Question 2

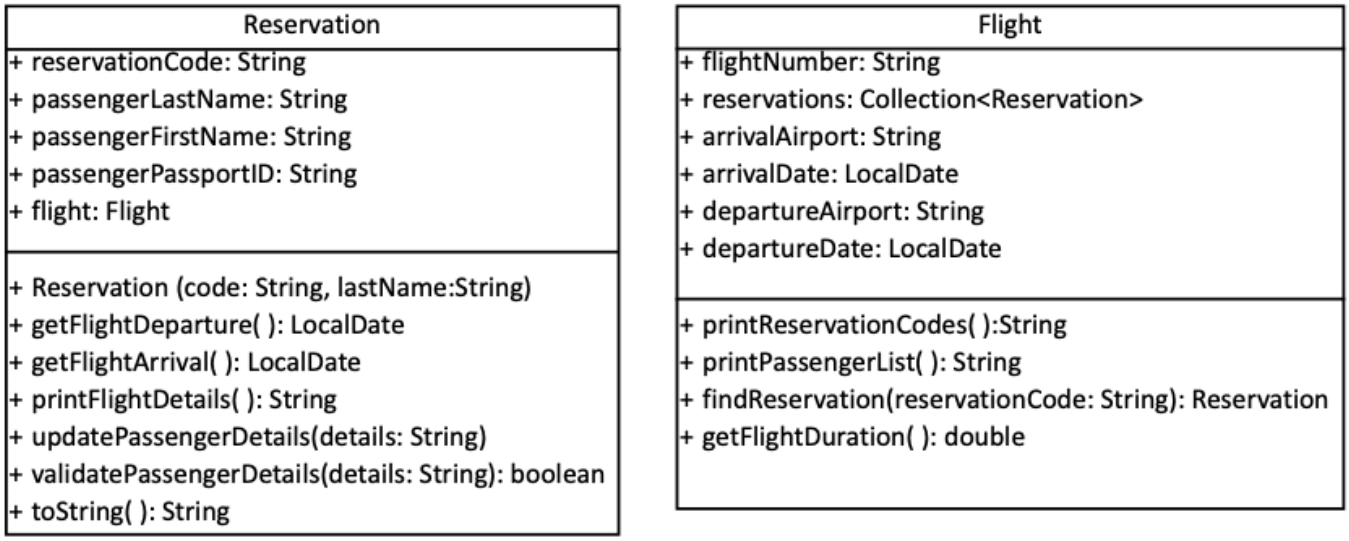

Figure 2

(a) Consider the UML for the Java classes, Reservation and Flight, shown in Figure 2. Analyse the design and comment on cohesiveness of each class and the extent of coupling present between them. [4 marks]

(b) What is the goal of the Model-View-Controller (MVC) architecture? [1 mark]

(c) Identify two advantages of using a model-view separation approach. [2 marks]

(d) Describe the purpose of each component in the MVC architecture. [3 marks]

TOTAL: 10 marks

Question 3

Suppose the ReservationSearch class from Question 1 is implemented where a Collection is used to store Reservation objects based on the class in Question 2.

(a) Write the declaration and initialisation statement(s) for a suitable polymorphic object called reservations that can be used in the ReservationSearch class to store only Reservation objects. [2 marks]

(b) Suppose that a Reservation object is stored in the collection from (a) for a passenger with the surname Booch and reservation code BH193S. Explain how the outcome of the following statement can vary based on the nature of the equals( ) method present in the Reservation class: [4 marks]

```
reservations.contains(new Reservation("BH193S", "Booch") );
```

(c) Write the Java code for a method with the following signature: [4 marks]

```
public String locateReservation(String code, String surname)
```

that locates a Reservation object in the Collection from (a) based on the code and surname supplied and returns the toString( ) details if found, null otherwise.

State any assumptions made which are necessary for your solution.

TOTAL: 10 marks

Question 1 - GUI, Event Driven Programming

The Turkish Airways website allows users to check-in and manage their bookings online by entering a reservation code and surname.

Visit this url to see how it works: https://www.turkishairlines.com/en-int/flights/manage-booking/

Suppose you replicated a portion of the website and some of its functionality in a Java GUI class called ReservationGUI.java that looks similar to Figure 1.

Figure 1

(a) Identify the GUI elements and the corresponding Java classes that you would have to use to achieve the look in Figure 1. [3 marks]

(b) Write working Java code for a complete event handling class that implements the functionality of the red submit button. It should have appropriate methods for associating instances of the ReservationGUI and ReservationSearch class with the event handler. [15 marks]

When the button is pressed, the event handling class should pass the data collected from the ReservationGUI elements to the ReservationSearch instance using a method with the following signature which returns the reservation details if found, or an informative message otherwise:
`public String locateReservation(String code, String surname)`

If the user does not enter a reservation code or a surname and clicks on the red submit button , instructions are shown on the GUI indicating if either piece of data needs to be entered just as in the original on the website.

You may assume:

- There are accessor methods in the ReservationGUI class that return references to the GUI elements in Figure 1 and that they are called `getReservationCodeField()` and `getSurnameField()`
- The ReservationGUI class has a method: `public void displayResult(String result)` that displays the supplied string on the screen.

(c) An interesting feature of the original GUI on the website is the conversion of the user's typed characters to uppercase as they are entered. Identify an appropriate EventListener subtype and method that can be used to achieve a similar result in the ReservationGUI class. [2 marks]

TOTAL: 20 marks