# Design, implementering og test

## Fridge app

### Design

### Implementering

### Test

## Web app

### Design

Beskrivelse af designprocessen. Hvilke designløsninger man har overvejet samt de valg man har gjort og begrundelsen herfor.

Asp.net frem for KUN html -> funktionalitet

ASP.Net Web Pages (Razor) vs. Webform

MVC -> Ikke mange overvejelser? Det var det vi lærte om?

Pro:

– Scale better

• Suitable for large web sites

• Suitable for high traffic sites

– Supports unit testing

MSSQL vs. MySQL -> nemmere synkronisering med allerede oprettet db

### Implementering

Beskrivelse af implementeringsprocessen og de færdige løsninger. Her skal man kun medtage kode og kredsløbsdiagram udsnit for særlig interessante udvalgte dele af projektet.

’Stillads’ givet af VS ved oprettelse af MVC projekt

Ikke den bedste implementeringsprocess pga. uddelegering af View og Controller -> manglende erfaring?

Evt. kodestump hvordan funktioner kaldes fra View.

Html Helpers

### Test

Da webapplikationen er bygget op omkring MVC-mønstret skal der kun testes på controller og model klasserne i projektet. Det er nemlig der funktionaliteten og logikken i applikationen ligger. Pga. den lave kobling, som dette mønstre giver, er unit tests nemme at lave. Udfordringen i dette projekt har været at koble det logik, som er lavet til applikationen, til webapplikationen. For at sikre os kontinuitet både i applikationen webapplikationen, er logikken blevet genbrugt og de tests der derfor blev lavet til applikationen kan bruges til webapplikationen. Nogen dog tilpasset, da de ikke bruger samme database.[[1]](#footnote-1)

Et af problemerne ved den måde, som funktionerne i applikationen er opbygget på, er at mange af dem returnere et View eller et ’RedirectToAction’ (Figur 1) efter et item har gennemgået anden logik i samme funktion. Et eksempel på dette kunne være i EditItemController hvor funktionen UpdateItem er implementeret. Denne funktion tager imod en FormCollection hvori de nye værdier der skal ændres i det gamle item ligger i. logikken i funktionen ændre det gamle item til de nye værdier, men returnere det view, som skal vises efter itemet er ændret. Dvs. det ikke er muligt bare at sammenligne returværdien fra funktionen med det der forventes at det pågældende item er ændret til. I stedet skal der testes på om der eksistere et item, i listen med items, der har de værdier der skulle ændres. Dette går igen med alle de funktioner der har ’ActionResult’ som retur parameter.



Figur - Udsnit af hvad der returneres fra UpdateItem-funktionen i EditItemController

## Database

### Design

### Implementering

### Test

1. Webapplikationen benytter sig af MySQL, hvorimod applikationen benytter sig af MS SQL. [↑](#footnote-ref-1)