**Dental Company:**

* name: String
* owner: String

branch = []

**Branch:**

* name: String
* city: String
* contactNumber: String

staff = []

patients = []

services = []

appointments = []

**Service:**

* name: String
* description: String
* duration: Float
* cost: Float

**Person:**

* firstName: String
* lastname: String
* gender: ENUM
* dateOfBirth: Date
* email: String

**Staff:**

* employeeID: String
* role: ENUM

**Patients:**

* phoneNumber: String
* insuranceStatus: ENUM

**Appointment:**

* date: Date
* time: Time
* patientName: String
* staffName: String
* serviceName: String
* estimatedDuration: Float
* status

**Bill:**

* date: Date
* patient: String
* dentalBranch: String
* services: String
* totalCost: Float
* VAT: Float
* discounts: Float
* insurance: ENUM
* paymentMethod: ENUM

**Relationship:**

*Inheritance:*

We have the parent class person and both staff and patient inherit from this parent class. Patient and staff are considered the child classes which inherit the attributes of the parent class and have some (unique) attributes of their own.

*Composition:*

We can see a composition relationship between the class branch and appointment. The class branch can be the container and the class appointment are the contained class. When we destroy branch then the appointments will be automatically destroyed with it. Let’s take the scenario where a dental branch closes and the dentists become unemployed or start working at another dental company. These dentists’ appointments will be cancelled because the initial place they were booked at was “destroyed.” Therefore, we can see that here is total dependency between these two classes.

*Aggregation:*

There is an aggregation between the class and staff and branch. In which we say that the branch “has a” staff. Whereby the branch can have on or more staff members. Additionally, it is aggregation because we can say that these two classes can exist independently. A staff can exist is the class branch is destroyed because the different employees can find a job some place else. And although it may not be efficient the branch can still exist without a staff.

***Ask professor:***

For appointment will it inherit from the other classes because or can I simply rewrite the info

Ask prof about the methods I created