# DOMAIN MODELING

Logo

Description automatically generated with medium confidence

# Assignment\_09: #Exercise: Domain Modeling – Dentist company B.V.T.

**Warning**

This is a big exercise. It can take up to at least 16 hours to complete this exercise. Take the time to read the instructions and requirements carefully. If you get stuck or have any questions, ask them in Slack!

You have started to program as a freelance developer! Your first customer is Dentist company B.V.T.!

A picture containing sky, cake, decorated

Description automatically generated

They want you to build a system in which you can manage appointments for dentists and assistants.

The focus will be on correctly modeling different data entities and operations (changes). Properly managing the state of the application will be important.

To make calculating with days easier, you can use numbers as days such as e.g. Monday is 1, Tuesday is 2, etc. For now, each month will consist of 4 weeks, which is 28 days. All appointments are made within this month. Right now we are in the month before the month where all the appointments are made, so it is still possible to change the dates.

To make dealing with time easier, each appointment takes 1 hour and starts at times as e.g. 9:00h-10:00h, 14:00h-15.00h, etc.

You may choose your names, phone numbers, email addresses for people in the system. With the help of [https://www.mockaroo.com](https://www.mockaroo.com/), you can create your own 'mock' API. You can use this to generate and retrieve random person data. It's also fine to use your own (mock) API.

Implement the following functionalities:

* The company has dentists.
* The company has assistants.
* Each dentist and assistant has a first name, last name, phone number, and e-mail address that ends with "dentistcompanybvt.com".
* Dentists and assistants can become sick. They won't be able to work that day.
* The company has clients.
* Clients can become sick, so the appointment has to be canceled and removed.
* Each client has a first name, last name, phone number, email address, and birth year.
* An appointment is always with **one** client.
* An appointment is always with **one** dentist.
* An appointment sometimes has an assistant.
* An appointment has a date (day number) and time (whole hours).
* The practice is closed on the weekend.
* A dentist or assistant cannot have two appointments at the same time.

When you start your React app, the following entities have to be in your system:

* 4 dentists
* 2 assistants
* 50 clients
* 150 appointments

Use one JaveScript object with all data as the state of the app.

**Views**

We have prepared the views for you. You won't have to create any new HTML and CSS code but you will generate existing HTML elements based on the app state.

Note:

* In the calendar view, the appointments are currently not sorted based on time. You still have to build this.
* In the day view, the appointments are not sorted on time. You will also have to implement this.
* When an appointment does not have a dentist due to e.g. sickness then give this appointment a red background color.

**Download the view components:**

* $git clone <https://github.com/WincAcademy/dentist_react> in a new directory (map) on your computer
* open the terminal in this new directory and use the commands npm install and npm start.
* open the app at http://localhost:3000

**Operations/changes**

Life is hectic at a busy dentist's practice. Throughout the day there are many changes in the system. Below you will find a list of all the operations that need to happen. It's your task to implement these operations and data mutations.

Make these changes "immutable".

You will get function calls but the implementation of the function still has to be built. The implementation will depend on how you deal with the state in the app.

- add a dentist: newState = addDentist(state, "Toos", "Trekker", "06-12345678", "toos@tandartspraktijkbvt.nl")

- add a client: newState = addPatient(state, "Piet", "Auw", "06-12345679", "piet@wincacademy.nl", 1985)

- a dentist becomes sick. Give each of his or her appointments a red background colour in the views: newState = makeDentistSick(state, dentistId)

- add an appointment without an assistant: newState = addAppointment(state, dayNumber, time, patientId, dentistId) Note: an appointment on a day + time can only be added when the choosen dentist and/or assistant is available.

- add an appointment with an assistant: newState = addAppointment(state, dayNumber, time, patientId, dentistId, assistentId)

- delete an appointment: newState = removeAppointment(state, appointmentId)

- a client is sick, delete his appointments: newState = makePatientSick(state, patientId)

- move an appointment: newState = moveAppointment(state, appointmentId, newDayNumber, newTime) Note: dentists and assistants can't have two appointments simultaneously.

**Bonus**

* Treatment types:
  + each dentist has a set of skills. E.g. pulling teeth, dental fillings, surgery, etc.
  + not every dentist has each skill.
  + each appointment has one treatment type.
  + in the calendar, view you cannot see the treatment type.
  + in the day view, you can see the treatment type.
* Make a working form and buttons for all operations.