CS3004 Software Design and Analysis (Subjective Solution)

Saturday, December 24, 2022

Course Instructors

Dr. M. Bilal, Ms. Pariwish Touseef

Serial No:

Final Term Exam

Total Time: 2 Hours and 20 Minutes

Total Marks: 70

_____

Signature of Invigilator

Roll No	Section	Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Verify at the start of the exam that you have a total of eight (8) questions printed on thirteen (13) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as 'useful in the paper' or else there may a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
6. Use only your own stationery and calculator. If you do not have your own calculator, use manual calculations.
7. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

	Obj.	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Q-7	Q-8	Total
Total Marks	20	4	4	4	10	5	6	5	12	70
Marks Obtained										

Vetted By: _____ Vetter Signature: _____

University Answer Sheet Required: No ☐ Yes ☐

Question No.1 (4 Marks)

What are the 5 views that 4+ 1 view model provides? For all of them, name one diagram type that provides that view.

View	Diagram providing the view

Solution:

View	Diagram providing the view
Process View	Sequence,State,Activity and Communication
Logical View	Class, Object and Composite Structure
Development/ Module/ Implementation View	Component and Package
Physical View	Deployment and Network
Scenario View	Usecase and User Stories

Question No.2 (4 Marks)

Discover the major difference between Package, Component and Deployment diagram.

Solution:

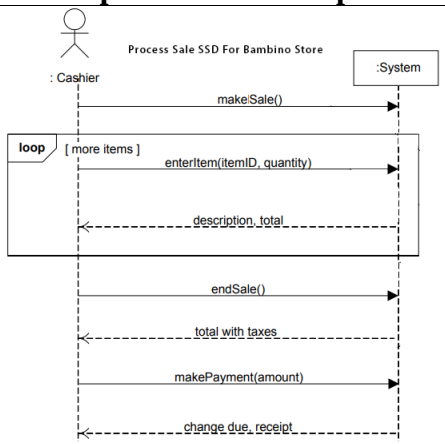
A Component diagram shows you how different elements of your system have been grouped together (into assemblies / dlls etc) - and the link between these components. Component diagram does not describe functionality of system but it describes compnents used to make those functionalities.

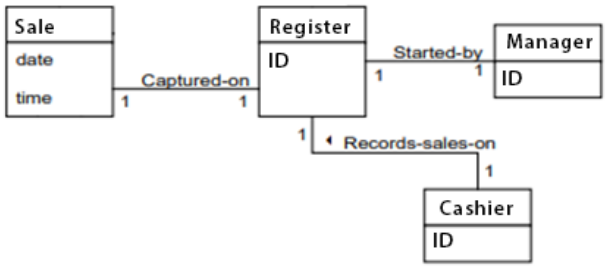
A Deployment diagram takes you one step further and describes on which hardware elements do these components reside. It **describes installation**, **configuration**, and **deployment** of the software application.

A package diagram composed of packages and dependencies between them. It enables you to organize model elements such as use cases, classes into groups.

Question No.3 (4 Marks)

Write Operational Contract for the make sale from the given system sequence diagram and partial domain model. Must show Instance creation, association and modification in post condition and also fill its pre-condition and reference block.





Solution

Operation : Cross	makeSale()
Reference	SSD: Process Sale
Pre-Condition	None
Post-Condition	- A Sale instance s was created (instance creation). - s was associated with the Register (association formed). - Attributes of s were initialized.

Question No.4 (10 Marks)

A: What are design Pattrens? What does GRASP Stands for. List down names for GRASP pattrens [3 Mark]

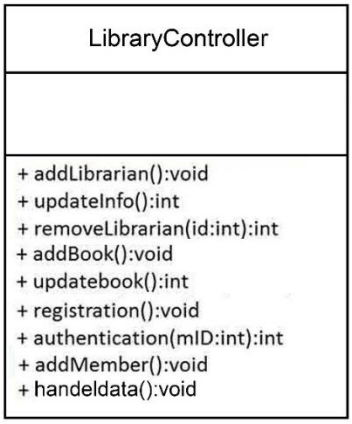
Solution

Grasp patterns describe fundamental principles of object design and responsibility. These are solutions to recurring problems. GRASP stands for General Responsibility Assignment Software Patterns.

Names of patterns are:

- Creator
  - Who creates?
- Information Expert
  - Who, in the general case, is responsible?
- Low Coupling
  - Support low dependency and increased reuse
- Controller
  - Who handles a system event?
- High Cohesion
  - How to keep complexity manageable?
- Polymorphism
  - Who, when behavior varies by type?
- Pure Fabrication
  - Who, when you are desperate, and do not want to violate High Cohesion and Low Coupling?
- Indirection
  - Who, to avoid direct coupling?
- Law of Demeter (Don't talk to strangers)
  - Who, to avoid knowing about the structure of indirect objects?

