML 分别绘制飞机票预订系统的领域模型（使用类图绘制）、用例模型（Use Case Diagram）、系统顺序图（Sequence Diagram）和操作契约。

1、领域模型



2、用例模型



Use case description for *Flight Reservation*

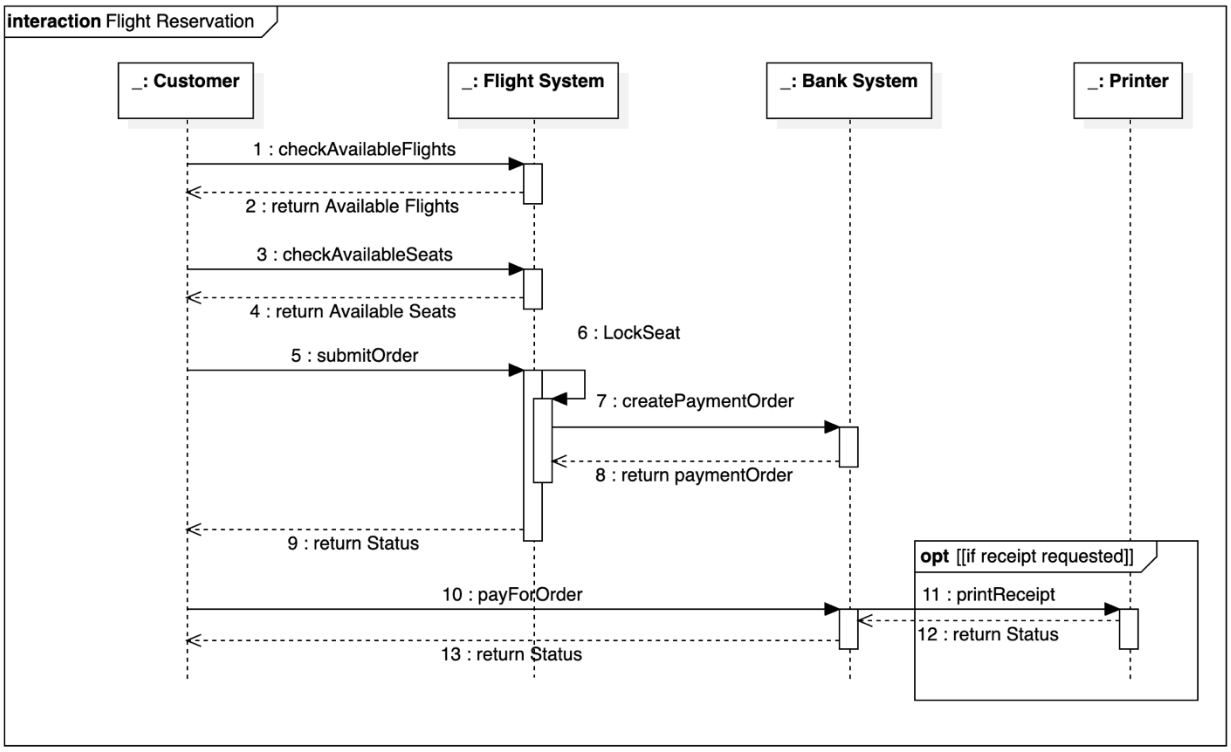
Summary	Flight reservation. User can reserve a flight with a given list of passengers and certain flight and seats.
Actor List	Customer, Flight System, Bank System
Precondition	1. The customer is logged in; 2. The flight and seat is available and not locked by other passengers 3. The system is not in 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*

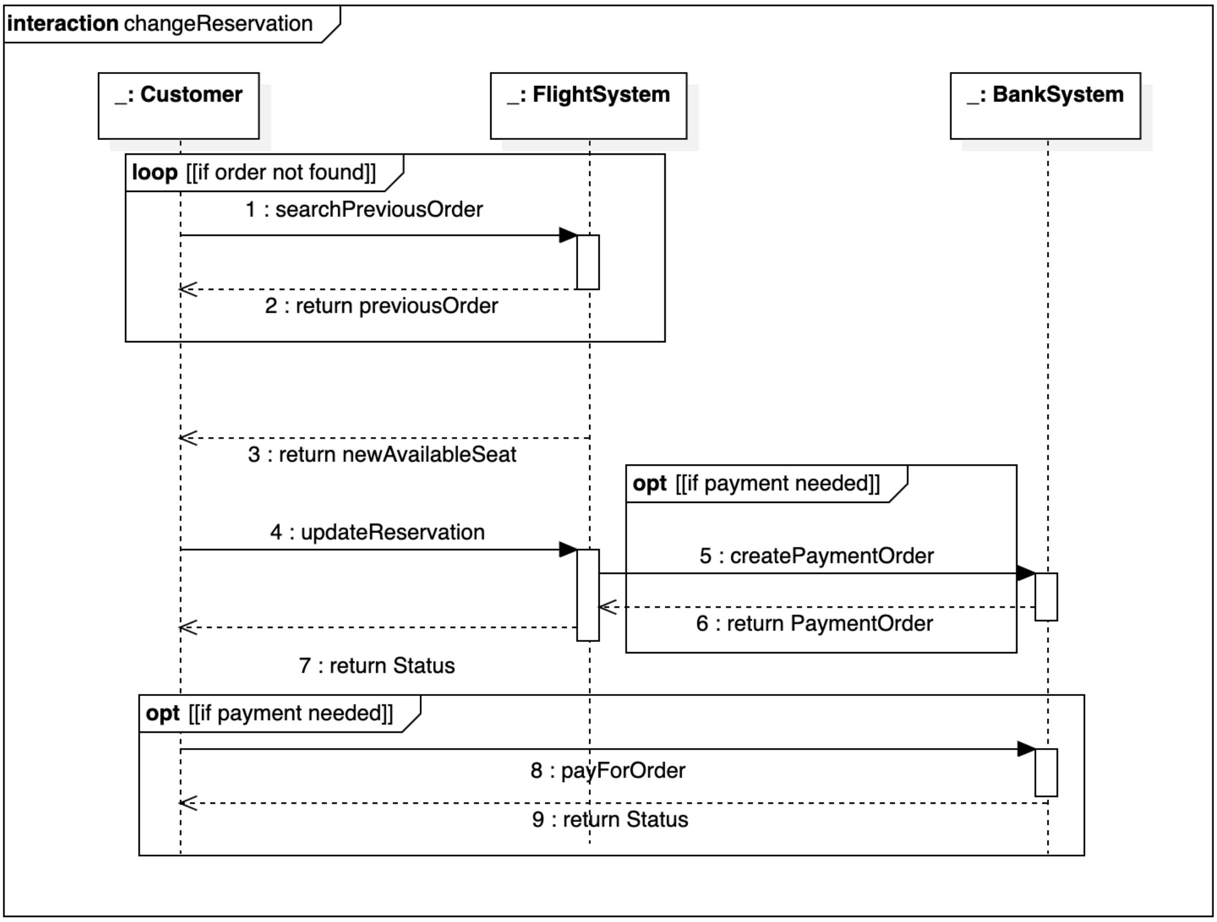
Summary	User can change seats or seat level of previous order.
Actor List	Customer, Flight System, Bank System
Precondition	1. The customer is logged in; 2. There are existing orders of the same customer; 3. The seat that the customer is willing to change is available and not locked by other unpaid customers. 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



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.



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系对象分析，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。

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/2020-11-24-LAB1_FlightReservation.mdj