Mini Project : Command Line Appointment Booking System

Project Specification:

In this project, you have to build a **command line based appointment booking system for patients**. This project should allow patients to schedule their appointments.

Classes Description:

The following are the list of classes, you have to implement -

Class Name	Purpose
Doctor	Class having properties of Doctors
Hospital	Class having properties of hospital
Patient	Class having properties of Patient

Class: Doctor

This class should have parameterised constructor to initialise doctor_name (string), specialisations (list), availability_days (list), appointment_seat (dict where key is the day (Eg: Monday, value is the total appointments available), hospital (class Hospital) —> For initialising hospital object inside Doctor class constructor, refer this sample below -

```
class A:
    def fun(self):
        print("Hi")
class B:
    def __init__(self, A):
        self.A = A

a = A()
b = B(a)
b.A.fun()
```

This Doctor class should have following methods -

Method Name	Return Type	Description
is_available(day)	True if available , False if not available	Eg: If availability_days = [‘Monday’], appointment_seat = {‘Monday’ : 5}, then **is_avaialble(‘Monday’) -> True** If If availability_days = [‘Monday’], appointment_seat = {‘Monday’ : 0}, then **is_avaialble(‘Monday’) -> False** If If availability_days = [‘Monday’], appointment_seat = {‘Monday’ : 3}, then **is_avaialble(‘Friday’) -> False**

is_specialist(disease)	True if the doctor is specialist , else False	Eg: If specialisations = [“Cancer”, “Diabetics”] , if is_specialist(“Cancer”) -> True is_specialist(“ENT”) -> False
book_appointment(day, disease)	-1 0 1	Call is_specialist(disease) if it return False, return -1. Call the is_available(day), if it returns False, then return 0, else return 1. **Make sure to update counts of availability accordingly**
do_booking(day, disease)	Appointment Details on successful booking, error message on booking failure	Call book_appointment(day, disease), if return -1, return “Requested Doctor is not a specialist for your request”, If it returns 0, return “Doctor not available on your requested date”, else return “Appointment Successful, followed by print the details of - 1. Doctor Name 2. Hospital Name 3. Address 4. Patient Name 5. Booking Day 6. Booked for (Disease name)

Class Hospital:

This class should have parameterised constructor with hospital_name, address of the hospital.

Class Patient:

This class should have parameterised constructor with name_of_patient (string) , disease he has (string), assigned_doctor (Doctor object)

book(requested_day)	Call do_booking of Doctor class and print the return value	-

Sample Test Case:

hospital = Hospital(“ABC Multi-speciality Hospital”, “71, South Street, Ambattur, Chennai”)

doctor = Doctor(“Dr. John”, [“Diabetics”, “ENT”], [“Monday”, “Friday”], {“Monday” : 5, “Friday” : 1}, hospital)

patient = Patient(“talentpY”, “Diabetics”, doctor)

SOME SAMPLE SCENARIO’s:
patient.book(“Friday”) => SHOULD BOOK (Friday 1 appointment available)
patient.book(“Friday”) => SHOULD NOT BOOK (Friday 0 appointment available)
patient.book(“Tuesday”) => SHOULD NOT BOOK (Doctor not available on Tuesday)

patient = Patient(“Roy”, “Dengue”, doctor)
patient.book(“Monday”) => SHOULD NOT BOOK (Doctor is not Dengue Specialist)