Queen Mary
University of London

BSc Examination by course unit

Thursday ..th April 2015 .. : .. – .. : ..

ECS505U Software Engineering Duration: 2 hours 30 minutes

YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR

Answer all questions

Calculators are not permitted in this examination.

Complete all rough workings in the answer book and cross through any work that is not to be assessed.

Possession of unauthorised material at any time when under examination conditions is an assessment offence and can lead to expulsion from QMUL. Check now to ensure you do not have any notes, mobile phones, smart watches or unauthorised electronic devices on your person. If you do, raise your hand and give them to an invigilator immediately. It is also an offence to have any writing of any kind on your person, including on your body. If you are found to have hidden unauthorised material elsewhere, including toilets and cloakrooms it will be treated as being found in your possession. Unauthorised material found on your mobile phone or other electronic device will be considered the same as being in possession of paper notes. A mobile phone that causes a disruption in the exam is also an assessment offence.

EXAM PAPERS MUST NOT BE REMOVED FROM THE EXAM ROOM

Examiners: Dr. Mustafa Bozkurt and Prof. Martin Neil

Question 1

The requirements below partially describe an airline system in which flight, ticket and passenger information is stored and manipulated. The system enables users to buy tickets, select seats and for passengers to check their itinerary. Draw a UML class diagram using the list of requirements below.

[20 Marks]

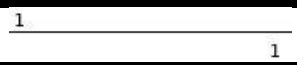
1. A passenger can hold one or more bookings.
2. A booking can only belong to a single passenger.
3. A booking is for a single flight.
4. A flight can have zero or more bookings.
5. A flight can only be carried out by one plane.
6. A plane can be assigned to zero or more flights.
7. Planes have one or more plane seat.
8. The system must not allow booking a plane seat that does not belong to a plane.
9. There are two types of plane seats: economy and business.
10. A flight has one or more flight seat.
11. A plane seat can be assigned to many flight seats.
12. A flight seat holds (is for) a single plane seat.
13. A booking is for a single flight seat.
14. A flight seat can be assigned to one booking.
15. The system must allow empty flight seats.

HINTS:

- A “flight seat” represents an available seat in a flight.
- A “plane seat” represents a physical seat in a plane.
- In order to fulfil RQ7 and RQ8, the system must make sure that all seats of a plane will be removed from the system when the plane is removed.
- A flight seat is considered empty (unbooked) if it is not assigned to a booking.

Please use the names below for the class names in your diagram.
Passenger, Booking, Flight, Plane, FlightSeat, PlaneSeat, Economy, Business

Please use the necessary class diagram elements in the format given below in your diagram. Marks will be reduced for every element which differs from the given format.

Bidirectional association	Unidirectional association	Aggregation
		

Unidirectional aggregation	Composition	Unidirectional composition
		

C	Range	Class
	4..*	

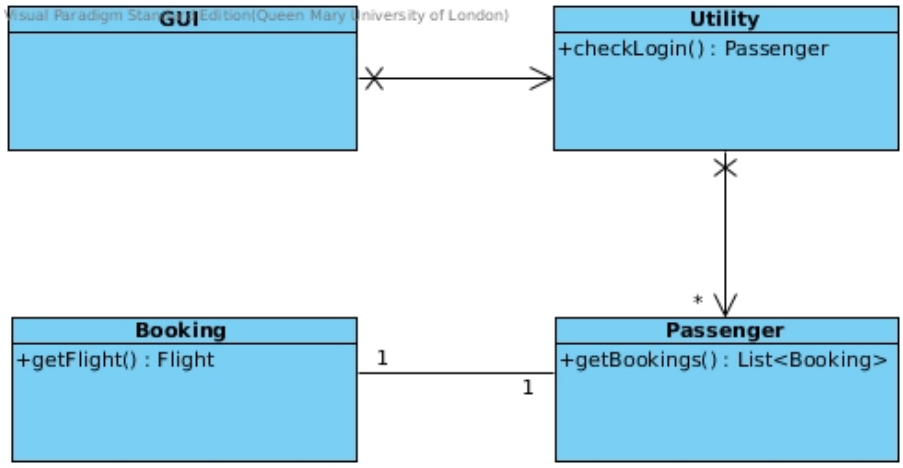


Figure 1: Class diagram for Question 2

Question 2

The UML class diagram in Figure 1 is part of the design of a ticket booking system. Draw a UML sequence diagram using the class diagram and the scenario below. In this scenario, we assume that the GUI (an object representing the graphical user interface of the system), utility (instance of Utility), passenger (instance of Passenger) and booking (instance of Booking) objects are already created and exist in the system.