B: Consider the following example. In the design for a Library Management system, the following class has been designed: [7 Marks]



a) Do you think is this a good design? [1 Mark]

Solution

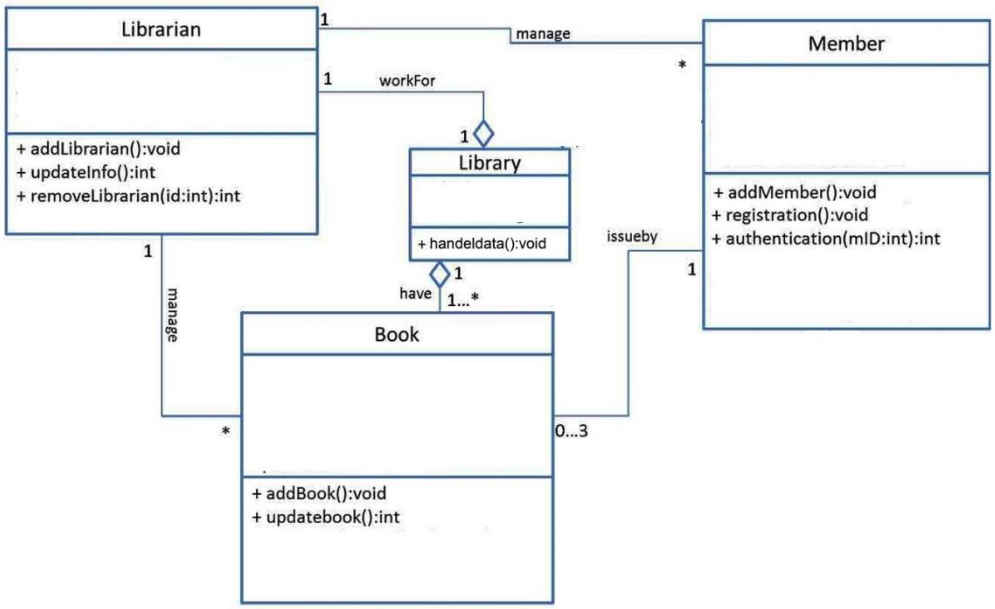
No, Its not a good design.

b) What is the issue with this class and how can we improve this scenario? [1 Mark]

Solution

The issue with this class is it is not following rules of GRASP patterns and functions that are not relatable are added in one class. Therefore, it is having low cohesion as the class os doing undesirable functionailities and suffer from problems like hard to comprehend, hard to reuse, hard to maintain and constantly effected by change. It is a poor pratice. OR Bloated controller and to solve this issue make more controller OR discuss according to pure fabrication or information expert pattern. OR discuss about coupling. OR discussion on distribution of classes. (1 Mark)
If data types etc discuss. (0.5 Mark)

c) Reconstruct with an improved version using GRASP Patterns. (This need redrawing of diagram). [5 Marks]

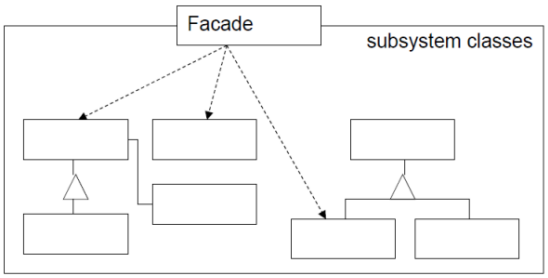


Question No.5 (5 Marks)

A. **You at FAST university developed a subsystem with 20 classes cls1-cls20. To enhance reusability, each class methods in the classes can be used in a variety of different ways. However, you know that most companion subsystems will mostly use the classes studentc1, teacherc4, and adminc8 more frequently. These three frequently used classes are accessed by other sub systems in a unidirectional way only. How can you simplify the view that the other subsystems have of your subsystem? Suggest a GOF design pattern for this scenario and justify your answer: [3 Marks]**

Solution

GOF Design Pattern Name: Façade design pattern
Reason: Façade will introduce a dedicated interface class that simplifies the view of the class collection.



