

Obligatorisk Opgave 1

Mikkels Bibliotek

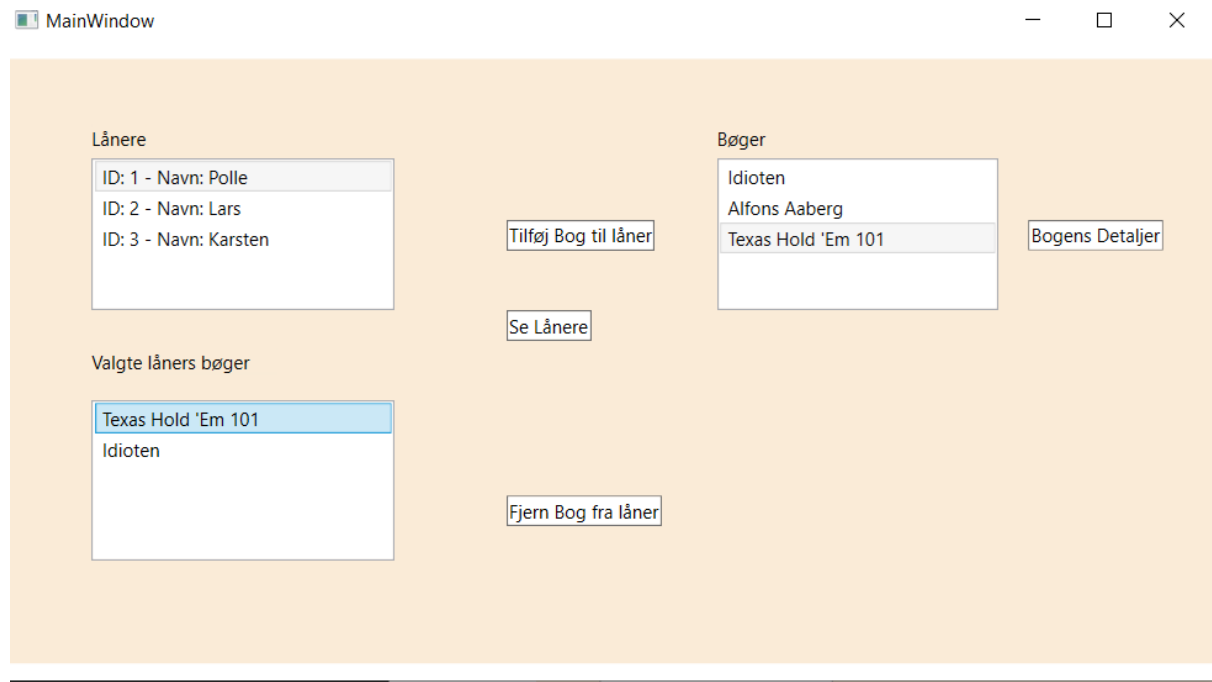
Mikkel Bak Markers

Præmissen

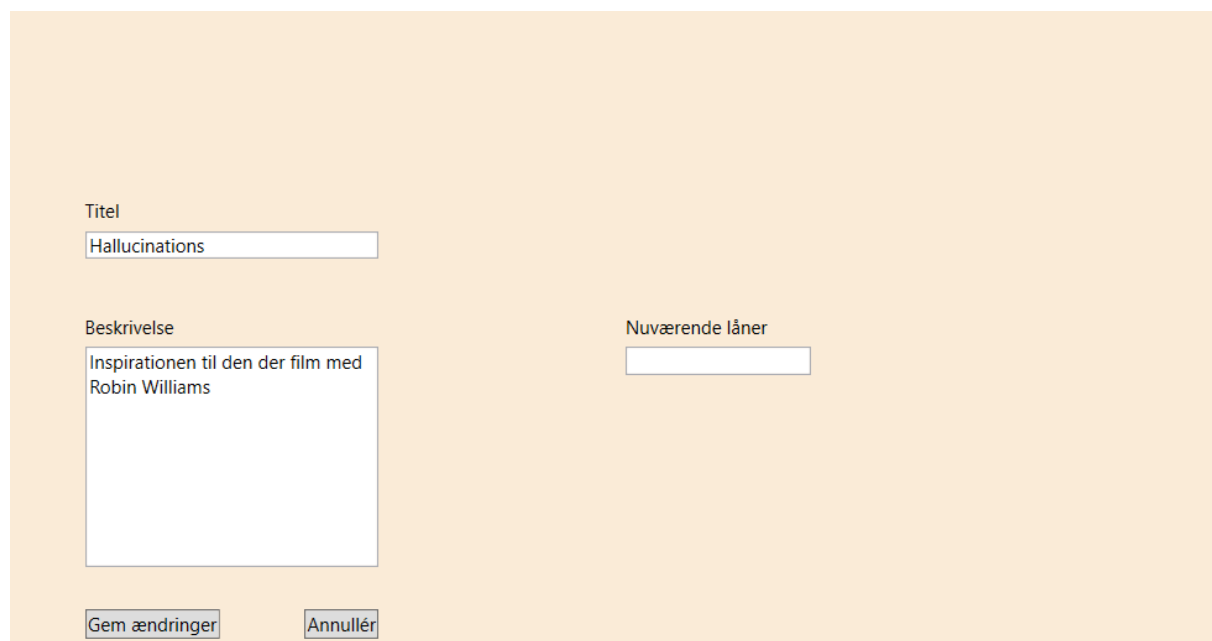
Jeg har lavet en lille applikation der meget simpelt emulerer et lånesystem på et bibliotek. Der er en række registrerede lånere samt et antal registrerede bøger, hvoraf der kun er ét eksemplar af hver.

En bog kan så registreres på en låner, og det vises så i en separat listbox hvilke bøger der er tilknyttet en valgt låner.

