MVVM

I forbindelse med produktets brugerinterface er der blevet anvendt et MVVM[[1]](#footnote-1) pattern. Dette er blevet valgt på grund af de fordele det medbringer.

Bindingen mellem GUI’en og buisness logikken løsnes idet at viewet primært er defineret i XAML filen, med et begrænset code-behind, som ikke indeholder noget buisness logik. Ved at binde en ICommand til view modellen, kan der kaldes funktioner fra viewet nede i view modellen. Derudover kan der oprettes events i GUI’en, som er bindet til et objekt, og lytter efter en trigger i view modellen.



Figur 1 - MVVM model

For eksempel, i Pristjek220 kan en forretningsmanager tilføje en ny vare til sin forretning. Når varens navn, samt pris, er indtastet og der bliver klikket på ’Tilføj vare’, tager view modellen navnet og prisen og forsøger at sende dette videre til modellen. Hvis dette er succesfuldt (varen kunne jo allerede eksistere i databasen), bliver varen gemt i databasen og en bekræftelses tekst bliver oprettet, som et label i GUI’en er bindet til, og derved kan forretningsmanageren se at hans vare tilføjelse var en succes.

Fordi buisness logikken helt fjernes fra code-behind, bliver den uafhængig af selve GUI’en og derved bliver det muligt at udføre Unit testing på koden. Det er vigtigt at koden er testbar, fordi ellers kan der ikke garanteres at programmet fungere efter hensigten.

En anden fordel ved MVVM er, at hvis der i Pristjek220 ønskes at den generede indkøbsliste skal vise prisen i Euro og i kroner, så er view modellen oplagt til dette. En view model’s primære opgave er at tage dataen fra modellen og formatere det således at det tilpasser viewet’s ønske. Så i stedet for at tilføje prisen i euro helt nede i databasen, bliver prisen bare konverteret til euro i view modellen og derfra bindet til endnu en kolonne i den generede indkøbsliste.

1. https://msdn.microsoft.com/en-us/library/hh848246.aspx?f=255&MSPPError=-2147217396 [↑](#footnote-ref-1)