ChipSoft Assessment

Inleiding

Voordat ik mijn keuzes bij het maken van deze opdracht verantwoord, wil ik enkele opmerkingen maken. Door de deadline van vier uur (ik weet dat meer mocht, maar heb besloten het bij de vier uur te houden) heb ik bepaalde zaken niet gedaan zoals ik ze zou doen wanneer ik meer tijd zou hebben. Bij de start heb ik ook het project naar de laatste dotnet versie overgezet om de laatste functionaliteit te kunnen gebruiken.

Indeling Solution

Er zijn verschillende manieren om een project op te delen op solution niveau, ik heb ervoor gekozen om het simpel te houden door één extra project toe te voegen dat alle logica bevat om de applicatie ook door een andere weergave te kunnen gebruiken (zoals een Web API bijvoorbeeld). In de praktijk, bij grote projecten die uit veel meer delen bestaan dan dit project, zou ik absoluut mijn één project opdelen in meerdere projecten om de herbruikbaarheid te verbeteren (Application, Infrastructure, Models, ...). Ik ben zelf een fan van YAGNI en KISS, waardoor ik dit nu niet heb gedaan.

DbContext

Ik heb de DbContext die bij de aanvang van het project aanwezig was, verplaatst naar het application project. Hierdoor kan deze hergebruikt worden wanneer er een andere weergave zou komen (zoals een Web API). Verder heb ik deze aangevuld met de DbSets die nog ontbraken en heb ik alle configuratie voor elke entiteit zijn eigen configuratie klasse gegeven.

Services

Het project bestaat uit enkele services die samen alle functionaliteit die de gebruiker nodig heeft bevatten. Ik ben geen fan van entiteiten rechtstreeks in de “Frontend” te gebruiken, waardoor ik ervoor kies om DTO’s hiervoor te maken die telkens enkel de informatie bevatten die opgevraagd is. Als datatype voor deze DTO’s opteer ik ook om van record gebruik te maken, aangezien deze makkelijk te definiëren zijn.

Validatie

Aangezien ik weinig informatie had over de data die effectief aanwezig moeten zijn voor een patiënt, arts of afspraak, is de validatie voor de gegevens die mij belangrijk lijken ook minder uitgebreid dan wat ik mij kan voorstellen. In een groter project zou ik deze validatie niet definiëren in mijn services, maar zou ik gebruikmaken van FluentValidations, aangezien deze meer leesbaar zijn. Ook zou ik zelf geen exceptions gebruiken voor validatiefouten, maar zou ik discriminated unions gebruiken door de package OneOf (binnenkort misschien te vervangen door type unions in c# zelf) te gebruiken.

Testing

Naar het einde van de opdracht had ik nog ongeveer 30 minuten over waar ik lang heb getwijfeld of ik testen zou toevoegen aan de opdracht. Ik heb besloten om mijn code nog eens te doorlopen en de testen niet toe te voegen, aangezien ik dan ver buiten de vier uur zou gaan. Indien ik wel meer tijd zou nemen, zou ik wat integratietesten toevoegen om na te gaan dat alle services werken zoals verwacht. Hiervoor zou ik dan ook Testcontainers gebruiken om een echte databank te laten starten om zo accuraat mogelijke resultaten te hebben.

Slot

Als laatste zou ChipSoft willen bedanken ongeacht het resultaat van dit assessment. Indien er toch nog bepaalde keuzes niet duidelijk zouden zijn, mag u mij altijd contacteren. Ik kijk alvast uit naar de feedback.