NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

SEMESTER 2 (2007/2008)

EXAMINATION FOR

CS2103 – SOFTWARE ENGINEERING

April 2008 **Time Allowed: 2 Hours**

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SEVEN (7)** questions and comprises **FOURTEEN (14)** printed pages, including this page.
2. Answer **ALL** questions within the space in this booklet
3. This is an Open Book examination.
4. Please write your Matriculation Number below.

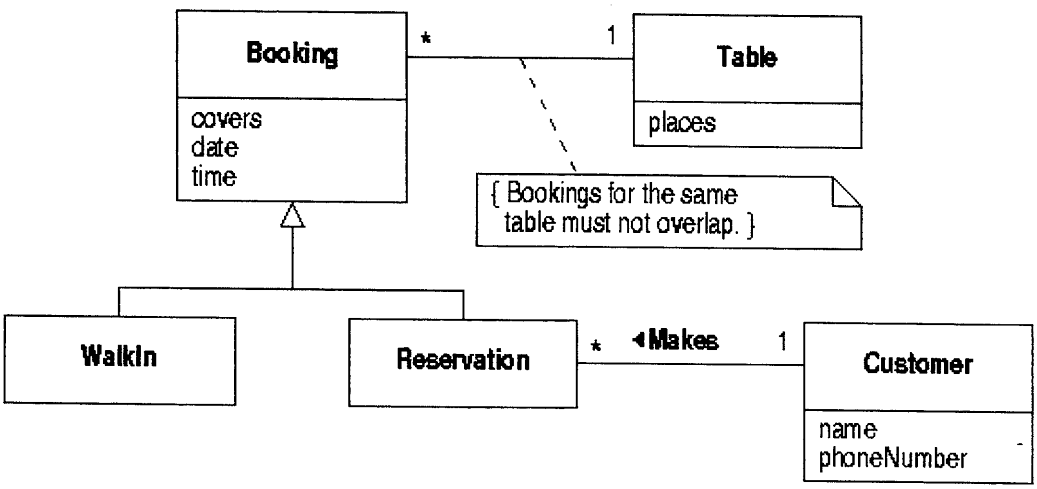
MATRICULATION NO: ______________________

This portion is for examiner's use only

Question	Marks	Remarks
Q1 (10)		
Q2 (5)		
Q3 (5)		
Q4 (8)		
Q5 (10)		
Q6 (5)		
Q7 (12)		
Total (max 55)		

Question 1 **[10 marks]**

Given is a domain model for the Table Booking System example requirements discussed in the lectures of the module CS2103.



Above model allows customer group of any size, represented as covers in Booking, to be allocated any table. Given model also assumes that all bookings are of standard length, and only arrival time is recorded. Provide changes in the given model, by re-drawing only the relevant sections of the model, to reflect each of the following new requirements:

(a) Restaurant decides to provide designated smoking and non-smoking tables.

(b) Restaurant decides to allow flexibility in the lengths of the bookings i.e. diners might be asked to make bookings only for a specified duration.

