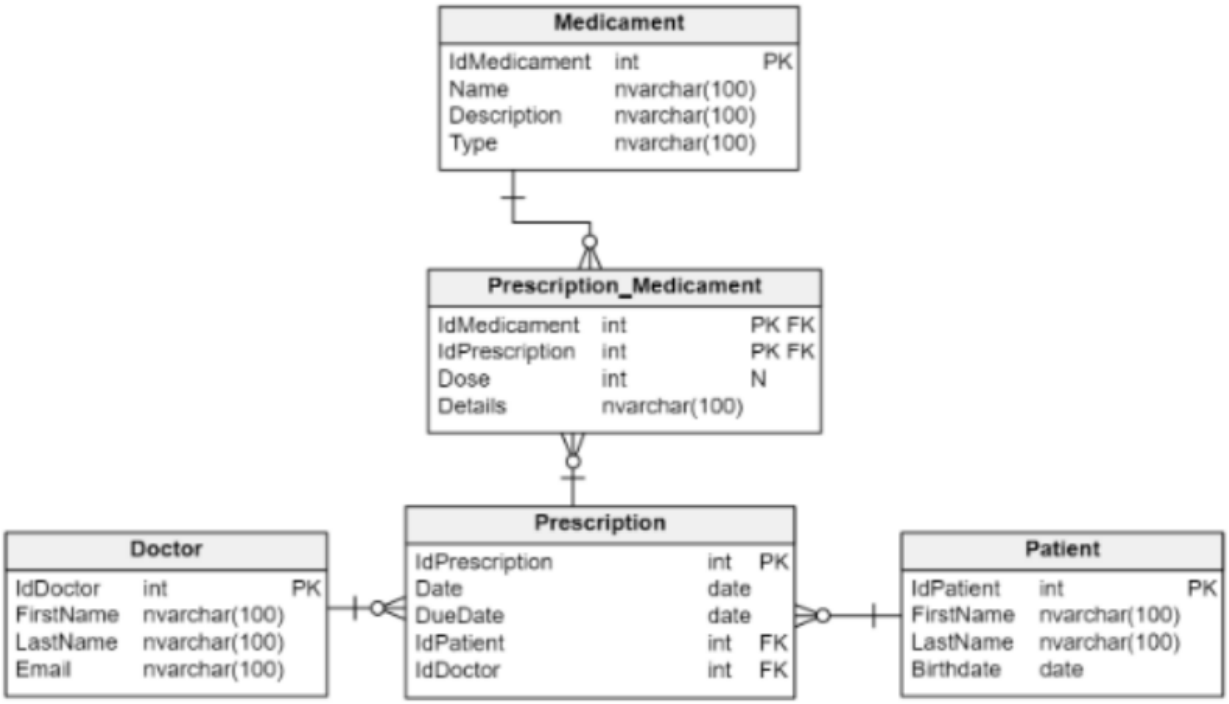
In this task, prepare an application based on the EF code-first approach. Implement the following endpoints:



Adding a new prescription

- An endpoint that allows issuing a new prescription. The endpoint should accept patient information, prescription information, and information about the doctor listed on the prescription as part of the request.
- If the patient provided in the request does not exist, insert a new patient into the Patient table.
- If the medication listed on the prescription does not exist, return an error.
- A prescription can include a maximum of 10 medications. Otherwise, return an error.
- We must check if DueDate >= Date.

Request example:

```
{
  "patient": {
     "IdPatient": 1,
     "FirstName": "John",
     //...
  },
  "medicaments": [
       {"IdMedicament": 1, "Dose": 3, "Description":
"Some description..." }
  ],
  "Date": "2012-01-01",
  "DueDate": "2012-01-01"
}
```

Patient data:
Prepare an endpoint that allows displaying all data about a specific patient, along with a list of prescriptions and medications they have taken. We would like the response to include all information about the medications and doctors. The data about prescriptions should be sorted by the DueDate field.

Response example:

```
{
    "IdPatient": 1,
    "FirstName": "John",
    //...
    "Prescriptions": [
        {
           "IdPrescription": 1,
           "Date": "2012-01-01",
           "DueDate": "2012-01-01",
           "Medicaments": [
                       {
                          "IdMedicament": 1,
                          "Name": "AAA",
```

```
                        "Dose": 10,
                        "Description": "AAA"
                    }
                ],
                "Doctor": {
                    "IdDoctor": 1,
                    "FirstName": "AAA"
                }
            }
        ]
}
```

Try to prepare unit tests to check your endpoints.

Ensure that:

- Naming conventions are correct.
- The code is divided into separate layers and is testable.
- Remember the principles of REST, SOLID, DRY, YAGNI, KISS.
- Consider optimistic/pessimistic locking (if necessary).