Turist app v5 dokumentation

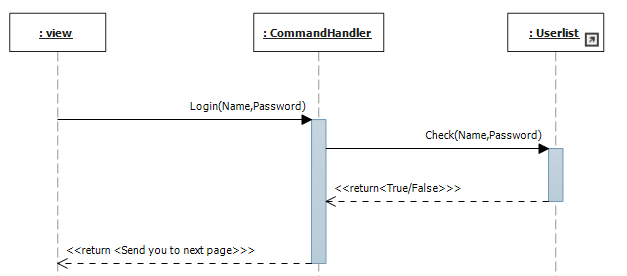
# Diagrammer

## Domain Model

## http://gabeazo.com/images/997e09746d1891ce2056.png

Vores app består af et login system og et ”galleri” som er vores måde at navigere på, vores Gallery er vores kategori, for at sætte vores ”Pictures” op, pictures er vores seværdigheder/attraktioner, som kan samles i favoritter, som bliver tildelt til den bruger der er logget ind, vores Userlist er vores Collection af oprettede brugere, som vi ligger User ind i når der bliver oprettet en bruger.

## Login sekvens diagram



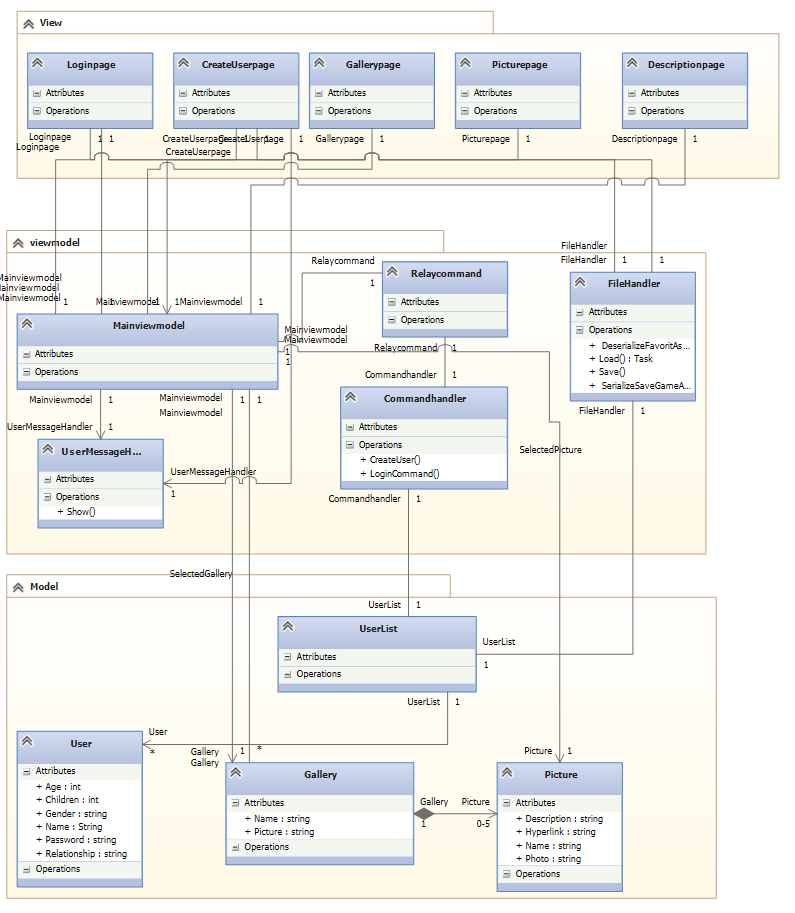
Dette er et sekvens diagram, for når du skal logge ind i appen. Du starter med at indtaste dine oplysninger, altså dit brugernavn og kodeord, og derefter trykker du på login knappen. Dette kalder så commanden login i commandhandleren, som tager de indtastede informationer, og sender dem videre til userlist. Her køres check user metoden, som checker om brugernavn og kodeord stemmer overens. Resultatet bliver så sendt tilbage via commandhandleren, hvor view modtager informationerne, og sender brugeren til næste side, hvis informationerne var rigtige. Hvis informationerne derimod ikke stemmer overens, så vil der komme en besked om, at informationerne ikke passer.

## Create user sekvens diagram

Dette er så sekvens diagrammet for når en bruger laver en ny user. Det starter med at commandhandleren tager de indtastede oplysninger, og sender dem videre til Userlisten. Her køres check user metoden, som checker om navnet er optaget. Hvis navnet er optaget sendes der en besked til brugeren, om at navnet er optaget, og de bliver så nødt til at indtaste et nyt brugernavn.

