

Assignment #3

Practice with Models and OOD

CS 4354 Summer II 2016

Instructor: Jill Seaman

Due: at the beginning of class **Monday, 7/25/2016**

Submit a "hard copy" (hand-written and/or computer-generated) only.

Do this assignment with your partner and submit one copy with both names on it.

1 Write a **use case** (including necessary variations) for the following interaction with a (computer simulated) vending machine:

After viewing the available products and prices posted on a vending machine, the customer adds coins, selects a product, and either receives the product (and change) or receives the coins added if insufficient money was supplied or if the product is sold out.

2 Consider the following use case "Add flight and seat to itinerary" from an Airline Reservation system, with initial **CRC cards** for *Passenger*, *Itinerary*, *Flight*, and *Seat*. Carry out a walkthrough of the use case with the cards and add responsibilities and collaborations (and additional cards) as necessary. Your answer should resemble slide 17 from the OODCaseStudy slide, showing the final CRC Cards.

Note: do not add cards for the User Interface (input and output). Assume the customer can communicate directly with the reservation system.

1. The customer selects one of their itineraries (or creates a new one for a given date).
2. The reservation system displays a list of available flights for the date on the itinerary with departure and arrival airports (including departure and arrival times, and price for each flight).
3. The customer selects a flight to add to the itinerary.
4. The reservation system displays the list of available seats on the given flight.
5. The customer selects one of the available seats.
6. The reservation system adds the given flight and seat to the itinerary, and updates the total cost of the itinerary.

3 Consider an online store that enables customers to order items from a catalog and pay for them with a credit card. Draw a simple **UML class diagram** that shows the relationships between these seven classes:

- Customer
- Order
- RushOrder
- Product
- Address
- CreditCard
- Catalog

Be sure to use the appropriate types of associations with multiplicity in your diagram. For the multiplicities, choose from these: {1, 0..1, 0..n, 1..n}

4 Consider the following 2 (incomplete) classes:

```
public class Console {
    private Scanner in;
    private PrintStream out;
    private ApptBook book;
    private void doAdd() {
        out.print("Enter the appointment (descr start end):");
        Appt a = new Appt(in.nextLine());
        book.add(a);
        out.println("Added appointment " + a + "\n");
    }
}

public class ApptBook {
    private ArrayList<Appt> apptList = new ArrayList<Appt>();
    public void add(Appt a) {
        apptList.add(a);
    }
}
```

Draw one **sequence diagram** that shows the method calls of the doAdd and add methods. Activation boxes are required. A complete solution has 6 objects.

5 Consider a program that simulates a vending machine as follows:

After viewing the available products and prices posted on a vending machine, the customer adds coins, selects a product, and either receives the product (and change) or receives the coins returned if insufficient money was supplied or if the product is sold out. Before selecting a product, the customer can press the coin return button to receive all deposited coins. Products are restocked and money removed by an operator who unlocks the machine with a key (if coins have been deposited, they must be returned before the machine can be opened).

Draw a **state diagram** of the different states the vending machine can be in during operation.