

## 1.1 Design subtask

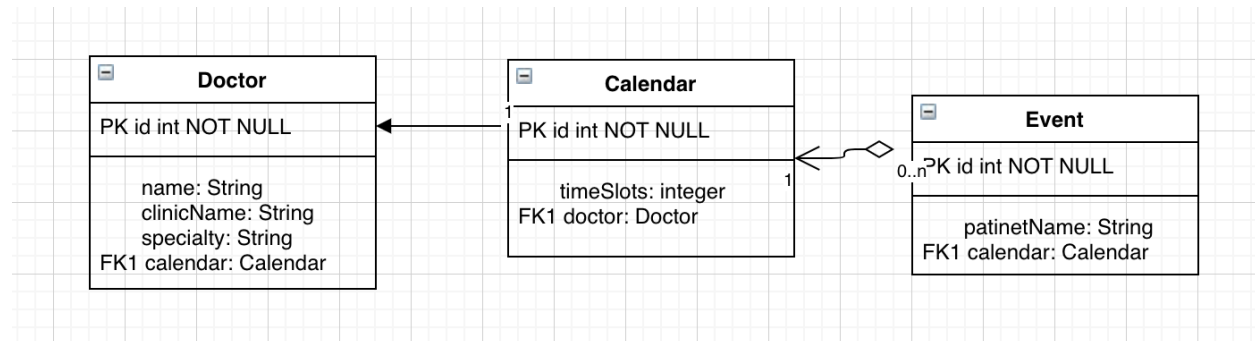
- Schema of GraphQL

```
Type Doctor {  
  id: ID!  
  name: String  
  clinicName: String  
  specialty: String  
}
```

```
Type Calendar {  
  id: ID!  
  Doctor: Doctor  
  timeSlots: integer  
}
```

```
Type Event {  
  id: ID!  
  Calendar: Calendar  
  patientName: String  
}
```

UML:



- Various queries that API will support (name, inputs, and outputs)

1. Purpose: get doctor details
2. Name: doctor
3. Input: name of doctor, String
4. Output: doctor object, Doctor

1. Purpose: get doctor's available timeslots for today
2. Name: calendar
3. Input: doctor name, String
4. Output: calendar object, Calendar

- Mutations that API will support (name, inputs, and outputs)

1. Purpose: book an appointment with a doctor
2. Name: bookAppointment
3. Input: doctor name, String
4. Output: event created, Event

1. Purpose: cancel an appointment with a doctor
2. Name: cancelAppointment
3. Input: doctor name, String, patient name, String
4. Output: whether successful, boolean

1. Purpose: update name of the patient of an appointment with a doctor
2. Name: updateAppointment
3. Input: old name, String, new name, String
4. Output: updated new event, Event

- Endpoint of API

<http://localhost:8000/DoctorCalendar>

## 1.2 Testcases design

Testcase Identifier	Testcase Description	Inputs	Expected Output	Remarks
Q1.1	Testcase for query 1, valid	Valid doctor name	Doctor details including name, clinic name and specialty	/
Q1.2	Testcase for query 1, invalid	null	error	Doctor name is required and cannot be null here
Q2.1	Testcase for query 2, valid	Valid doctor name	Number of available timeslots left of the doctor	/
Q2.2	Testcase for query 2, invalid	null	error	Doctor name is required and cannot be null here
M1.1	Testcase for mutation 1, valid	Valid doctor name	event	Return the new event created

M1.2	Testcase for mutation 1, invalid	Doctor name does not exist in the system	error	
M2.1	Testcase for mutation 2, valid	Patient name and doctor name	true	Cancel successfully
M2.2	Testcase for mutation 2, invalid	Doctor name	error	Patient name is also required, error returned because of missing parameter
M3.1	Testcase for mutation 3, valid	Valid old patient name and new patient name	New updated event	/
M3.2	Testcase for mutation 3, invalid	Invalid old patient name and new patient name	error	Cannot find the old patient name in the system