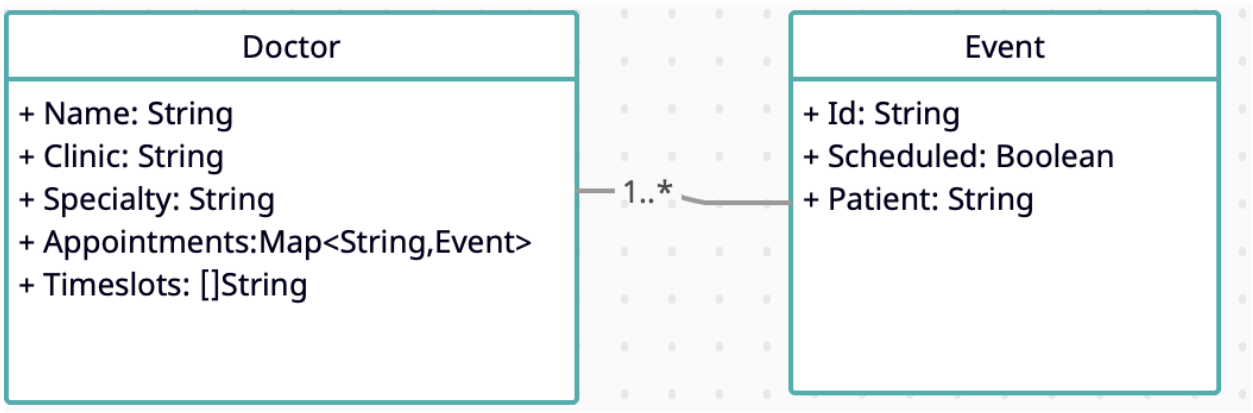
1. UML class diagram



2. Queries to support

QueryName	Input	Output
doctorByName	nameOfDoctor	Doctor
availabletimeslotByDoctorName	nameOfDoctor	Timeslots

3. Mutations

bookAppointment	nameOfDoctor, patientName, timeslotChosen	EventId
cancelAppointment	nameOfDoctor,Event id	Success(boolean)
updatePatientname	nameOfDoctor, Event id, newPatientName	Success(boolean)

4. Endpoints

Action	Method	Url
doctorByName	GET	/doctor/{doctorName}
availabletimeslotByDoctorName	GET	/timeslots/{doctorName}

bookAppointment	PUT	/appointment/create
cancelAppointment	DELETE	/appointment/cancel
updatePatientname	PUT	/appointment/update

1.2 Testcases design subtask:

Testcase Identifier	Testcase Description	Inputs	Expected Output	Remarks
doctorByName	Doctor exists, check return of doctor info	doctorName	Doctor Info	
doctorByName	Doctor does not exist, check empty	doctorName	empty	Doctor does not exist
availabletimeslotByDoctorName	Doctor exists,check return of timeslots	doctorName	timeslots	
availabletimeslotByDoctorName	Doctor does not exist, check error	doctorName	error	Doctor does not exist
bookAppointment	Doctor exist, Time Slot available, check after booking, the timeslot should not be available	doctorName, timeSlot, patientName	EventId	
bookAppointment	Doctor does not exist, check error	doctorName, timeSlot, patientName	error	Doctor does not exist

cancelAppointment	Given eventId and doctor name, delete an event, Check after deleting, timeslots add one more and event not exist any more	eventId, doctorName	Success	
cancelAppointment	Event does not exist	eventId, doctorName	Fail	Event does not exist
updatePatientname	Given eventId, and doctorName, find the event and update the patient name, Check the patient name of the event after update	eventId, doctorName, patientName	patientName	
updatePatientname	Event does not exist	eventId, doctorName	Fail	Event does not exist