Q1: OOD, Contracts and Testing [35 pts]

Below is a UML diagram showing a Rectangle class and associated classes for testing it. Assume a Cartesian coordinate system, where x increases from left-to-right, and y increases from bottom-to-top. Use Java for any code you need to write.

a) [6pts] Provide an implementation for the translate() method of Rectangle class.

```java
public void translate(double deltaX, double deltaY) {
    this.leftTop.setLocation(this.leftTop.getX() + deltaX,
                             this.leftTop.getY() + deltaY);
}
```
b) [4pts] Identify one pre or post-condition for the translate() method of Rectangle class. Remember that a pre/post-condition is a contract or constraint that must be true before/after an operation is invoked.

```java
// At the end of the operation, the left of the rectangle must equal previous
// left plus deltaX specified.

class Rectangle:
  translate(deltaX, deltaY) post:
    getLeft() = @pre.getLeft() + deltaX
```

c) [12pts] Identify two dissimilar invariants and implement their check within check() method of Rectangle class. Simply return false if either of the invariants fail. Remember that an invariant of a class is a predicate that is always true for all instances of that class.

```java
private boolean check() {
    if (this.height < 0) // height should be non-negative
        return false;
    if (this.getRight() < this.getLeft()) // right should not be less than left
        return false;
    return true;
}
```

d) [3pts] Classify the kind of tests that are performed by classes RectangleTest and SampleTestSuite.

These classes implement a unit test for Rectangle class.

e) [10pts] Regarding RectangleTest class, if you need to initialize the same data for each test, you put that data in instance variables and initialize them in a setUp() method. This method is called once before each test is performed. Similarly, the tearDown() method is called once after all tests are finished. Provide the test code for the following methods:

- testTranslate(): tests whether or not the translate() method works properly.
- testNewLeftTop(): tests whether or not setting a new top-left corner works properly.

You may use the assertTrue(boolean condition) method that asserts that a condition is true (i.e., throws an exception if condition is not true).

```java
void testTranslate() {
    double currentLeft = this.rect.getLeft();
    double currentTop = this.rect.getTop();
    double deltaX = 5;
    double deltaY = 10;
    this.rect.translate(deltaX, deltaY);
    assertTrue(this.rect.getLeft() == currentLeft + deltaX &&
                this.rect.getTop() == currentTop + deltaY);
}

void testNewLeftTop() {
    this.testTranslate(); // only way to change left-top is with translate()
}
```
Consider the following problem description for the remaining questions:

Consider an online reservation system (TRS) for a travel company. TRS will initially offer services for travel by bus, train, and airplane. However, in the future, they plan to extend their offerings by cruises in the Mediterranean. Each trip starts from a certain source city and ends in a different destination city, with possible stops in between. Each carrier (bus, train or airplane) is identified by a unique ID (e.g., its plate for a bus). The trips are based on a predefined schedule and they optionally stop at predefined stations/cities. Each carrier can have only one trip per day. Each carrier includes one or two drivers and up to 10 hostesses for each trip depending on the carrier capacity. There are two types of trips, normal trips and express trips. Express trips do not stop at intermediate stations and get faster at the destination.

The customers can make reservations (with a certain time option to buy) and purchase tickets on the website of the company using TRS. However, the company won’t offer a call center to help out customers to reduce costs as everybody has access to the Internet these days. Reservations not purchased by the option date are cancelled automatically by TRS. The customer can pay for the seat through the website before the specified option date. The option dates are to be determined by a dedicated optimization module of TRS. The price of each trip is fixed; however, the company may sell up to a certain number of seats in each trip with a promotion price with feedback from the marketing department. Changes and cancellations are possible for flexible tickets only, with certain penalties as determined by the company for each trip.

The assignments of the personnel to trips are done by company staff in operations department, at most one week before each trip using TRS. For this purpose, TRS will again use the aforementioned optimization module, taking into account the personnel's time-off days and holidays. Such information is to be entered into the system by the human resources department of the company since drivers and hostesses don't have access to and won't use TRS.

The accounting of payments will be handled by a previous software system, used by the company for a number of years.

Both the customers and the company staff must authenticate themselves for performing operations with TRS.

If anything is unclear, please state your assumptions when you answer the following questions.
Q2: OOA and UML [15pts]

Draw a use case diagram for describing the functional requirements of the above system.

Q3: OOD and UML [50pts]

a) [4 pts] List and provide rationale for three design goals for the above system. Briefly discuss if any two of them are in conflict.

Here are some design goals for TRS:

- **Scalability**: The system should be able to serve hundreds if not thousands of customers simultaneously.

