Queen Mary
University of London

Main Examination Period 2017

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. Please state on your answer book the name and type of machine used.

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, smartwatches 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 describes a student housing management system that is used for performing day-to-day operations such as renting available rooms to students, searching for empty rooms and adding new students to the system. The system keeps track of students, buildings, available and rented rooms and rent contracts. Draw a UML class diagram describing an object-oriented system using the list of requirements below.

[23 Marks]

1. The system must be able to determine the status of buildings and rooms.
2. The system must be able to determine the status of rented and available rooms (free to rent).
3. There are three types of rooms in all buildings: single, double and triple.
4. The system must be able to determine status of students and their contracts.
5. Each student signs one contract based on the room they rent.
6. Each student can rent only one room.
7. There are three types of contracts: single, double and triple.
8. A single room can be rented (contracted) to one student.
9. A double room can be rented to two students.
10. A triple room can be rented to three students.
11. The system must store details of each contract type.
12. The system must determine the status of the institution each student attends.
13. The system must store the address of buildings and institutions.
14. All buildings are stored in the building inventory.

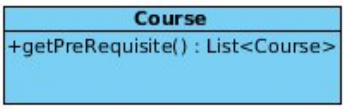
HINTS:

- No need to add any methods or attributes to your classes.
- Buildings consist of rooms. In your design, consider whether to keep the rooms of a deleted building in the system or not.
- Details for each type of contract are the same and the system will contain many instances of each type. Think of an efficient design pattern to handle this.
- Expect to have many students from each institute in the system.

Please use the names below for the class names in your diagram.
Building, Room, SingleRoom, DoubleRoom, TripleRoom, BuildingInventory, Address, Institution, Student, Contract, SingleContract, DoubleContract, TripleContract, SingleContractDetails, DoubleContractDetails, TripleContractDetails

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	Composition	Aggregation
		

Generalisation	Multiplicity	Class
	4..*, 0, *	

Question 2

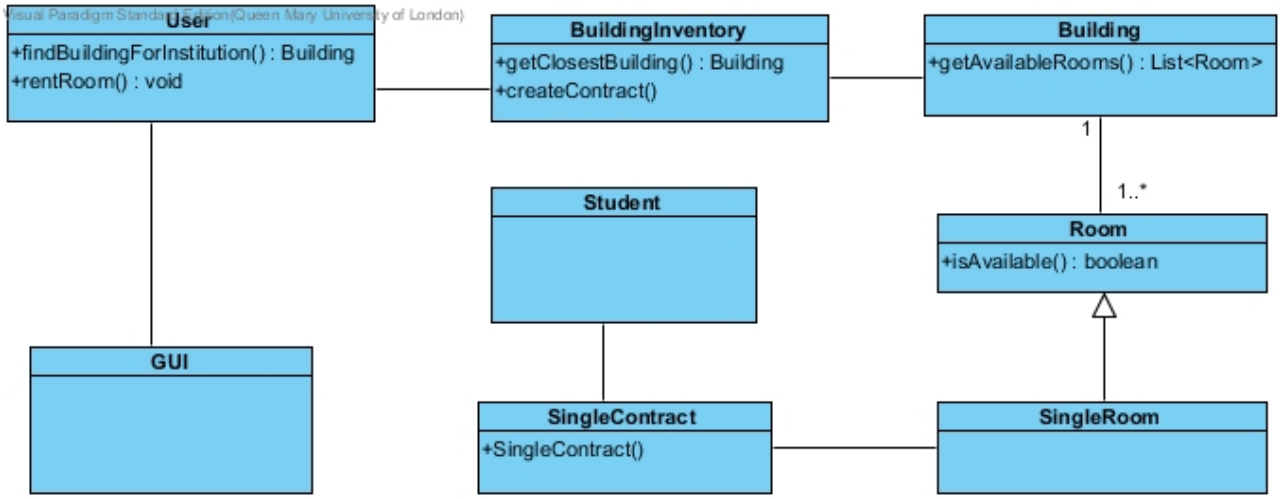


Figure 1: Part of system class diagram (This is not the answer to Question 1)

The UML class diagram above is a part of the design of a student housing management system. Draw a UML sequence diagram using the class diagram and the following scenario. The specifications below describes the interaction among system objects when performing one of the use cases. In this use case, a user rents a room to a student.

Preconditions of this use case are that the user already logged-in and several objects already exists in the system. Some of these objects are: GUI (an object representing the graphical user interface of the system), user (instance of User), inventory (instance of BuildingInventory), building (instance of Building) and room (instance of SingleRoom).

[18 Marks]

1. The user **select**s **institution and room type** from the GUI.
2. GUI gets the building to the given institution by calling the user object and the user object gets the closest building from the inventory.
3. GUI then asks for a list of available rooms from the building.
4. Building checks all single rooms for their availability.
5. Building returns list of rooms to the GUI.
6. If the list is empty GUI **displays no available rooms**.
7. Otherwise GUI **displays available rooms**.
8. Then the user selects a room (if rooms are available).
9. GUI calls the user object to rent the selected room.
10. User object calls the inventory to create a single contract.
11. Inventory creates the contract (instance of SingleContract).