Hvis brugernavnet ikke er optaget, bliver optional delen af diagrammet udført. Her bliver en user oprettet, med de oplysninger der er blevet indtastet i viewet. Bagefter bliver useren tilføjet til userlistens liste over brugere. Bagefter sendes der en besked til viewet om at brugeren er oprettet, og derefter bliver man sendt tilbage til login siden.

## Design class diagram



Vores DCD er delt op i 3 dele. View, hvor alle vores viste sider er. Viewmodel, hvor Mainviewmodel er samt handlere og ligene. Til sidst har vi Model, og der har vi vores klasser, der indeholder den data der bliver vist på siden.

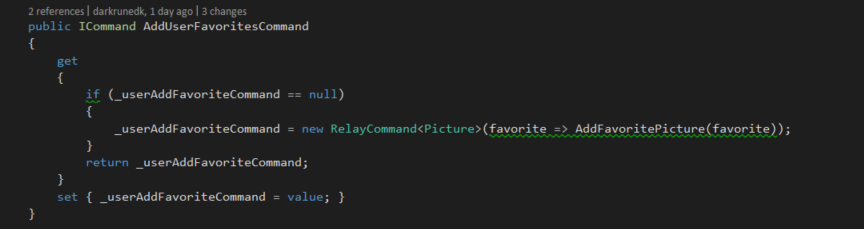
Alle klasserne i View er associeret med Mainviewmodel en til en, da de alle bliver instancieret og styret derfra, og der kun er et view af hver. Loginpage og Createuserpage har begge en association med filehandler, fordi der skal tjekkes i en local file, om brugeren eksisterer. Createuserpage kan, gennem en til en association med Usermessagemodel, udskrive en besked til Mainviewmodel, når en bruger er oprettet, eller en fejl opstod.

I model er Gallery associeret med Mainviewmodel to gange. Første er til favorit galleriet og er derfor en til en. Den anden er til de fire kategori gallerier vi har på forsiden. Den er derfor 1 til mange(\*). Hvert gallery indeholder 0 til 5 instancer af Picture klassen. Userlist klassen indeholder mange instancer af user klassen, og der bliver givet adgang til denne liste via en commandhandler, og til den via et relaycommmand.

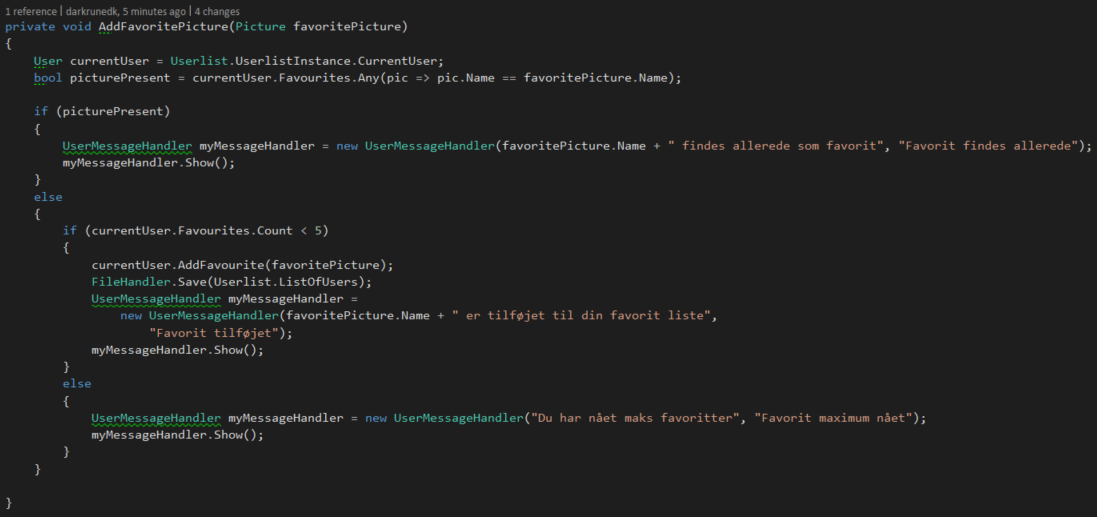
# Udvalgte dele af koden

## Commands

### Tilføj favorit billede

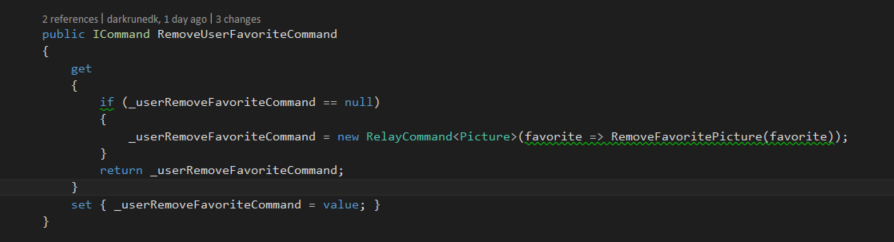


Det første der sker er, at man opretter en RelayCommand, hvis den ikke findes i forvejen i variablen. Derefter sender den informationen, som den har modtaget videre til en anden metode/funktion.

