# Specifikation og analyse

Her snakker man om de ting man har valgt ved teknologiundersøgelser.  
Beskrivelse af specifikations- og analysearbejdet. Dvs. de overvejelser man har gjort – de løsninger man har valgt og begrundelsen herfor. En domæne model vil være relevant at tilføje i dette afsnit.

Da udviklingen af WPF-applikationen gik i gang, var det meget nyt. Gruppen lærte undervejs hvordan det skulle programmeres, og der var ikke store overvejelser omkring hvilken arkitektur programmet skulle bygges op efter. Dette har givet nogle problemer med at bevare overskueligheden, og holde koden let at vedligeholde. I slutningen af april blev der foretaget en analyse af MVC, MVP og MVVM, og om det kunne betale sig at refakturere koden, for at overholde disse arkitekture. Gruppen fandt ud af at det ville have været en god idé at opbygge WPF-applikationen efter MVVM arkitekturen, helt fra starten af. Hvis det var sket, ville af mængden af kode i codebehind været blevet reduceret væsentligt. Problemet med kode i codebehind er at det gør det vanskeligt at teste softwaren i form af unittests. MVC og MVP blev også undersøgt. Grundlæggende har det vist at WPF-applikationen tildels følger MVC/MVP arkitekturen. Der er et View i form af XAML, presenter/controller i form af codebehind, et BLL og et DAL lag. Det ville have været en fordel at fjerne koden i codebehind, og rykke det ud i et lag for sig selv. Fordelene ved at gøre dette kunne dog ikke gøre op for den tid det ville tage at skrive koden om. Gruppen besluttede derfor at beholde koden som den er, men at lære af det og bruge den nye viden i forbindelse med udviklingen af Web-applikationen.

Figur : MVC/MVP til 3-lags arkitektur

Et andet stort problem har været mapningen imellem objekt-udgaven af en entity i koden, og dens relationelle database-udgave. Der er brugt lang tid på at få denne mapning til at være korrekt, og sørge for at udgaven i koden er den samme som den eksisterende i databasen. Løsningen på dette blev fundet igennem undervisningen i I4DAB. Grundet den fremskredne udvikling i WPF-applikationen blev det ligeledes besluttet at det ikke kunne betale sig at refakturere koden, men at implementere det som en del af Web-applikationen.

I Web-applikationen er der også blevet indført et GOF-pattern ved navn ”Facade”. Facade mønsteret har i dette tilfælde til opgave at give et enkelt adgangspunkt til DAL-laget. I forbindelse med DAL-laget er der et repostory for hver tabel i databasen og en databasekontekst. For at overskueliggøre dette er der lavet en facade, så man i den kode der skal bruge disse elementer blot opretter en facade, og ikke alle de forskellige elementer. Dette gør koden mere vedligeholdelsesvenlig, og overskueliggøre koden set fra et højere abstraktionsniveau.

## Entity Framework

* Kom ind på domænemodel

## Facade

## Web-applikation VS Web-site, HTML, CSS, JavaScript.

Noter:

Fra referat:

i. Skulle have brugt MVVM for at kunne teste ordenligt. For omfattende at implementere nu. Bliver implementeret igennem Web-App.

ii. Husk at få alle overvejelser med i rapporten!

D:\IHA\4. Semester\I4SWD\Lectures

* 13\_GUI\_ArchitecturePatterns - MVC, MVP
* 14\_ModelViewViewModel - MVVM

DAB slides - 206 Entity framework

ADO.NET intro slide 36

Ufærdige afsnit skrives disse af Mathis?  
Er noterne til dig selv, eller er det til reference som Kristoffer skal sørge for?