Pair: Quiambao, Arianna Marie B. Date: 8/20/25 Quiambao, Aron Daniel B.

7OOP

1. **Following the OO workflow as discussed in class,** you are task to design the OO Model of the given problem (use draw.io) of the scenario below:

   Problem Statement. Tiny Hospital keeps information on **patients** and **hospital rooms**. The system assigns each patient a patient ID number. In addition, the patient's name and date of birth are recorded. Some patients are resident patients (they spend at least one night in the hospital) and others are outpatients (they are treated and released). Resident patients are assigned to a room. Each room is identified by a room number. The **Tiny hospital system** also stores the room type (private or semi-private) and room fee. Overtime, each room will have many patients who stay in it. Each resident patient will stay in only one room. The hospital system has features that can view patient information and view whether a room is occupied or not. Both patient and room entities must have features that allows adding, updating and searching of records.

   STEP1. IDENTIFY all the necessary **OBJECT** within the problem domain
   STEP 2. IDENTIFY all the properties and methods/behaviors in the problem statement
   STEP 3. Design the MODEL using a Class Diagram (You may use draw.io to represent the Blueprint of all the class that you need to create)
   STEP 4. Implement the **class using Java code** construct of each interacting entities that you have identified.

STEP 1: IDENTIFY OBJECTS

- OBJECT: PATIENTS, HOSPITAL ROOMS, HOSPITAL SYSTEM

STEP 2: IDENTIFY PROPERTIES AND METHODS/BEHAVIORS

- PROPERTIES:

PATIENTS HOSPITAL ROOMS HOSPITAL SYSTEM

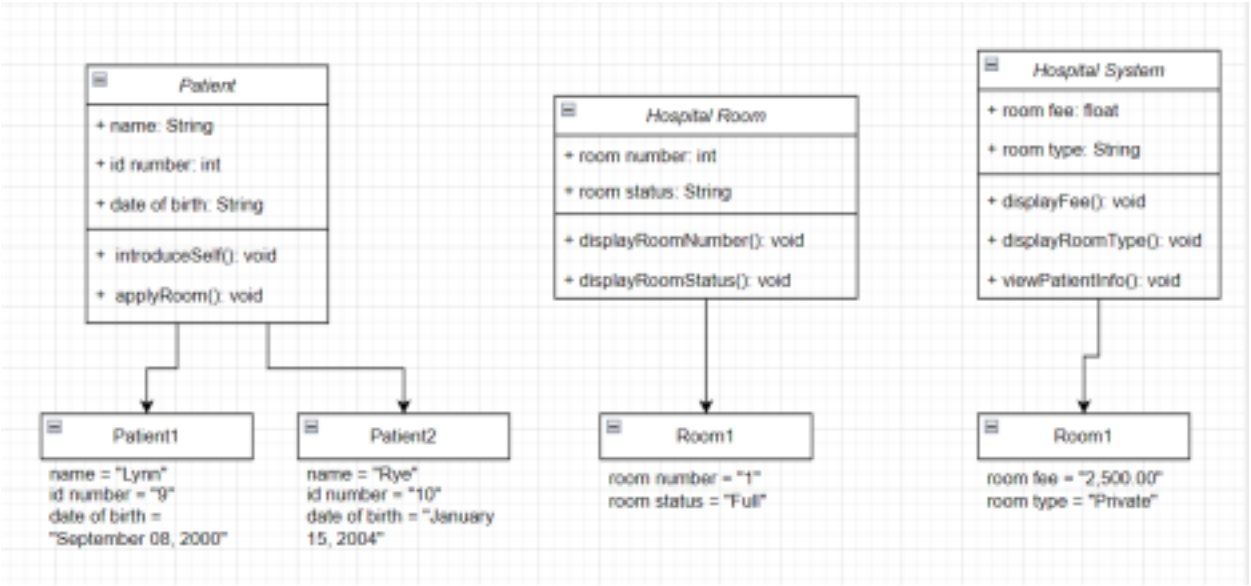
= NAME = ROOM NUMBER = ROOM FEE
= ID NUMBER = ROOM STATUS = ROOM TYPE
= DATE OF BIRTH

- BEHAVIORS:

PATIENTS HOSPITAL ROOMS HOSPITAL SYSTEM

= introduceSelf() = displayRoomNumber() = displayFee()
= applyRoom() = displayRoomStatus() = displayRoomType() = viewPatientInfo()

STEP 3:



STEP 4:

```
public class Patient
{
    //instance variables
    String name;
    int idnum;
    String date;

    //behaviors - methods

    public void introduceSelf()
    {
        System.out.println("Hi! I am "+ name + ". I was born in "+ date +". I feel very sick");
    }

    public void applyRoom()
    {
        System.out.println("I would like to apply a room with my Given Patient ID No. "+ idnum +" please!");
    }

}

public class HospitalRooms
{
    //instance variables
    int roomnum;
    String roomstatus;

    //behaviors - methods

    public void displayRoomNumber()
    {
        System.out.println("The Room is"+ roomnum +".");
    }

    public void displayRoomStatus()
    {
        System.out.println("The Room is"+ roomstatus +".");
    }
}
```

```java
public class HospitalSystem
{
    //instance variables
    float roomfee;
    String roomtype;

    //behaviors = methods

    public void displayRoomFee()
    {
        System.out.println("The Room Fee is ₱"+ roomfee+ " .");
    }

    public void displayRoomType()
    {
        System.out.println("The Room is"+ roomtype + " .");
    }

    public void viewPatientInfo()
    {
        System.out.println("The Information is Empty");
    }
}
```

```
public class TinyHospital
{
    public static void main(String[] args) {

        Patient patient1 = new Patient();

        patient1.name = "Lynn";
        patient1.idnum = 9;
        patient1.date = "September 8, 2000";

        patient1.introduceSelf();
        patient1.applyRoom();

        Patient patient2 = new Patient();

        patient2.name = "Rye";
        patient2.idnum = 10;
        patient2.date = "January 15, 2004";

        patient2.introduceSelf();
        patient2.applyRoom();

        HospitalRooms room1 = new HospitalRooms();

        room1.roomnum = 1;
        room1.roomstatus = "Full";

        room1.displayRoomNumber();
        room1.displayRoomStatus();

        HospitalSystem Room1 = new HospitalSystem();

        Room1.roomfee = 2500;
        Room1.roomtype = "Private";

        Room1.displayRoomFee();
        Room1.displayRoomType();
        Room1.viewPatientInfo();

    }
}
```