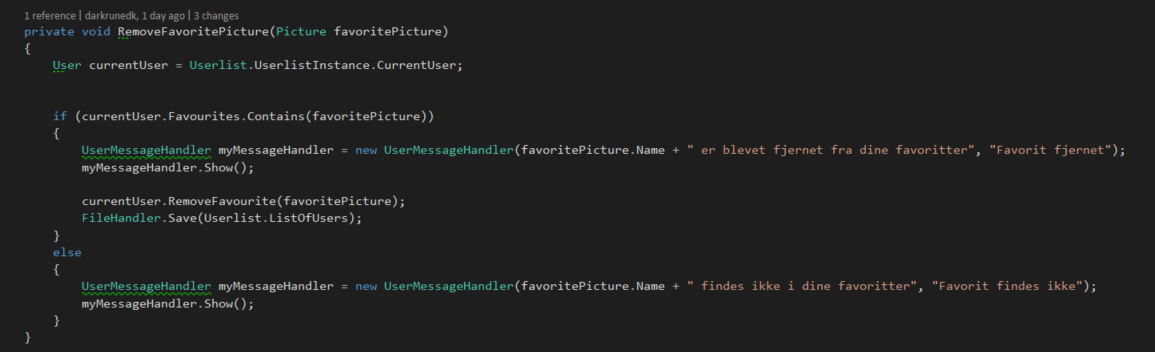


Denne funktion vil derefter tjekke om brugeren allerede har tilføjet det favorit billede, som man har prøvet at tilføje. Hvis man har det, så får man en fejl. Hvis man ikke har det, så tjekker den om man har under max favoritter. Hvis man så har det, så bliver den tilføjet, gemmer ændringerne og udskriver en besked om hvad der er sket. Hvis ikke, så får man en fejl besked.

### Fjern favorit billede

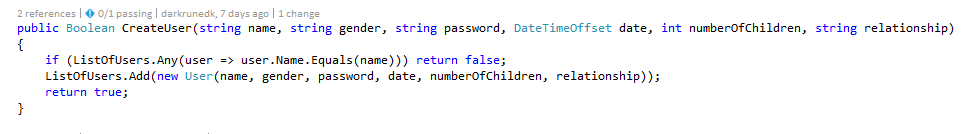


Det der sker her, er at man opretter en RelayCommand, hvis den ikke findes. Derefter sender man informationen som den har modtaget videre til en metode/funktion.



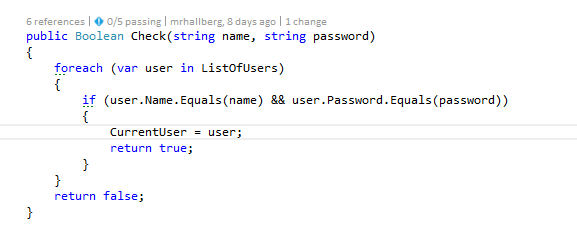
Det der sker her er, at man først tager den nuværende bruger og tjekker om billedet findes i ens favoritter. Hvis den findes, så fjerner den favoritten, gemmer ændringen og udskriver en besked om, hvad der er sket. Hvis den ikke findes, så skriver den en fejlbesked ud.

### Create user



CreateUser koden tager data fra vores createuserpage og bruger dette til at oprette en bruger, før brugeren bliver oprettet, tjekker funktionen dog først om det angivet brugernavn findes i vores database. Hvis brugernavnet findes i forvejen returner funktionen false og bruger bliver ikke oprettet. Hvis brugernavnet ikke findes i forvejen bliver bruger tilføjet til list of users og funktionen returner samtidigt en true værdi. Om der bliver retuneret true eller false afgøre så hvilken funktion der bliver kørt efterfølgende.

### Check user



Check user funktionen bruges til at tjekke om en brugernavn og kode findes i vores database, dettte bliver gjort ved at tage det indtastede brugernavn og kode, og køre en foreach løkke hvor denne data bliver sammenlignet med vores database for at tjekke om der ligende data. Hvis brugernavn og kode bliver fundet i databasen returnere funktion en true værdi, og hvis brugernavn og kode ikke findes i databasen, returneres der en false værdi. Disse 2 værdier afgøre så hvilken funktion der så bliver sat igang efterfølgende.