[18 Marks]

1. A passenger makes an itinerary request using the GUI.
2. The GUI calls the utility object to check the login of the passenger and assigns the return value to a local variable “aPassenger”.
3. If login attempt is successful (if “aPassenger” is different than null) then the GUI gets bookings from the passenger object.
4. For all the bookings in the list, the GUI calls booking object to get its flights.
5. The GUI displays itineraries to the passenger (actor).
6. If login attempt is unsuccessful then the GUI displays error message to the passenger (actor).

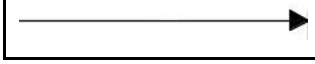
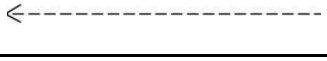
HINTS:

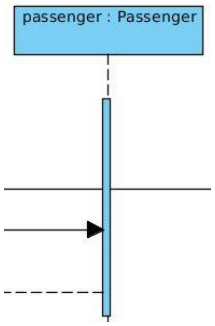
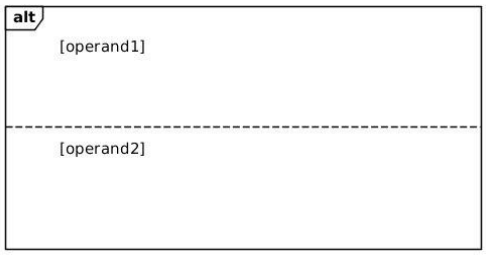
- Use appropriate “message” and “return message” diagram elements to represent interactions between the GUI and actor.
- Use different than operator “!=” in your operand for RQ3.
- For RQ4, use a range for the operand where “number of bookings” represents the number of bookings in the list. Follow the range format depicted in the question 1.
- Use “List<Object>” format for return messages where a list of objects must be returned.
- Beware, some of the return messages are implied not explicitly stated.

Continued on the next page…

Please use the names below for the actor name, object names and messages in your diagram.
Itinerary request, Display itinerary, Display error message, utility, GUI, passenger, booking

Please use the necessary class diagram elements in the forms given below in your diagram. Marks will be reduced for every element which differs from the given format.

Message	Return message	Actor
		

Timeline	Alternative fragment	Loop fragment
		

Question 3

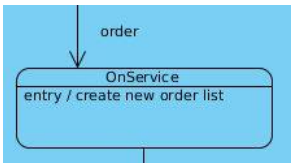
Draw a UML state machine diagram using the following list of specifications. The specifications describe the states of a flight seat in the ticket booking system.

[12 Marks]

1. Initially, seats of a flight are empty.
2. An empty flight seat is selected for booking when a passenger starts the booking process.
3. If a passenger cancels the booking process, the selected flight seat becomes empty.
4. When the passenger completes the booking process the selected flight seat becomes booked.
5. If a passenger changes a flight seat that he/she booked then the flight seat becomes empty.
6. If a passenger cancels his/her booking then the booked flight seat becomes empty.
7. When a flight is closed (ready for take-off) all its empty flight seats are removed from the system.

Please use the names below for the state names and transitions in your diagram.
Empty, Selected, Booked, selectSeatForBooking, cancelBookingProcess, completeBooking, changeSeat, cancelBooking, closeFlight

Please use the necessary class diagram elements in the forms given below in your diagram. Marks will be reduced for every element which differs from the given format.

Initial state	State and transition	Final state
		

Question 4 (Total 50 Marks)

a) Explain the three software engineering design principles modularity, abstraction and encapsulation. **[8 Marks]**

b) Explain what software design patterns are and discuss the merits and drawbacks of using design patterns. **[7 Marks]**

c) Explain what domain analysis is and why it is important. **[9 Marks]**

d) Suppose you are asked to test a software component with a public method that converts a given temperature in Celsius degrees (°C) to Fahrenheit degrees (°F). You don't have access to the source code however specifications of the method are available. Specifications of the method are as follows:

The method has only one input which is a 32-bit integer.

The return type of the method is a 32-bit integer.

The method only converts temperatures between -100°C and 100°C.

If the given input temperature is 100°C or more the method returns -99999.

If the given input temperature is -100°C or less the method returns -99999.

d1. Briefly explain why performing exhaustive testing (testing with all possible inputs) on this method might not be considered efficient testing and propose an efficient testing strategy. **[5 Marks]**

d2. Explain how the proposed testing strategy is performed and list the minimum amount of test inputs required to perform it. **[5 Marks]**

e) Explain what project management is and briefly discuss three of the challenges in project management. **[8 Marks]**

f) Explain what metric and measure are and briefly discuss the challenge of using the software metric "number of lines of code" as a measure of size. **[8 Marks]**

End of Paper