

# Machine Coding Round: FLIPMED

## Description:

We are required to build an app that lets patients connect to doctors and book appointments. The day is divided into time slots of 60 mins each, starting from 9 am to 9 pm. Doctors can login to the portal and declare their availability for the given day in terms of slots. Patients can book appointments/ cancel existing appointments. **For simplicity you can assume that the doctors' availability is declared for that particular day only.**

## Features:

- A new doctor should be able to register, and mention his/her speciality among (Cardiologist, Dermatologist, Orthopedic, General Physician)
- A doctor should be able to declare his/her availability in each slot for the day. For example, the slots will be of 60 mins like 9am-10am, 11am-12pm..
- Patients should be able to register. Patients should be able to search available slots based on speciality. Any order of slots is okay for viewing to patients.
- Patients should be able to book appointments with a doctor for an available slot. A patient can book multiple appointments in a day. A patient cannot book two appointments with two different doctors in the same time slot.
- A patient/doctor should be able to view his/her booked appointments for the day.

## Bonus requirements:

- Patients can also cancel an appointment, in which case that slot becomes available for someone else to book.
- Different ranking strategy - use rating of doctor as a strategy to display a list of available doctors for a given specialization.
- The slots should be displayed in a ranked fashion. Default ranking strategy should be to rank by start time. But we should be able to plugin more strategies like Doctor's rating etc in future.

## Other Notes:

1. Do not use any database or NoSQL store, use in-memory data-structure for now.
2. You are free to use any language of your choice.
3. Do not create any UI for the application.
4. For demo purposes, write a driver class which will execute all the commands at one place in the code and have test cases to test multiple users.
5. Work on the expected output first and then add good-to-have features of your own.
6. **Please prioritize code compilation, execution and completion.**

## Expectations:

1. Make sure that you have working and **demonstrable code.**
2. Make sure that code is functionally correct.
3. Code should be modular and readable.

4. Separation of concern should be addressed.
5. Code should easily accommodate new requirements with minimal changes.
6. Code should be easily testable.
7. Code should have proper error handling

**Sample Test cases:**

The input/output need not be exactly in this format but the functionality should remain intact

```
i: input
o: output
i:registerDoc -> Curious-> Cardiologist
o: Welcome Dr. Curious !!
i: markDocAvail: Curious 9:00-10:30
o: Sorry Dr. Curious slots are 60 mins only
i: markDocAvail: Curious 9:00-10:00, 12:00-13:00, 16:00-17:00
o: Done Doc!
i:registerDoc -> Dreadful-> Dermatologist
o: Welcome Dr. Dreadful !!
i: markDocAvail: Dreadful 9:00-10:00, 11:00-12:00, 13:00-14:00
o: Done Doc!
i: showAvailByspeciality: Cardiologist
o: Dr.Curious: (9:00-10:00)
o: Dr.Curious: (12:00-13:00)
o: Dr.Curious: (16:00-17:00)
i: registerPatient ->PatientA
o: PatientA registered successfully.
i: bookAppointment: (PatientA, Dr.Curious, 12:00)
O: Booked. Booking id: 1234
i:showAvailByspeciality: Cardiologist
o: Dr.Curious: (9:00-10:00)
o: Dr.Curious: (16:00-17:00)
i: cancelBookingId: 1234
o: Booking Cancelled
i: showAvailByspeciality: Cardiologist
o: Dr.Curious: (9:00-10:00)
o: Dr.Curious: (12:00-13:00)
o: Dr.Curious: (16:00-17:00)
i: bookAppointment: (PatientB, Dr.Curious, 12:00)
o: Booked. Booking id: 5678
i:registerDoc -> Daring-> Dermatologist
o: Welcome Dr. Daring !!
i: markDocAvail: Daring 11:00-12:00 14:00-15:00
o: Done Doc!
i: showAvailByspeciality: Dermatologist
o: Dr.Dreadful: (9:00-10:00)
o: Dr.Dreadful: (11:00-12:00)
```

o: Dr.Daring: (11:00-12:00)  
o: Dr.Dreadful: (13:00-14:00)  
o: Dr.Daring:(14:00-15:00)  
i: bookAppointment: (PatientF, Dr.Daring, 11:00)  
o: Booked. Booking id: 5587  
i: bookAppointment: (PatientA, Dr.Curious, 12:00)  
o: Booked. Booking id: 5678  
i: bookAppointment: (PatientF, Dr.Curious, 9:00)  
o: Booked. Booking id: 5280  
i: bookAppointment: (PatientC, Dr.Curious, 16:00)  
o: Booked. Booking id: 5701  
i: showAvailByspeciality: Cardiologist  
o: Dr.Curious: No slots available  
i: bookAppointment: (PatientD, Dr.Curious, 16:00, waitlist=true)  
o: Added to the waitlist. Booking id: 5710  
i: cancelBookingId: 5701  
o: Booking Cancelled  
o: Booking confirmed for Booking id: 5710  
i: showAppointmentsBooked(PatientF)  
o: Booking id: 5280, Dr Curious 9:00  
o: Booking id: 5587 , Dr Daring 11:00