Aquino, Aaron Jan O.

Manaois, Ivan Bryan R.

BSCS 204

STEP1. IDENTIFY all the necessary OBJECT within the problem domain Object: Patient, Room, Tiny Hospital

STEP 2. IDENTIFY all the properties and methods/behaviors in the problem statement

Patient

Properties:

Name:
BirthDate:
Patient status
Patient Room

Behavior

updatePatient()
updatePatien()
patientStatus()
patientRoom()

Room

Properties:

Room Number:
Room Type:
Room Fee:
Room Status:

Behavior

updateRoom()
searchRoom()
viewRoomStatus()

Tiny Hospital

Properties:

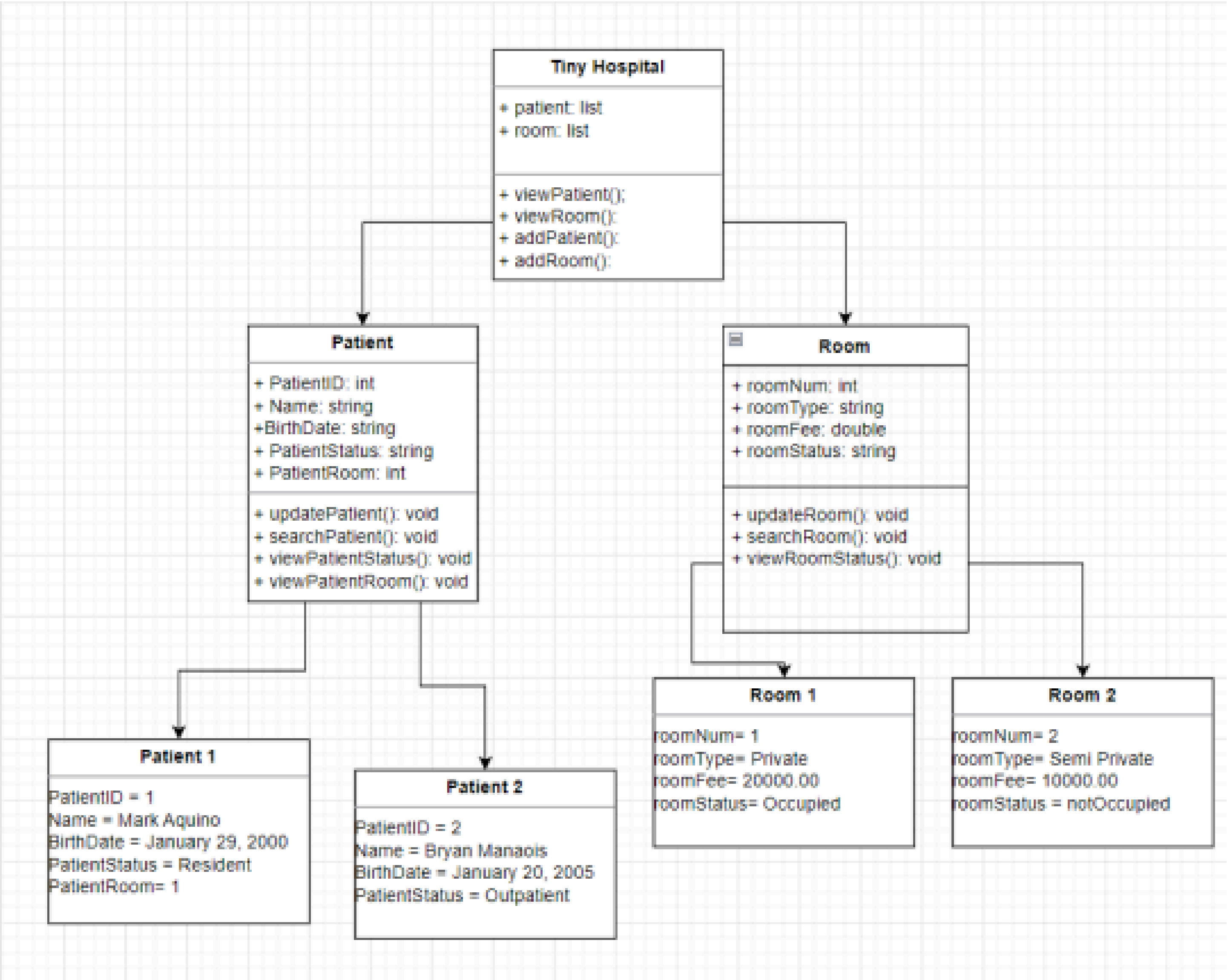
Patients
Rooms

Behavior:

viewPatien()
viewRoom()

addRoom()
addPatient()

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.

```
class Patient {
int patientID;
String name;
String birthDate;
String patientStatus; // Resident or Outpatient
```

```
 int patientRoom; // Room number
public Patient(int patientID, String name, String birthDate, String patientStatus, int patientRoom)
{ this.patientID = patientID;
this.name = name;
this.birthDate = birthDate;
this.patientStatus = patientStatus;
this.patientRoom = patientRoom;
}

public void updatePatient(String newName, String newBirthDate, String newStatus, int newRoom)
{ this.name = newName;
this.birthDate = newBirthDate;
this.patientStatus = newStatus;
this.patientRoom = newRoom;
}

public void viewPatient() {
System.out.println("PatientID: " + patientID);
System.out.println("Name: " + name);
System.out.println("BirthDate: " + birthDate);
System.out.println("Status: " + patientStatus);
System.out.println("Room: " + (patientStatus.equals("Resident") ? patientRoom :
"N/A")); System.out.println("--------------------------------");
}
}

class Room {
 int roomNum;
String roomType;
double roomFee;
String roomStatus; // "Occupied" or "NotOccupied"

public Room(int roomNum, String roomType, double roomFee, String roomStatus)
{ this.roomNum = roomNum;
this.roomType = roomType;
this.roomFee = roomFee;
this.roomStatus = roomStatus;
}

public void updateRoom(String newType, double newFee, String newStatus)
{ this.roomType = newType;
this.roomFee = newFee;
this.roomStatus = newStatus;
}



public void viewRoom() {
System.out.println("Room Number: " +
roomNum); System.out.println("Type: " +
roomType); System.out.println("Fee: " +
roomFee); System.out.println("Status: " +
roomStatus);
System.out.println("--------------------------------"); }
}

class TinyHospital {
List<Patient> patients = new
ArrayList<>(); List<Room> rooms = new
ArrayList<>();

public void addPatient(Patient p) {
patients.add(p);
}

public void addRoom(Room r) {
rooms.add(r);
}

public void viewPatients() {
for (Patient p : patients) {
p.viewPatient();
}
}

public void viewRooms() {
for (Room r : rooms) {
r.viewRoom();
}
}

public Patient searchPatient(int id) {
for (Patient p : patients) {
if (p.patientID == id) return p;
}
return null;
}

public Room searchRoom(int roomNum)
{ for (Room r : rooms) {
if (r.roomNum == roomNum) return r;
}
return null;
}
}
```