# Chipsoft Assignment EPDConsole

## Layered Architecture

Het project is gestructureerd in verschillende lagen:

Data Access Layer (DAL): gebruikt Entity Framework voor interactie met de database (EPDDbContext).

Business Logic Layer (BLL): omvat de kernlogica binnen PatientService, PhysicianService en AppointmentService.

Validatielaag: Afzonderlijke validators (PatientValidator, PhysicianValidator, AppointmentValidator) handhaven bedrijfsregels en gegevensintegriteit.

Presentatielaag: De ConsoleUI beheert de gebruikersinteractie.

Bevordert Separation of Concerns, waardoor de codebase modulair en gemakkelijker te onderhouden wordt. Maakt mogelijke wijzigingen mogelijk (bijvoorbeeld het verwisselen van de database of het toevoegen van een frontend webapplicatie) zonder andere lagen te beïnvloeden.

## Validation as a Separate Concern

Validatielogica is ingekapseld in statische validatorklassen (PatientValidator, PhysicianValidator, AppointmentValidator). Dit houdt de bedrijfslogica schoon en gericht op kernverantwoordelijkheden . Validatielogica kan onafhankelijk worden hergebruikt en uitgebreid, waardoor de onderhoudbaarheid wordt verbeterd.

## Exception Handling

Ik heb hier een custom exception aangemaakt namelijk ValidationException en try-catch blocks in services gebruikt.

## Readability and Sustainability

Er zijn pogingen gedaan om de broncode schoon te houden, met beschrijvende methodenamen, modulair ontwerp en commentaar.

Bevordert samenwerking en begrip voor toekomstige ontwikkelaars. Vermindert de technische schulden en zorgt voor duurzaamheid op de lange termijn.

## User Feedback

De applicatie geeft duidelijke feedback bij ongeldige invoer en fouten (bijvoorbeeld ongeldige datums of dubbele verzekeringsnummers).

Verbetert de gebruikerservaring door de gebruiker te begeleiden en frustratie te minimaliseren. Garandeert de gegevensintegriteit door te voorkomen dat ongeldige gegevens aan de database worden toegevoegd.

## Scalability for New Features

Het project is ontworpen om tegemoet te komen aan toekomstige eisen:

Extra velden in modellen (bijvoorbeeld facturatiegegevens toevoegen aan Patiënt).

Nieuwe entiteiten (bijvoorbeeld Factuur, Behandelplan) kunnen eenvoudig worden toegevoegd. Uitbreiding van de gebruikersinterface voor geavanceerde functionaliteiten.