B. **Write major difference between Façade and Mediator Pattern. [2 Marks]**

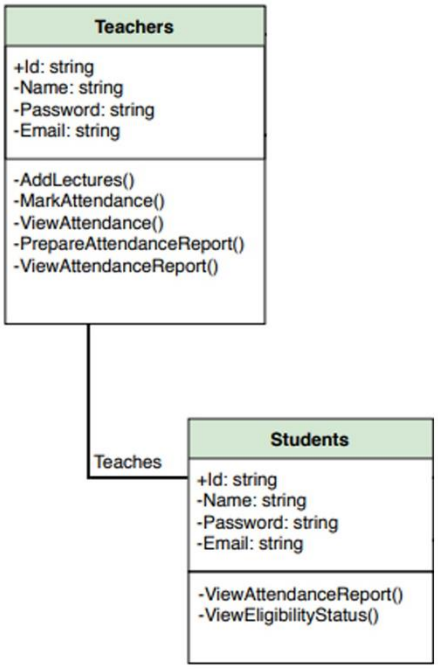
Solution

- Façade, unlike Mediator, abstracts a subsystem of objects to provide a convenient interface. Unidirectional. Façade objects make requests of the subsystem, but not vice-versa.

- Mediator enables cooperative behavior, that colleagues don't or can't provide. Multidirectional.

Question No.6 (6 Marks)

You have go through the concept of model to code. Write code for the given Teachers class only. Moreover, do write simple attributes and reference attributes for Teachers class and Highlight them.



Solution

```
Public class Teachers{
        Public String Id;
        Private String Name;
        Private String Password;
        Private String Email;
        Private Students students;
}
```

Simple Attributes

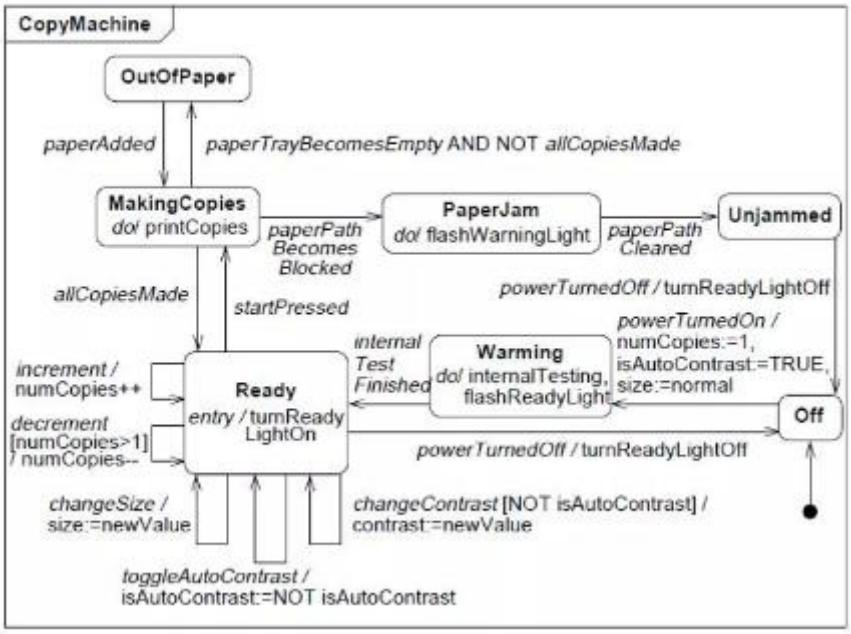
Reference Attribute

Question No.7 (5 Marks)

Identify states and events for a Photocopier machine from the description given below and draw the state diagram for the same.

Initially the machine is off. When the operator switches on the machine, it first warms up during which it performs some internal tests. Once the tests are over, machine is ready for making copies. When operator loads a page to be photocopied and press 'start' button, machine starts making copies according to the number of copies selected. While machine is making copies, machine may go out of paper. Once operator loads sufficient pages, it can start making copies again. During the photocopy process, if paper jam occurs in the machine, operator may need to clean the path by removing the jammed paper to make the machine ready.

