

Assignment 2.1 – GraphQL API Design

Identifying Participants (Only based on current requirements)

1. Doctors
2. Patients

Creating API Schema

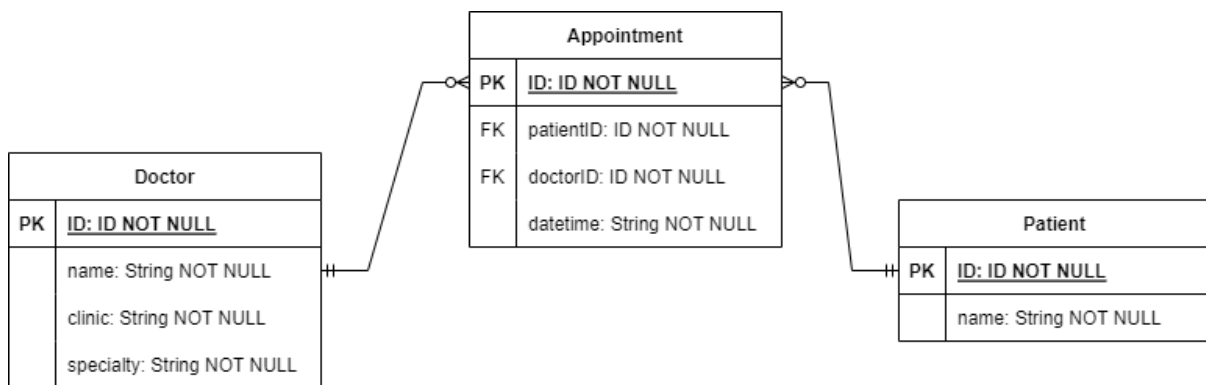


Fig1: ER Diagram for Schema of API

```
type Doctor {
  id: ID!
  name: String!
  clinic: String!
  specialty: String!
  calendar: [Appointment!]
}

type Patient {
  id: ID!
  name: String!
  calendar: [Appointment!]
}

type Appointment {
  id: ID!
  doctor: Doctor!
  patient: Patient!
  datetime: String!
}
```

Fig2: Schema definitions for the API

API Queries

QUERY: DOCTORS

Description: Gets list of doctors with their details

Input: None

Output: JSON object containing details of doctor

Output Fields: DoctorID, Name, Clinic, Specialty

QUERY: APPOINTMENTS

Description: Gets list of today's appointments for a certain doctor

Input: ID of doctor we want to fetch appointments for

Input Fields: DoctorID

Output: List of appointments happening today for specified doctor

Output Fields: [Appointment]: AppointmentID, Doctor, Patient, Datetime

API Mutations

MUTATION: ADDAPPOINTMENT

Description: Books an appointment for a certain doctor today

Input: ID of doctor & patient, and Date & Time of appointment

Input Fields: DoctorID, PatientID, DateTime

Output: Newly created appointment object

Output Fields: [Appointment]: AppointmentID, Doctor, Patient, Datetime

MUTATION: CANCELAPPOINTMENT

Description: Cancels an existing appointment

Input: ID of appointment

Input Fields: AppointmentID

Output: Newly deleted appointment object

Output Fields: [Appointment]: AppointmentID, Doctor, Patient, Datetime

MUTATION: MODIFYAPPOINTMENT

Description: Modifies an appointment for a certain doctor

Input: ID of appointment and new patient ID

Input Fields: AppointmentID, PatientID

Output: Newly modified appointment object

Output Fields: [Appointment]: AppointmentID, Doctor, Patient, Datetime

API Endpoint

<http://localhost:4000/graphql>

API Tests Documentation

TestCase Identifier	TestCase Description	Inputs	Expected Output	Remarks
Get Doctor	Gets list of doctors	None	List of Doctor Objects with requested fields	Happy Path TestCase. We can verify list of doctors returned.
Get Doctor Invalid Query	Gets list of doctors, but unwanted query is added	Unwanted Query like specialty details or clinic name	Error object citing unwanted fields provided	Error TestCase for getting doctor's details.
Get Appointments	Gets a list of today's appointments for specified doctor	ID of Doctor whom we want to fetch appointments for	List of Appointments Objects for requested doctor	Happy Path TestCase. We can verify list of appointments returned.
Get Appointments invalid doctor	Gets list of appointments, but specified doctor does not exist	ID of Doctor whom we want to fetch appointments for	Error object saying specified doctor was not found (Error 404)	Error TestCase for getting list of appointments. We can verify if doctor does not exist by getting list of doctors.
Book Appointment	Books an appointment for a certain doctor and patient at certain date-time	ID of Doctor and Patient, and Date-Time of appointment	Newly created Appointment object	Happy Path TestCase. We can verify if appointment is added by getting list of appointments.

Book Appointment Invalid Doctor	Books an appointment for a doctor and patient at date-time, but doctor does not exist	ID of Doctor and Patient, and Date-Time of appointment	Error object saying specified doctor was not found (Error 404)	Error TestCase for booking new appointment. We can verify if doctor does not exist by getting list of doctors.
Cancel Appointment	Cancels an existing appointment	ID of Appointment to be cancelled	Newly deleted Appointment object	Happy Path TestCase. We can verify if appointment is deleted by fetching list of appointments.
Book Appointment Invalid Appointment	Cancels an appointment, but the appointment does not exist	ID of Appointment to be cancelled	Error object saying specified appointment was not found (Error 404)	Error TestCase for cancelling appointment. We can verify if appointment does not exist by getting list of appointments.
Modify Appointment	Changes Patient in existing appointment	ID of Appointment and Patient	Newly modified appointment object	Happy Path TestCase. We can verify if appointment is modified by fetching list of appointments.
Modify Appointment invalid Patient	Changes patient in existing appointment, but new patient does not exist	ID of Appointment and Patient	Error object saying specified patient was not found (Error 404)	Error TestCase for modifying appointment.
Modify Appointment invalid Appointment	Changes patient in appointment, but appointment does not exist	ID of Appointment and Patient	Error object saying specified appointment was not found (Error 404)	Error TestCase for modifying appointment. We can verify if appointment does not exist by getting list of appointments.