COMP201: Software Engineering I
Object Oriented Design
Coursework Assignment 2 (2023/2024)

Modelling with UML

Assessment Information

Assignment number	2 of 2
Weighting	20%
Assignment Circulated date provided to class	21/11/2023
Deadline Day & Date & Time	18ʰ of December 2023 at 17:00
Submission Mode	Electronic
Learning outcome assessed	be fully aware of the principles and practice of an O-O approach to the design and development of computer systems; be able to apply these principles in practice.
Submission necessary in order to satisfy Module requirements	No
Purpose of assessment	To assess the students ability to produce an OO design in UML.
Marking criteria	See end of document
Late Submission Penalty	Standard UoL Policy

REPORT
Title page: put your name, your student number and the course on the first page.

TASK 1. (25%) Given the following informal specification, **identify good candidates** for classes and attributes, and **identify things that are outside** of the problem domain. You should use the noun identification technique and show your working. Also identify **all potential inheritance relationships**. You should ensure that data is **NOT** duplicated across classes even if a user places multiple bookings. Use the noun identification method of class elicitation for the first pass. For full marks please try and avoid as much as possible duplication of data between classes and within them.

In your submission show all your working (noun analysis)

Present your design as a class diagram, including all relevant attributes and relationships

> *Your customer is a travel agency that wants a reservation system that will run on the Internet. This reservation system will allow clients to keep track of all their travel reservations for airlines, hotel, rental cars and travel insurance. The client must enter the names of all his/her traveling companions, but all reservations will be referenced by the primary client. The system needs to make it easy for a client to have multiple reservations. All reservations will include a booking number as well as a reference to the names, passport numbers and dates of birth of all the travellers involved in the reservation. The system should also have an address for the primary client. Airline reservations will include the airline, flight number, class of seat and travel date and time. Hotel reservations will include the type (twin, single, double) of room, the start date and the end date as well as the name and address of the hotel. Car rental reservations will include the type of car requested, start date, days of hire and the drivers' license numbers. The insurance booking will have a start date, an end date and level of cover bronze, silver or gold.*

TASK 2. (25%) You are required to draw a UML **activity diagram** to represent the following scenario of a hairdresser's salon.

> *Customers enter the salon and wait until the next hairdresser is free. They then indicate whether they would like their hair washed first or a "dry-cut" without having their hair washed. The hairdresser washes the hair (if asked for) and then cuts it. After finishing the customer's hair the hairdresser moves onto the next waiting customer, or waits for another one to enter the salon. The customer goes to the till and waits for a cashier to be free to take their payment. They can pay by either cash or by credit card (where they need to type their pin into the machine) and they then leave the salon.*

TASK 3. (25%) Read the following passage **carefully**.

> *An employee has a name, address, phone number, dateofbirth and job title. Employees can be appointed and can leave, and are either monthly paid employees or weekly paid employees.*

> *Monthly paid employees have a bank sort code, bank account number and number of holidays while weekly paid employees are paid in cash on a specified day of the week - their payday. Weekly paid employees may apply to be promoted to a monthly paid employee. Monthly paid employees can take a holiday if they have sufficient number of holidays remaining.*

All employees are entitled to use the <u>Sports Centre</u> if they <u>register</u> to do so. The Sports Centre is made up of two <u>gyms</u> (with a <u>maximum capacity</u>), three <u>tennis courts</u> and a <u>bar</u>.

The bar can be <u>booked</u> for special events, and has three rates of hire - a <u>working hours' rate</u>, an <u>evening rate</u> and a <u>weekend rate</u>. The Sports Centre holds a <u>list of employees</u> who have registered.

An employee's <u>age</u> can be <u>calculated</u> from their date of birth, in order to prevent under-age drinking at the bar.

You are required to draw a UML **class diagram** for the above system. All the keywords you need to include are underlined – do *not* invent any details additional to those given above:

1. Illustrate the various classes that exist, with their attributes and operations (including any derived ones, represented in the usual way)
2. Mark on the relationships that exist between the classes using the standard UML symbols to represent the *type* of each relationship
3. Add multiplicities
4. for any relationships of **association:**
   a. mark on the navigability
   b. appropriately name the two roles

TASK4. (25%) Draw a UML **sequence diagram** that specifies the following protocol of initiating a two-party phone call. NOTE: ArgoUML does not fully support Sequence Diagrams, it maybe better to use a different program (such as OpenOffice Draw/ Microsoft Powerpoint) or (neatly) draw the diagram by hand. Let us assume that there are four objects involved:

- two Callers (s and r),
- an unnamed telephone Switch, and
- Conversation (c) between the two parties.

The sequence begins with one <u>Caller</u> (s) sending a message (<u>liftReceiver</u>) to the <u>Switch</u> object. In turn, the Switch calls <u>setDialTone</u> on the <u>Caller</u>, and the <u>Caller</u> iterates (7 times) on the message <u>dialDigit</u> to itself. The dialled digits are then sent to the <u>Switch</u>. The <u>Switch</u> object then calls itself with the message <u>routeCall</u>. It then creates a <u>Conversation</u> object (c), to which it delegates the rest of the work. The <u>Conversation</u> object (c) <u>rings</u> the <u>Caller</u> (r), who asynchronously sends the message <u>liftReceiver</u>. The <u>Conversation</u> object then tells both <u>Caller</u> objects to <u>connect</u>, after which they talk. Once <u>Caller</u> (r) sends a <u>disconnect</u> message to <u>Conversation</u> then <u>Conversation</u> tells both <u>Caller</u> objects to <u>disconnect</u> and also it tells the <u>Switch</u> to <u>disconnect</u>. After that <u>Switch</u> deletes the object <u>Conversation</u>.

All the keywords you need to include are underlined – do *not* invent any details additional to those given above.

Marking Criteria

Task	A++ to A 70%+	B 60%-69%	C 50%-59%	D 40%-49%	E+ 35%-39%	E- to G < 35%
1	Well chosen classes for the scenario and a description of any removed superfluous classes. Correct inheritance used.	Mostly correct classes chosen but without solid justification. Inheritance relations denoted correctly.	Inappropriate classes or inheritance relations defined with minor omissions and/or poor justification of chosen classes.	Major omissions of classes and incorrectly chosen/missing inheritance relations.	Some understanding of classes and how they should be derived but with a poor choice of classes and no justification	No evidence of understanding the concept of deriving classes from a scenario.
2	Correct notation and a good level of abstraction used throughout	Mostly correct notation but with minor errors and/or minor omissions for the modelling of the scenario	A good attempt to model the scenario but with slightly incorrect notation and an inadequate level of detail.	Insufficient level of detail but some evidence of correct understanding of activity diagrams.	Some evidence of understanding activity diagrams and an attempt to model the scenario in some meaningful way.	No serious attempt to model the scenario with an activity diagram.
3	Correct notation and a good level of abstraction used throughout	Mostly correct notation but with minor errors and/or minor omissions for the modelling of the scenario	A good attempt to model the scenario but with slightly incorrect notation and an inadequate level of detail.	Insufficient level of detail but some evidence of correct understanding of class diagrams.	Some evidence of understanding class diagrams and an attempt to model the scenario in some meaningful way.	No serious attempt to model the scenario with a class diagram.
4	Correct notation and a good level of abstraction used throughout	Mostly correct notation but with minor errors and/or minor omissions for the modelling of the scenario	A good attempt to model the scenario but with slightly incorrect notation and an inadequate level of detail.	Insufficient level of detail but some evidence of correct understanding of sequence diagrams.	Limited evidence of understanding sequence diagrams and their uses.	An inadequate attempt to use sequence diagrams in a meaningful way.