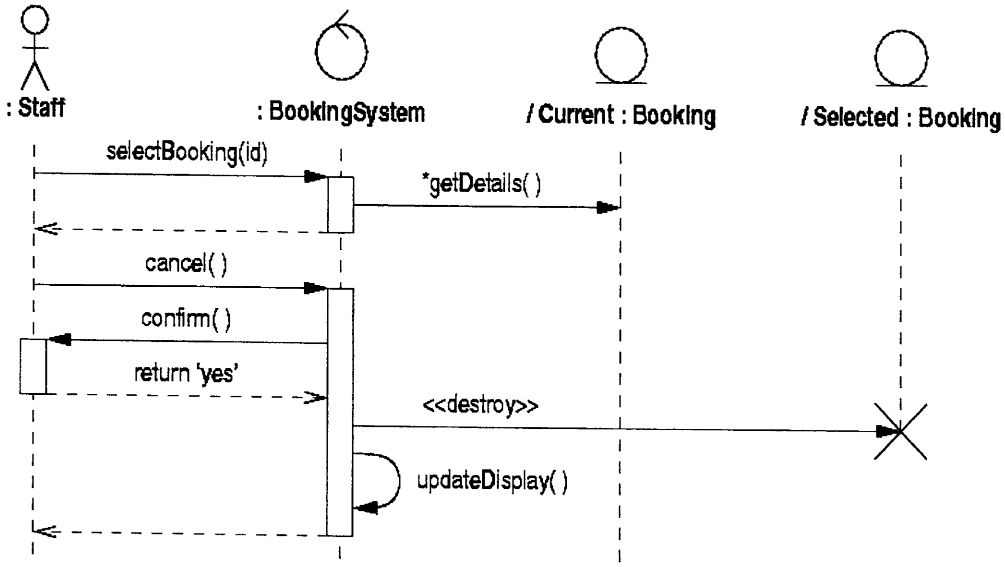
(c) Restaurant allows bookings to be made for more than one table simultaneously.

(d) Restaurant decides to formalize the number of additional places that can be squeezed onto a table as its overflow capacity. Provide changes to reflect the overflow capacity of the table. Show the constraint that covers will not be allowed to be more than table size (i.e. places + overflow capacity).

(A blank space for writing your answers)

Question 2 **[5 marks]**

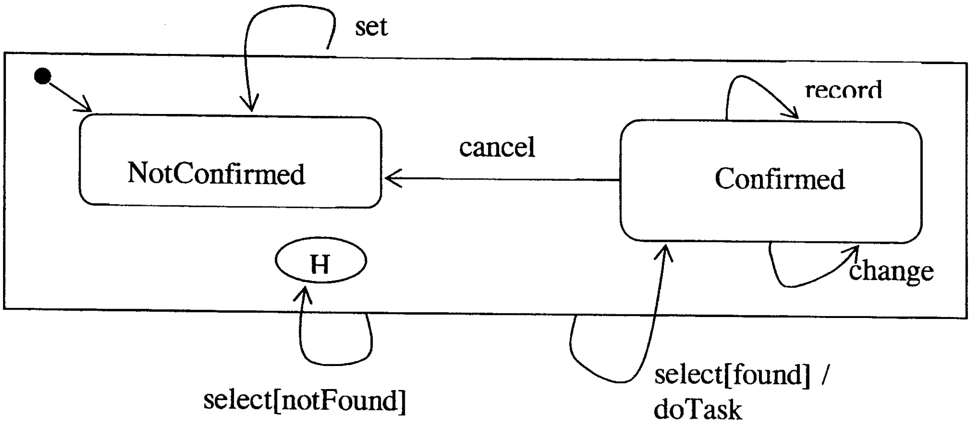
Following figure shows an interaction between user and the booking system to get confirmation of a cancellation of a booking in table booking system. Is the interaction consistent with the layered architecture ? If not, explain why not, and produce a revised sequence diagram showing how confirmation of a cancellation can be obtained.



