Qatar University, College of Engineering, Computer Science and Engineering Department
Course: CMPS251: Object Oriented Programming **Instructor:** Mohammed Al-Sada

Final Exam
Duration: 120 minutes, **Date:**

Name: ..... **ID:** ..... **Score:** ...../30

Instructions:

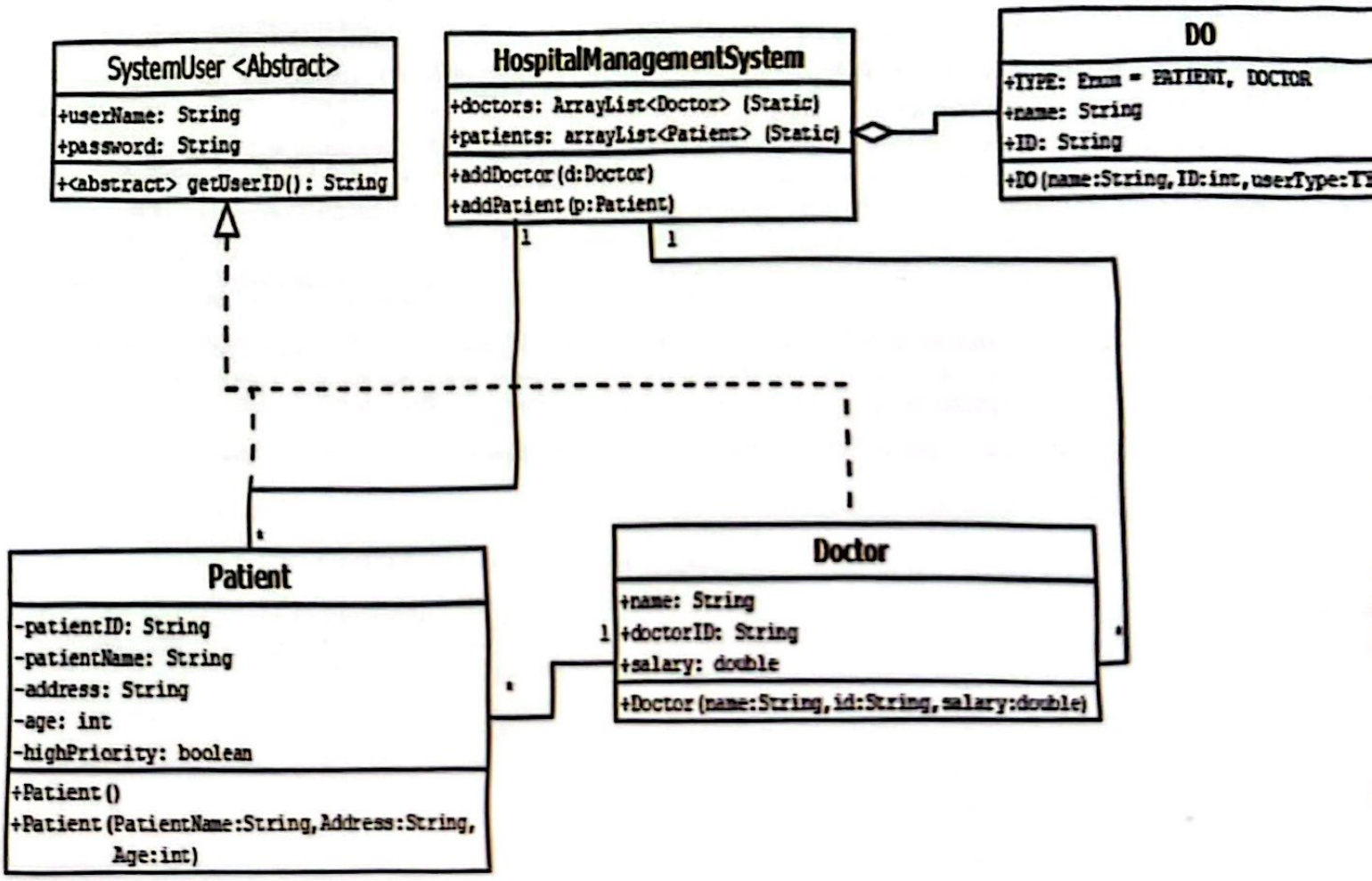
This exam consists of 6 questions (one bonus Question (Q5)). Do not answer questions separately from one another, the whole program should be written to compile and run correctly as one project

Read the below system description carefuly:

The below hospital system includes three classes:

1- Patient: which includes basic information about patients.
2- Doctor: which includes each doctor's information, and their assigned patients. Each doctor is assigne to different patients, which are stored in the arrayList called *patients*.
3- HospitalManagementSystem: for managing different aspects of doctors, patients and appointment Assume that the static arrayLists *doctors* and *patients* are already filled with all the doctors and patien in the hospital.
4- DO: which will be used as data-object for saving information about patients and doctors.