Solution



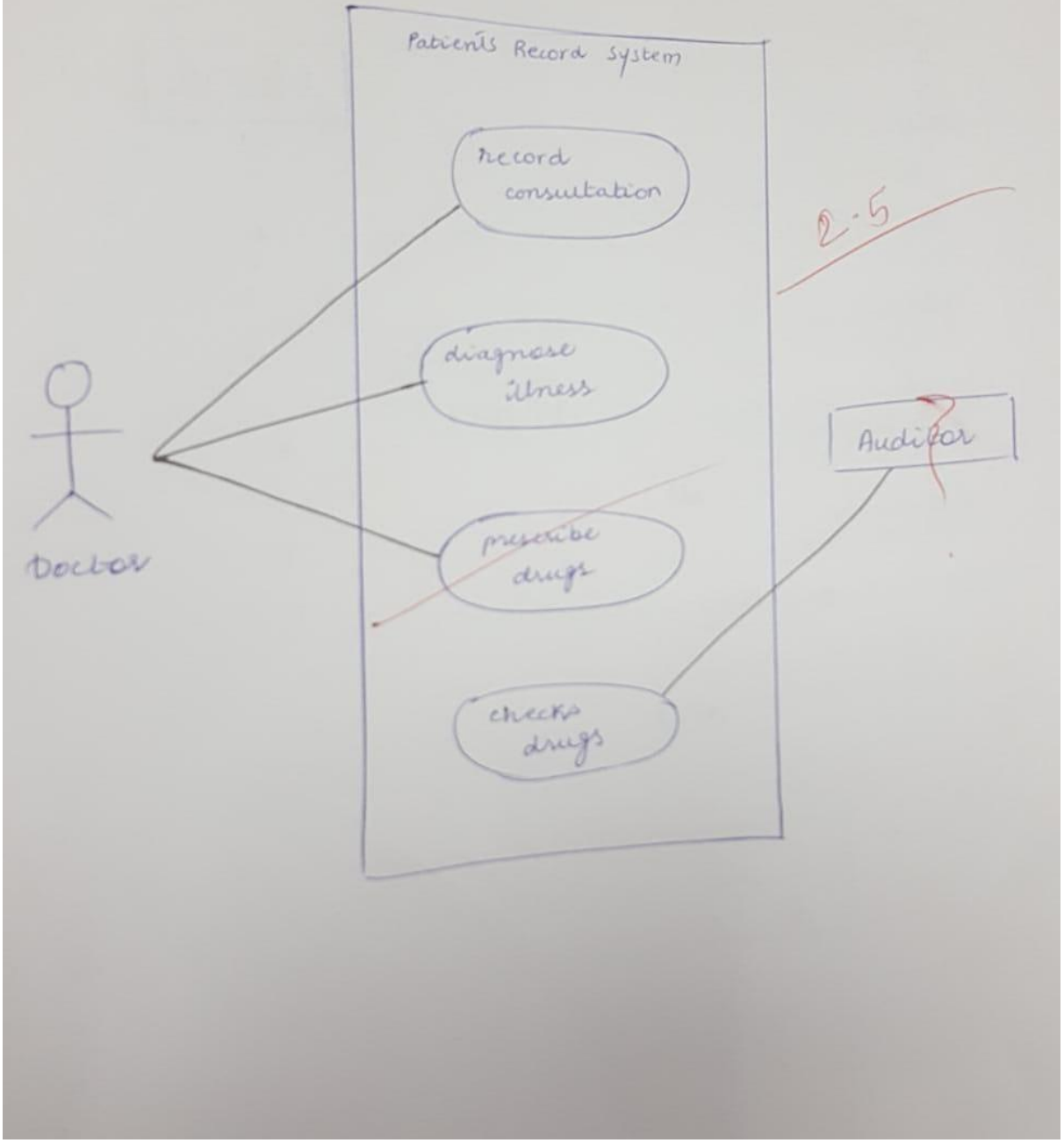
Question No.8 (12 Marks)

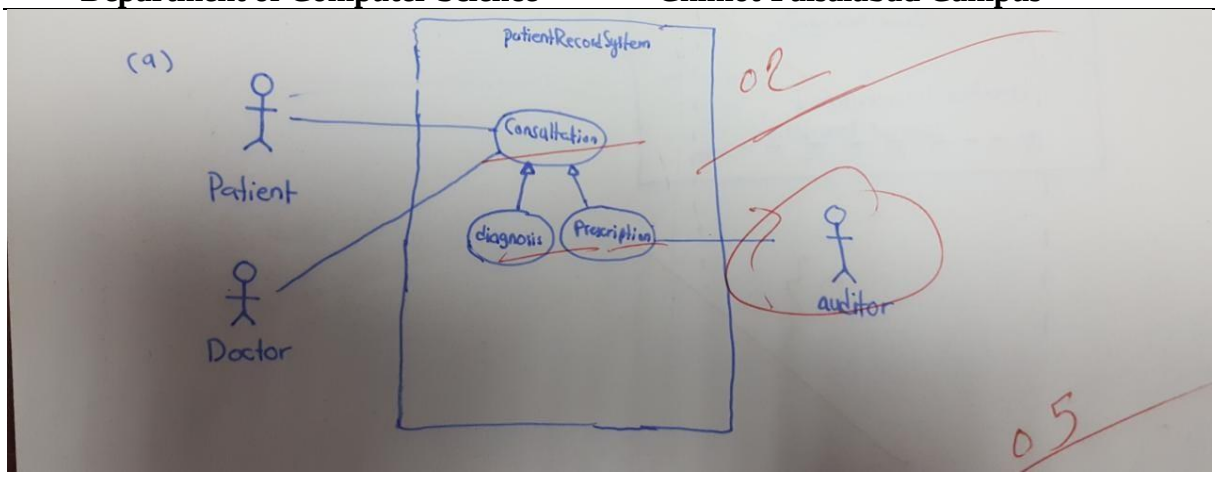
You are required to present an outline design of a system that will be used by doctors at a medical practice to keep basic patient records. The system should record each consultation between a doctor and patient, any illness diagnosed by the doctor, and any drugs prescribed to the patient. At regular intervals, an auditor will use the system to check whether the same drug is being prescribed repeatedly to a particular patient.