Question 3

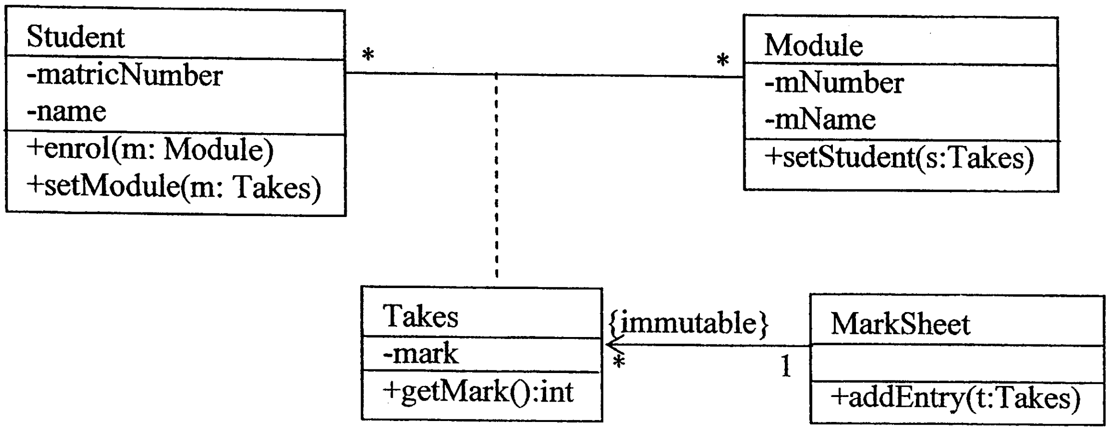
[5 marks]

Following state-chart is drawn to model states of a TimeSlot class. Draw a state-chart, without the use of composite states, that is equivalent to the following state-chart.

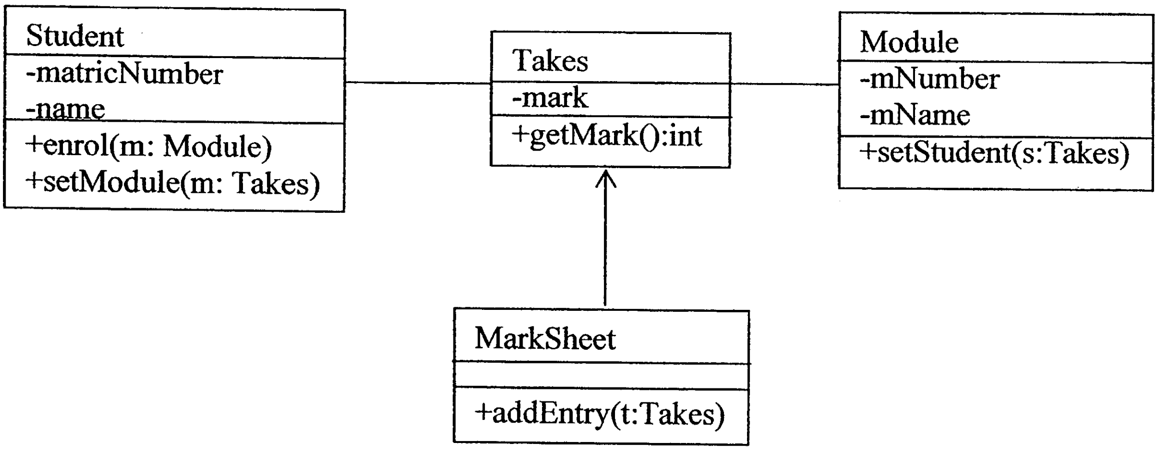


Question 4

Consider the Class Diagram below. Note that a student can take the same module many times.



a) The Class *Takes* is an association class, and stores the attribute value representing the mark a student has in a module. A common strategy is to link the association class into a simple class linked to the two original classes, as shown below. On this diagram, show the new multiplicities and mutabilities of the associations. **[2 marks]**