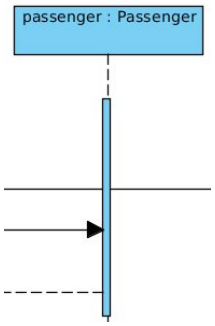
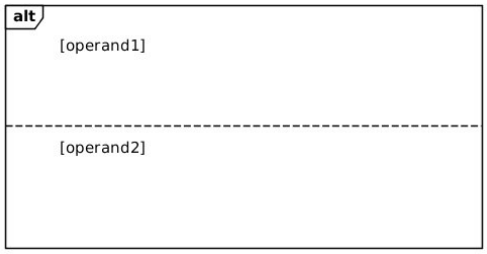
HINTS:

- All input parameters are omitted from class methods to simplify the diagram. No need to add any parameters.
- Beware that each building has many single rooms.
- This scenario doesn't include the steps where student and room associations to contract are set.

Please use the names below for the actor name, guards and messages in your diagram.
GUI, user, inventory, building, room, select institution and room type, display no available rooms, display available rooms, select room, for all single rooms

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

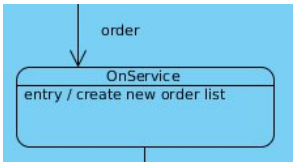
Draw a UML state machine diagram using the following list of specifications. The specifications partially describe the states of a room in a student housing management system.

[9 Marks]

1. When a new room is added to the system it's in good condition and available.
2. A new ocuppant can be added to available rooms.
3. An existing occupant can be removed from available rooms.
4. A room becomes full when number of occupants is equal to the number of maximum occupants (room capacity).
5. New occupants cannot be added to full rooms.
6. When an occupant is removed from a full room it becomes available.
7. Rooms in good condition are either available or full.
8. When an available room needs repairs, a user marks it for maintenance and the room becomes in maintenance.
9. Available rooms can be marked for maintenance if the number of ocuppants is zero.
10. Rooms in maintenance are initially under repair.
11. When the repairs are finished the room under repair goes through an inspection.
12. If the room passes the inspection it becomes available again.
13. If the room fails the inspection it goes through repairs again.
14. A rooms is considered in maintenance when it's under repair or under inspection.
15. Available rooms can be removed from the system if the number of occupants is zero.
16. Rooms in maintenance can be removed from the system.

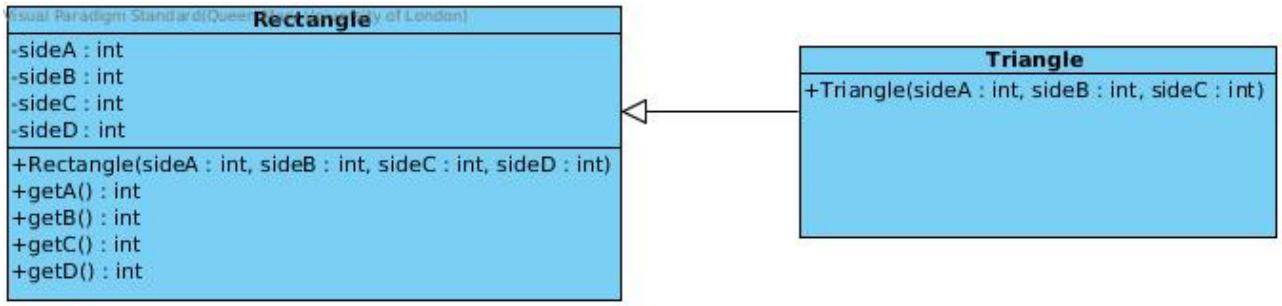
Please use the names below for the state names and transitions in your diagram. You can use them more than once.
InGoodCondition, Available, Full, InMaintenance, UnderRepair, UnderInspection, removeOcuppant, addOcuppant, numberOfOcuppants, maxOcuppants, markForMaintenance, finishRepairs, passedInspection, failedInspection, removeRoom

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

a) Briefly explain if the class hierarchy below goes against any design principles and suggest a better hierarchy (explain why it's better). You can use UML to depict your suggested hierarchy. **[8 Marks]**



b) Suppose you are asked to test the following Java method and you decided to test all the paths in the code. **[10 Marks]**

```
public int verifyMessageLength(char[] content){
    int length = content.length;
      if(length > 10)
        return 0;
      if(length > 1)
        return 1;
      if(length == 0)
        return -1;
  }
```

i) Briefly explain what a path is and how path coverage of a test suite is measured in software testing.

ii) List all paths in this code including the infeasible ones.

iii) Will a test suite that executes of all of the feasible paths reveal any defects?

c) Outline in one paragraph the information you would need to gather in order to perform domain analysis for an hotel reservation system. **[8 Marks]**

d) Suggest an efficient design pattern for each of the design issues below and briefly explain how would you implement the suggested design pattern. **[9 Marks]**

i) Your design includes an object manager and you want to make sure only a single copy of this manager class is created.

ii) Your design includes many objects that need to be updated as soon as the state of another object in the system is changed.

iii) You want to incorporate an existing class from another hierarchy to your hierarchy. However, you want don't want to use the existing method names since they don't follow your hierarchy's naming style. You want achieve this without modifying the existing class as it might require a lot of effort.

e) Discuss the main differences between iterative and non-iterative software development. Briefly explain advantages and disadvantages of iterative development.

[9 Marks]

f) Suppose you need to estimate the cost and effort of the two software development project scenarios below. Propose an estimation method for each scenario and briefly explain how the method is performed.

[6 Marks]

i) You have team members who worked on a project similar to your current project and they can estimate the size of the project.

ii) You have documentation from similar completed projects and project members who has experience in cost estimation.

End of Paper