Applikationen



Ovenfor ses MainWindow, hvor oversigten over bøger, lånere og en given låners bøger. I dette tilfælde er det Polles to lånte bøger der vises.



Her vises det vindue der åbner, når man markerer en bog og trykker "Bogens Detaljer". Her kan man ændre properties på bogen og se den nuværende låner.

The screenshot shows a Windows application window with a light orange background. On the left, under the heading "Lånere", there is a list box containing three items: "ID: 1 - Navn: Polle", "ID: 2 - Navn: Lars", and "ID: 3 - Navn: Karsten". To the right of this list, there is a text input field labeled "Indtast maksimum alder". Below this field is a button labeled "Filtrer på alder". Further down, there is a label "Filtrer på property:" followed by a dropdown menu currently showing "ID". Below the dropdown is a button labeled "Reset filter". At the bottom of the window, there are two more text input fields: "Valgte låners alder" and "Valgte låners adresse".

Her er det vindue der åbner, når man trykker på "se lånere". Her kan man så sortere listen baseret på ID, alder, adresse og navn.

Væsentlige/vanskelige dele af programmet

Det er svært at bestemme hvad der er væsentligst, men det er i hvert fald godt at have styr på, hvordan en bogs detaljer vises i det nye vindue.

```
private void DetaljeBtn(object sender, RoutedEventArgs e)
{
    if (((Bog)BogBox.SelectedItem) != null) {
        var BogDet = new BogDetaljer(((Bog)BogBox.SelectedItem), context);
        BogDet.Owner = this;
        BogDet.ShowDialog();
        context.SaveChanges();
        BogBox.Items.Refresh();
        BeggeBox.Items.Refresh();
    }
    else
    {
        MessageBox.Show("Der skal vælges en bog at se detaljer på, før der kan vises detaljer for en bog.");
    }
}
```

Det nye vindue der skal vise detaljerne, tager den valgte bog fra BogBox og så vinduets BiblioteksContext som parametre, så vises det som en dialog hvor man kan manipulere dataen, og så bagefter gemmes ændringerne i context, og så bliver de to relaterede listboxes refreshed.

LinQ Queries

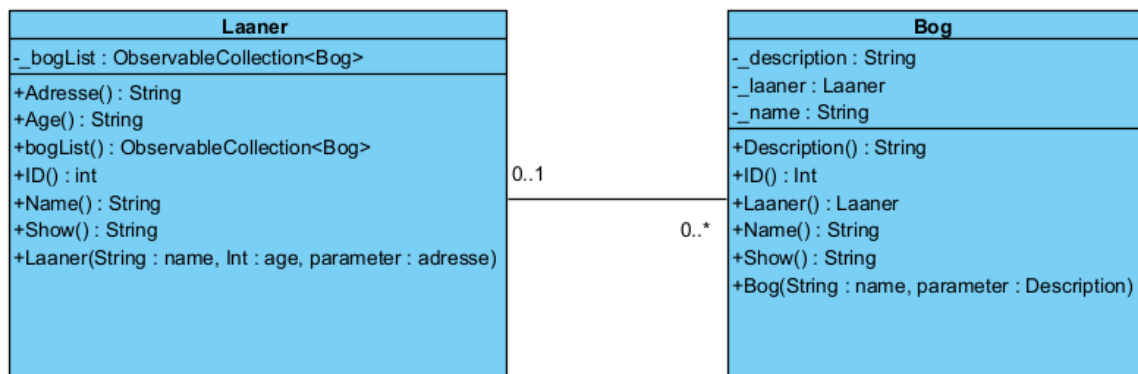
Jeg har benyttet LinQ queries i nogle tilfælde, til at filtrerer en liste af lånere baseret på forskellige variable:

```
1 reference
private void Adresse_Selected(object sender, RoutedEventArgs e)
{
    LaanerBox.ItemsSource = from laaner in Context.Laanerer.Local
                             orderby laaner.Adresse
                             select laaner;
}

1 reference
private void Navn_Selected(object sender, RoutedEventArgs e)
{
    LaanerBox.ItemsSource = Context.Laanerer.Local.OrderBy(i => i.Name);
}
```

De to metoder sorterer på henholdsvis navn og adresse. Funktionaliteten er den samme, men her er de to forskellige syntakser man kan skrive det på. Den ene minder meget mere om SQL-syntaks end den anden.

UML



Det er en forholdsvis simpel applikation. En låner kan have en-til-mange bøger, og en bog kan have en enkelt låner ad gangen.

Konklusion

Der var en del ting der kunne gøres bedre. For det første fandt jeg aldrig en måde at sætte `ItemsSource` i `BeggeBox` (listboxen for valgt låners bøger) til en query, og så stadig få den opdateret løbende. Ideelt ville den bare være noget a la `"ItemsSource = context.Boeger.Local.Where(i=> i.laanerID == valgtLaaner.ID)"`, men fordi jeg ikke kunne finde en måde at opdatere den listbox på

løbende når en bog blev tilføjet/fjernet den valgte låner, så endte jeg med lidt mere lavpraktisk at finde den valgte låners bogliste i context, og så smække den i BeggeBox.itemsSource for hver SelectionChanged event i lånernes listbox. Det krævede så, at jeg oprettede LaanerWindow for at vise, at jeg kunne bruge LinQ queries til sortering, men det har gjort applikationen lidt klodset. Et problem med det vindue er også, at man kan ikke filterer efter alder og en anden property samtidigt, fordi de hver især sætter itemsSource for listboxen til en query, og jeg har ikke kunne finde en måde at kombinere de to indenfor tidsrammen. Ud over det ville jeg have lavet en reactive search på lånere, så man kunne søge efter en given property i listen af lånere, og så få dem frem, der passede til kriteriet.