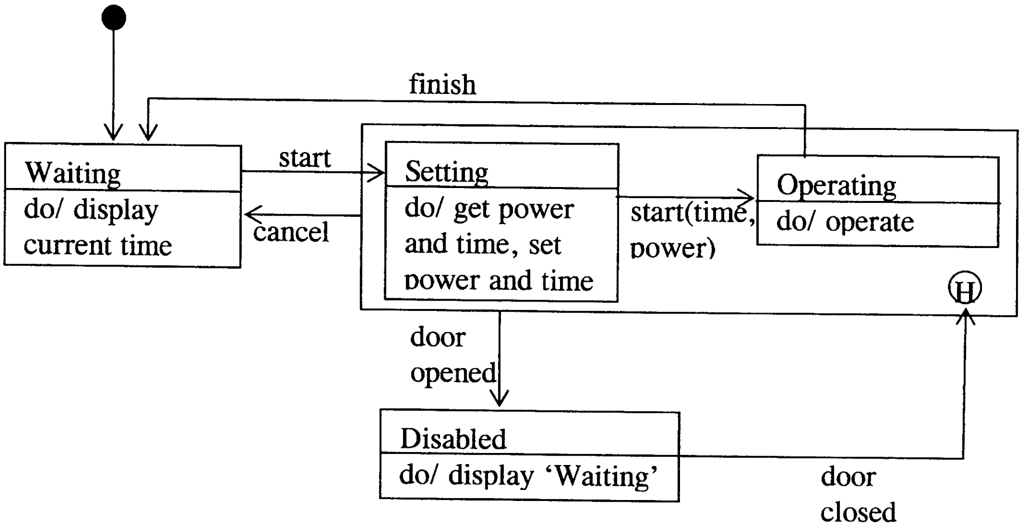


b) Write Java classes that capture the mutability, multiplicities, referential integrity and navigability of the *Takes* class as shown in the diagram above. Assume that referential integrity between a *student* and *module* is enforced in the *Takes* class. **[4 marks]**

c) Write Java classes that capture the mutability, multiplicities, referential integrity and navigability of the *MarkSheet* class as shown in the diagram above. **[2 marks]**

Question 5

Consider the State Chart below, and the Java code that lists the variables declared in the class *Microwave*, a constructor and helper methods.



A statechart for a microwave

```
public class Microwave
{
        private int power;
        private int time;

        private final int Waiting = 0;
        private final int Setting = 1;
        private final int Operating = 2;
        private final int Disabled = 3;

        private int state;

        private int historyState;

        public Microwave() {
          state = setWaitingState();
        }

        public int setWaitingState() {
                System.out.println{//SystemTime};
                return(Waiting);
        }

        public int setSettingState() {
                System.out.println{//get power and time};
                // set power and time
                return(Setting);
        }

        public int setOperatingState() {
                // operate oven at power for time
                return(Operating);
        }

        public int setDisabledState() {
                historyState = state;
                System.out.println{"Waiting"};
                return(Disabled);
        }

        // the code for methods goes here
}
```

a) Write Java methods for the *start* transition. **[4 marks]**

b) Write Java methods for the *cancel* transition. **[3 marks]**

c) Write Java methods for the *door closed* transition. **[3 marks]**

Question 6 **[5 marks]**

Suppose a system has been built to implement the functionality described in the State Chart above, and test cases are being defined to test the system. For each of the tests below complete the entry (including the expected output and the objective of the test).
Assume that power must be either 300 (for half power) or 600 (for full power), and an error message is displayed if any other value is entered.

id	precondition	input	output	objective
1	Power off	Power up microwave		
2	Displaying prompt for time and power	Enter 10 for time and 600 for power		
3	Displaying prompt for time and power	Enter 10 for time and 601 for power		
4	Oven Operating	Open door		
5	Previous state was Setting and current state is Disabled	Close door		
6	Displaying prompt for time and power	Cancel		

Question 7

Consider the State Chart above. The intent of the State pattern is to "allow an object to alter its behaviour when it's internal state changes. The object will appear to change its class.". The following questions are about how the State Chart will be coded in Java when the State pattern is applied.

a) List the names of the classes, their variables, any inheritance, whether they are abstract classes or interfaces. **[4 marks]**

Class name	Variable name and type	Inherits from	Abstract/Interface

b)Write the Java code for the *door opened* transition from the composite state that includes states Setting and Operating. **[3 marks]**

c) Write the Java code for the *door closed* transition from state Disabled. In which class is the method defined? **[3 marks]**

d) Write the Java statement that calls the method defined in part (c) above. Which class will call this method? **[2 marks]**

End of Paper