(a) Sketch a UML use case diagram for the above functionality. [3 marks]
(b) Sketch a UML class diagram for a system architecture to support this functionality. [6 marks]
(c) Sketch a UML interaction diagram showing the operation of the audit function. [3 marks]

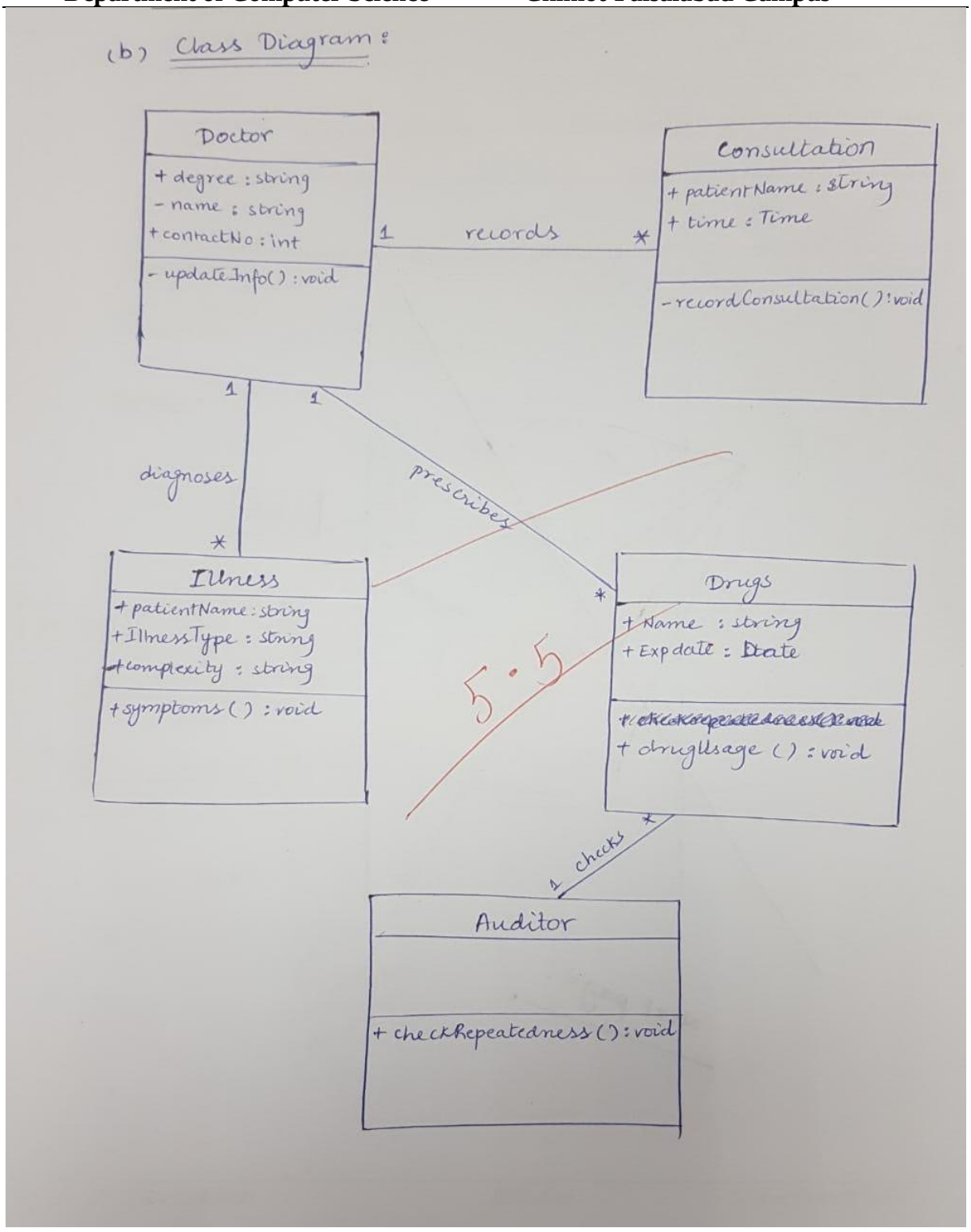
Solution

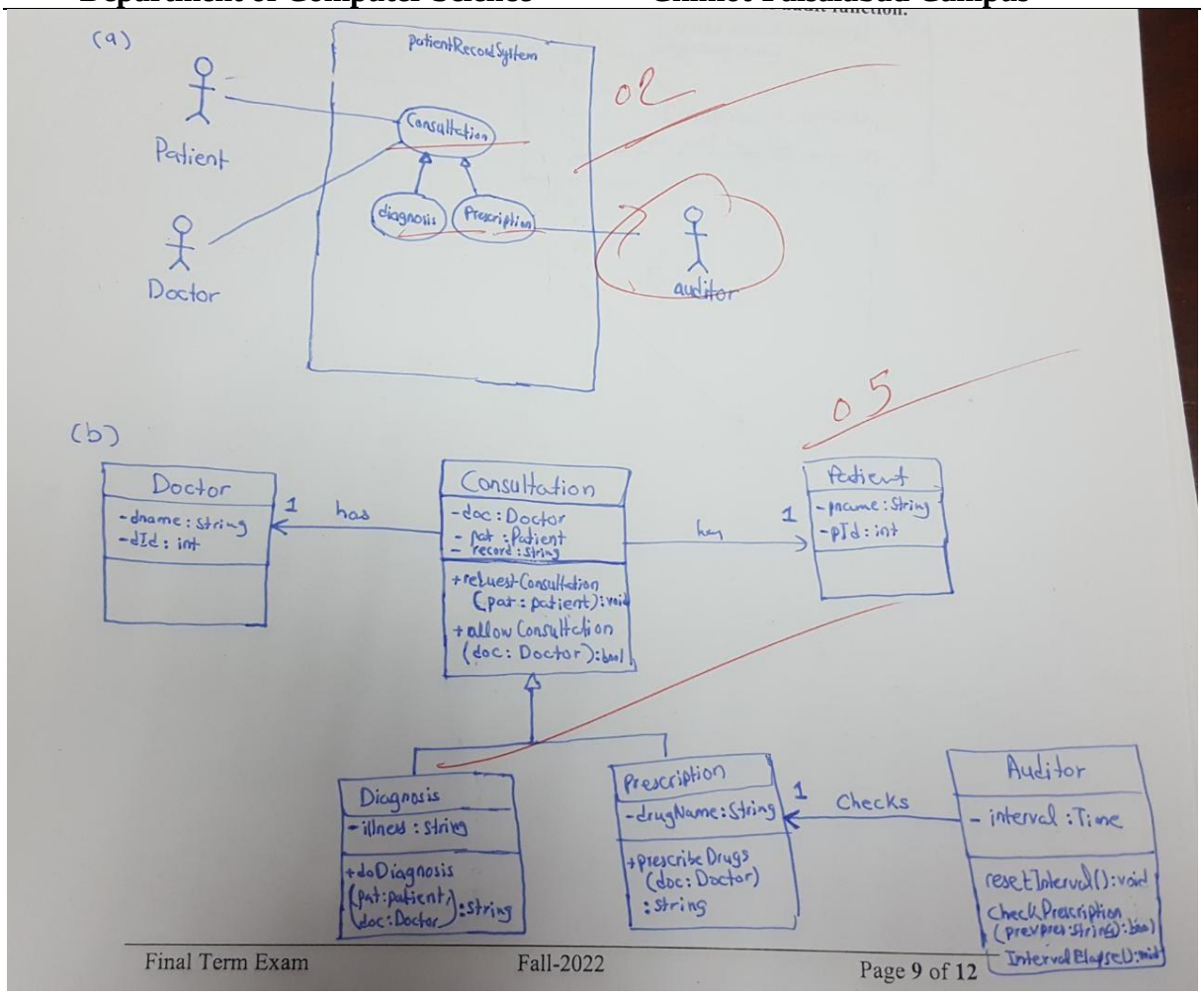
(a) Use case Diagram:

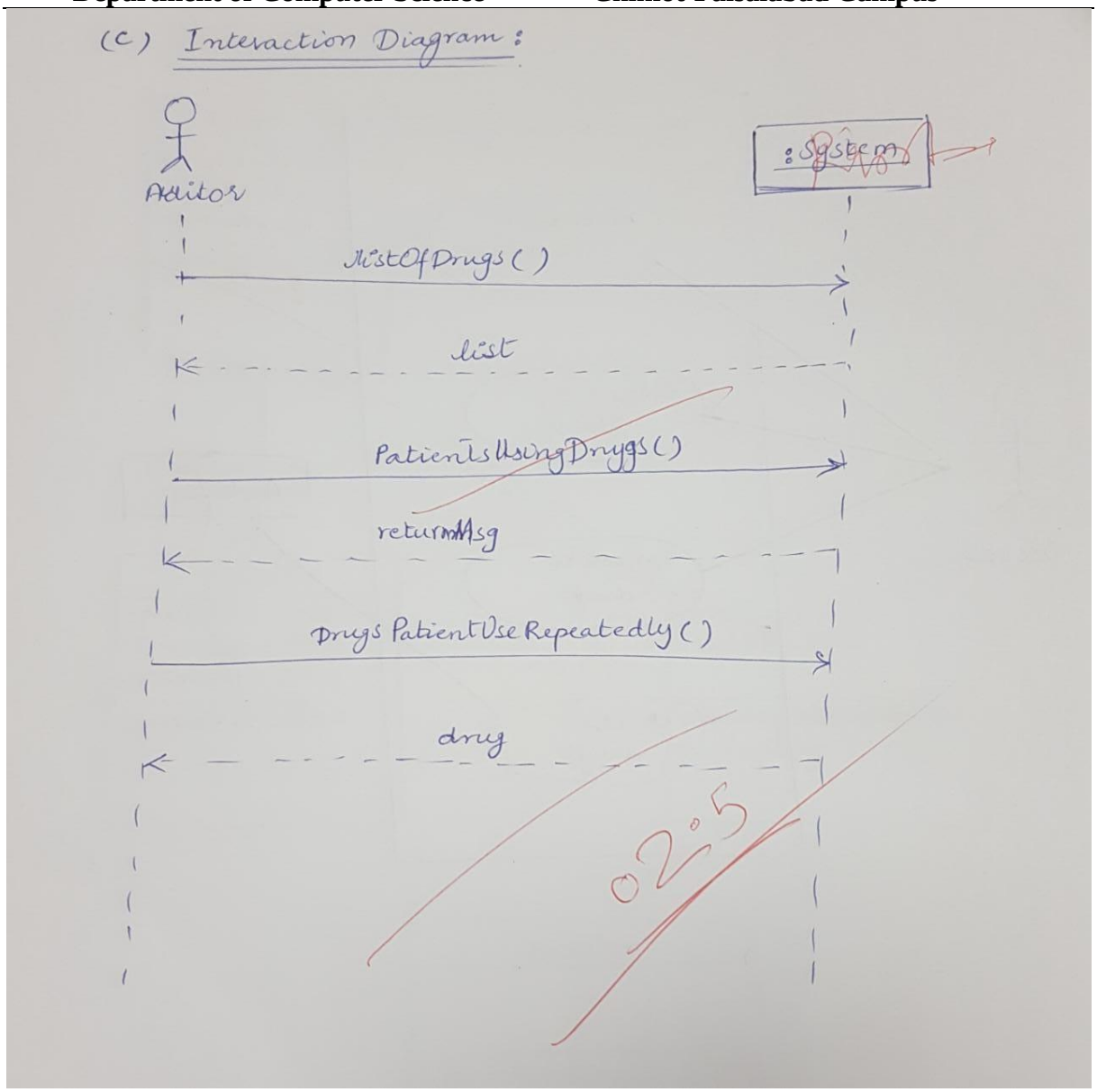


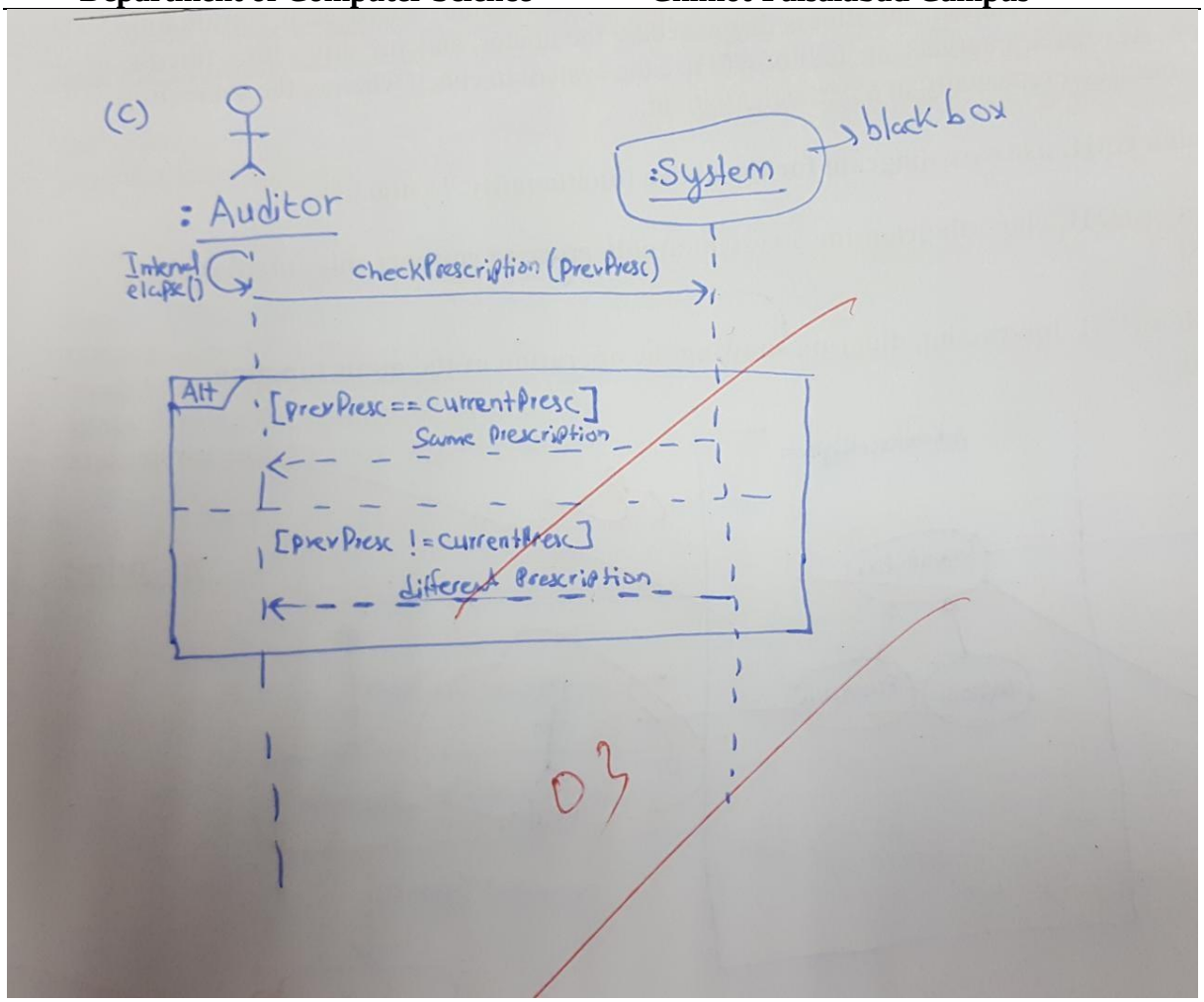

(a)
PatientRecordSystem
Patient
Doctor
Consultation
diagnosis
Prescription
auditor
02
05

(b) Class Diagram:




(a)
PatientRecordSystem
Patient
Consultation
diagnosis
Prescription
Doctor
auditor
02
(b)
05
Doctor
-dname : String
-dId : int
1
has
Consultation
-doc : Doctor
- pat : Patient
- record : String
+request Consultation (par : patient) : void
+allow Consultation (doc : Doctor) : bool
1
Patient
-pname : String
-pId : int
Diagnosis
- illness : string
+doDiagnosis (pat : patient, doc : Doctor) : string
Prescription
-drugName : String
+prescribe Drugs (doc : Doctor) : string
1
Checks
Auditor
- interval : Time
resetInterval() : void
CheckPrescription (prevPres : String[]) : bool
IntervalElapsed() : void


(c) Interaction Diagram:
Auditor
:System
ListOfDrugs()
list
PatientIsUsingDrugs()
returnMsg
DrugsPatientUseRepeatedly()
drug
02:5


(c)
: Auditor
:System
black box
Internal elapse()
checkPrescription (prevPresc)
Alt
[prevPresc == currentPresc]
Same Prescription
[prevPresc != currentPresc]
different Prescription
03