Below is the UML class diagram



Assumptions and considerations:

1- You are not allowed to rip-off the papers.
2- Your code should be written to execute on a computer. Write complete code and correct code, syn errors will be counted. Add comments if needed.
3- All setters and getters are already implemented.
4- Assume all import statements are already provided.
5- You may add extra methods or attributes to answer questions, however, use libraries or external code at your own risk.
6- You must implement correct exception handling for checked exceptions.

Q1. (9 points). Implement the **EVERYTHING** in the UML Class diagram.

Q2. (2 point). In the class *SystemUser*, create the method *boolean validate(String username_temp, String password_temp).* This method should return true if the attributes userName and password in the class SystemUser are EXACTLY the same as the ones passed as arguments of the method (username_temp and password_temp). Otherwise, the method should return false.

Q3. (8 points) In the the class *HospitalManagementSystem*, create the method called *boolean save()*, whi does the following:

I. Create an a *local* variable of type arrayList that is of type *DO*, where DO is the inner class. C the arrayList *DOs*.
II. Fill the arrayList *DOs* with objects of type DO, where each DO object corresponds to a docto to a patient stored in the arrayLists doctors and patients in the class HospitalManagementSys The Dos should be as follows:
    a. The attribute name in a *Patient* or *Doctor* object should be set to the DO's *name* attribute.
    b. The DO ID attribute should be set according the object type:
        i. If object is of type Doctor, the attribute id in DO should be the doctorID attribute of the object.
        ii. if the object is of type Patient, the attribute id in the DO should be patientID attribute o object.
    c. The DO enum attribute "type" should be set based on the object's type (to Doctor or to Patient), object creation by calling the DO constructor. Add a comment with your name for an extra hal
III. Save all the DOs as objects to a file called System*Users.DB*.
IV. The method should finally return return true if all objects have been written successfully, otherwise returns false.

Q4. (8 points) In class *SystemUser*, create the *static* method called *getID* that returns the ID of an o is passed as a parameter to the method, where the passed object can be an instance of the classes *Pa Doctor*. For example if we call the code:

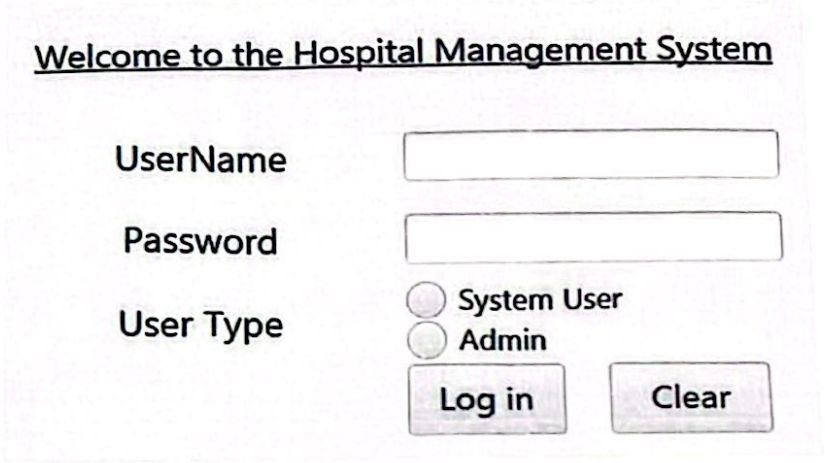
```
=Patient p1=new Patient("Salem", "Doha", 22);
 Doctor d1=new Doctor("Rashid","23419", 20400.0);
 ystemUser.getID(p1); // method will return the id of p1.
 stemUser.getID(d1); // method will return 23419
```

Bonus – Q5 (2 points) In Class *Patient*, write the code for the method to generatePatientID(), which generates a unique *patientID* for each newly created patient object. The method should do the following:

a. generate an ID number consisting of 4-5 randomly generated numbers.
b. Makes sure the generated number is **unique** for each patient object (no two patients should have the same patient ID).
c. set the *patientID* attribute to the created patient object when calling its constructor.

Reference: the method **Math.Random()** generates and returns a random number of type double between 0 and 1, for example, 0.32 – 0.43 – 0.5, which are all of type double. You can use this method to generate the ID. **You may use other random number generation methods.**

Q6. (8 points) Consider the below GUI for the log in screen for the hospital management system.



In the last page, you can find the general structure of a javaFX program, you can use it as base write your code.

The following are the Todos:

1- **(3 points)** Write the full code for the above GUI (No Scene Buider or FXML) You program be called "Interface.java". Your code has to start from main.

2- **(5 points)** Implement the functionality of "Log in" button as follows:

a. Add an object reference called HospMan of type HospitalManagementSystem some your class (set it to public).

b. If the user clicks the button, your program should read the values from the **Text fie ONLY**, which are user **name** and **password**.

c. Then, In *HospitalManagementSystem* class, there are already two static arrayLists and *doctors*. Your program should:

i. Find the patient or doctor who has the same user name as the one in the G
ii. Call the method *validate* (found in the *abstract* superclass "*SystemUser*") object.

If the method *validate* returns true, you should print the message (in command line): "log in su Otherwise, if the object with the same username is not found or vlaidate returns false: print out ailed"