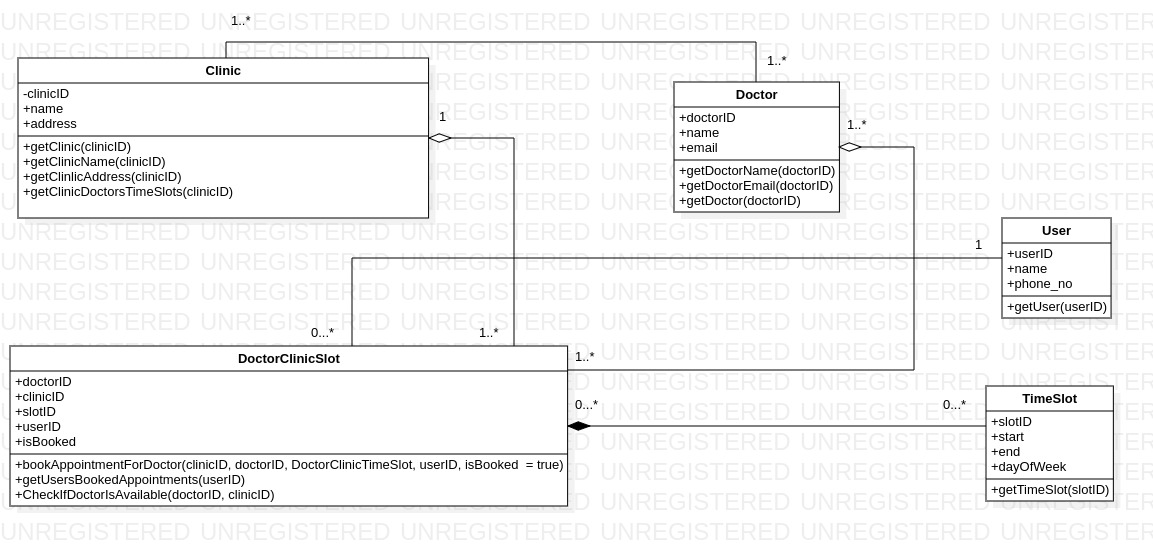
**Case Study Round**

**Task 1 Solution**

**Requirements :-**

* Users can set appointments with doctors.
* Users can pick their favorite doctor and set up with appointment at a convenient time.
* One Doctor can work at multiple clinics.
* Multiple Clinics can have multiple doctors.
* Users should be able to set an appointment with doctors at a time slot he is free ( for a particular clinic).
* One User can book multiple appointments at multiple clinics with multiple doctors.

**LLD Design Using UML Diagram**

****

**List Of Some Functions/Requests :-**

**1. bookAppointmentsForDoctors() -** We would accept the doctor id and clinic id and the time slot of the doctor and check if doctor is free using CheckIfDoctorIsAvailable() module if so then send a message stating its already booked else book the slot. This way we take care if the doctor is free for a slot for a particular clinic or not. Users can set up appointments on the same day with the same doctor in different clinic also.

2. **CheckIfClinicHasClashingTimeSlotOnSameDay()**- This can help us check if the same clinic does not have clashing time slot for a doctor by accepting params like doctorID, clinicID, and TimeSlot t1 and TimeSlot t2 and this would be a simple check to see if t1 or t2 are not in the same range. This module should be called whenever we are adding time slots for doctors in a partiuclar clinic.

**2. getUsersBookedAppointments() -** Will get us the booked slots for the user.

**Rest of the functions are getters and setters to get and set certain properties of our classes.**

Task 2

1. Time Complexity is O(n\*(sqrt((n-5)/6))) and Space Complexity is O(n).

7. UI Wireframe link - <https://www.figma.com/file/oGkGUKR6wtZJJxWPb7NFSK/Kennet-Task-2-UI-Wireframe?type=design&node-id=0%3A1&mode=design&t=6tqRIMhPz5vucd9w-1>

8. Github Repo Link -<https://github.com/shazm12/kennet_hiring_challenge/tree/master/task_2>