- **Security**: The system should be secure to avoid access of private customer information to third parties.

- **Modifiability**: The system should support cruise travel in the future.
b) [9pts] Perform subsystem decomposition for the above system with a UML diagram. State any architectural styles used.

Below is a Component Diagram for TRS that uses client-server architectural style:
c) [20pts] **Perform a quick draft object design** for the above system such that it can support implementation of all the features specified above for only entity objects. Be sure to include all the details (instance variables, methods, and relations between classes) necessary to implement the main functional requirements. You do not need to include getter/setter methods, specify visibility, or provide operation contracts.
d) [5pts] Briefly discuss how at least two distinct design patterns would be appropriate to use in the object design in part c).

Design patterns that could be used in object design of TRS:

- **Singleton**: Could use it for controller classes such as TRSManager and ScheduleManager.

- **Façade**: A Façade class named TRSManager could be used on the server side to delegate all requests to the server to the appropriate module. Also, the optimization module should have a Façade class OptimizationManager that should delegate/adapt optimization requests to appropriate classes. It's very likely that the implementation of this module will be off-the-shelf and is likely to change in the future.

- **Abstraction-Occurrence**: Carrier and Trip classes form an example of an abstraction-occurrence pattern since the same carrier will be used for different trips with distinct per trip data.

---

e) Consider the following scenario for TRS:

Cengiz is a regular customer of a bus company named SpeedyWest that uses TRS. He is a student at Bilkent and final exams are about to start. He would like to leave Ankara for his hometown Izmir as soon as finals end (May 29). He is not sure he will have to attend any re-take exams though as he hasn’t done particularly well during the semester. So he decides to make a reservation for 9am in the morning of May 31st. He successfully does so (with reservation number 123) and is provided with an option to buy his ticket until noon on May 29th. Cengiz enters his last final exam on the morning of May 29th and having done very well on all final exams, he decides to purchase his reserved ticket by providing his credit card details (no: 4567, expiration: 12/2014). Unfortunately his credit card exceeded its limit and the transaction cannot be completed. Then, he calls his father feeling frustrated. You may skip any authentication steps.

- [2pts] What use case(s) does this scenario belong to?

It partly belongs to two separate use cases: "Make A Reservation" and "Purchase Ticket".
- [10pts] Construct a UML sequence diagram that describes this particular scenario in terms of the interaction between software/solution objects as designed earlier in part (c). Feel free to add user interface objects if needed.
Please write your name only in the last page.
Mini Dictionary:

<table>
<thead>
<tr>
<th>Access</th>
<th>Erişim</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aforementioned</td>
<td>Adı geçen, söz konusu</td>
</tr>
<tr>
<td>To assert</td>
<td>İddia etmek, ileri sürmek</td>
</tr>
<tr>
<td>To authenticate</td>
<td>Doğrulamak</td>
</tr>
<tr>
<td>Carrier</td>
<td>Taşıyıcı</td>
</tr>
<tr>
<td>Cruise</td>
<td>Gemi gezisi</td>
</tr>
<tr>
<td>Destination</td>
<td>Gidilecek yer</td>
</tr>
<tr>
<td>Exceed</td>
<td>Sınırı aşmak</td>
</tr>
<tr>
<td>Expiration</td>
<td>Sona erme tarihi</td>
</tr>
<tr>
<td>Favor</td>
<td>Desteklemek, tercih etmek</td>
</tr>
<tr>
<td>Feedback</td>
<td>Geri bildirim</td>
</tr>
<tr>
<td>Flexible</td>
<td>Esnek</td>
</tr>
<tr>
<td>Frustrated</td>
<td>Hayal kıırıklığına uğramış</td>
</tr>
<tr>
<td>Hometown</td>
<td>Memleket</td>
</tr>
<tr>
<td>Human resources</td>
<td>İnsan kaynakları</td>
</tr>
<tr>
<td>To offer</td>
<td>Sunmak</td>
</tr>
<tr>
<td>Source</td>
<td>Başlangıç yeri</td>
</tr>
<tr>
<td>To take into account</td>
<td>Dikkate almak</td>
</tr>
<tr>
<td>Time-off</td>
<td>İzin, çalışılmayan zaman</td>
</tr>
<tr>
<td>Transaction</td>
<td>İşlem</td>
</tr>
<tr>
<td>Unique</td>
<td>Benzersiz</td>
</tr>
</tbody>
</table>

I hereby affirm that the work submitted in this examination is my own exclusively.

Name & Signature: Uğur Doğrusöz  Section: