

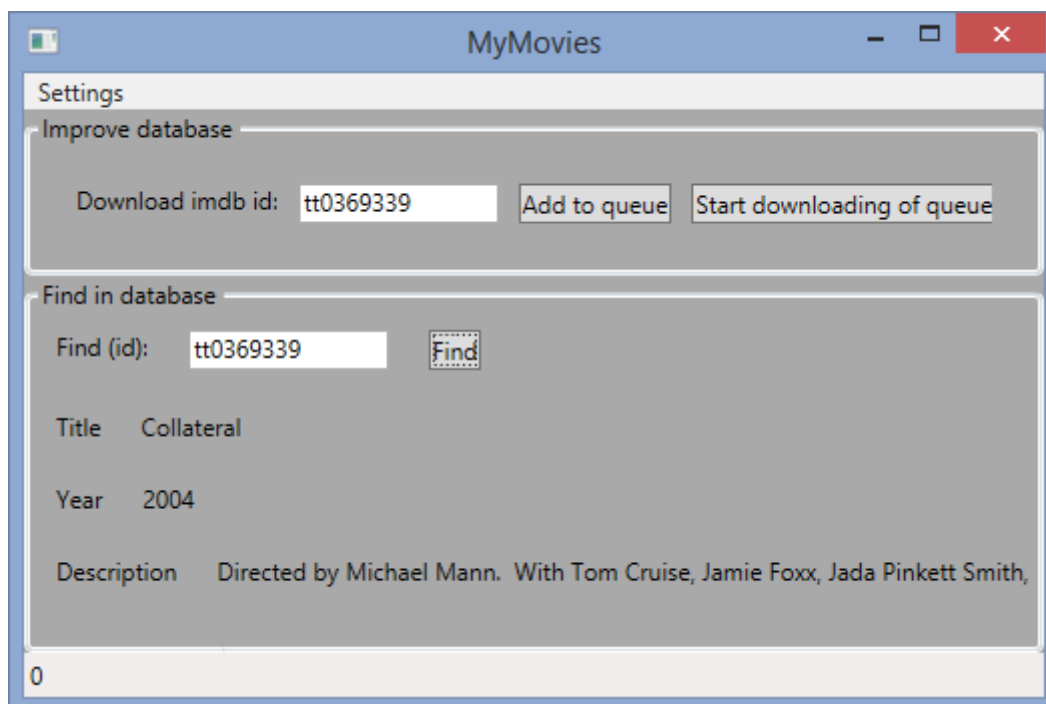
## My movie database

We maken een programma met 2 hoofdfunctionaliteiten:

- De gebruiker kan imdb id's invoeren en het programma zal de data die bij die id hoort (i.e. filmdata) downloaden en in een lokale "database" bewaren.
- De gebruiker kan zoeken of een bepaalde imdb id voorkomt in de lokale database, als dat het geval is dan wordt de filmdata getoond van dat id.

## Layout

De layout is louter illustratief en mag gewijzigd worden:



Er zijn 4 duidelijke onderdelen aanwezig in je programma:

1. Menu met daarin minstens 2 menuitems: "clear db" (maakt database leeg") en "db info" (toont aantal films in database)
2. "Improve database" waar gebruiker nieuwe id's kan invoeren aan queue die later gedownload moeten worden
3. "Find in database" dat gebruiker toelaat om adhv id meer filminfo te verkrijgen
4. Een status bar die toont hoeveel elementen nog in de downloadqueue verwerkt moeten worden

## Werking

**Improve database:**

- De gebruiker voegt via de textbox imdb id's (bv tt0369339) en drukt dan op 'add to queue'.
- De gebruiker kan dit blijven doen, telkens zal de statusbar geupdate worden om nieuwe aantal id's in queue te tonen

- Wanneer de gebruiker op 'start downloading of queue' klikt dan zal de queue 1 voor 1 verwerkt worden tot deze leeg is
- Tijdens het downloaden/verwerken van queue kunnen geen nieuwe id's worden toegevoegd
- Een Timer (zie verder) zal om de x seconden kijken of er nog iets in queue staat en, indien er nog iets in staat, het eerste element beginnen downloaden.
- Het gedownloade element wordt gescrape't (zie verder) en in een dictionary gestoken. De imdbid wordt gebruikt als de unieke key van ieder element in de dictionary. Er wordt minstens de film, het jaartal en een beschrijving bewaard in de database.
- De statusbar wordt geupdate na iedere download en toont hoeveel downloads er nog te gaan zijn.
- Er kunnen geen id's worden gedownload die reeds in de database aanwezig zijn.

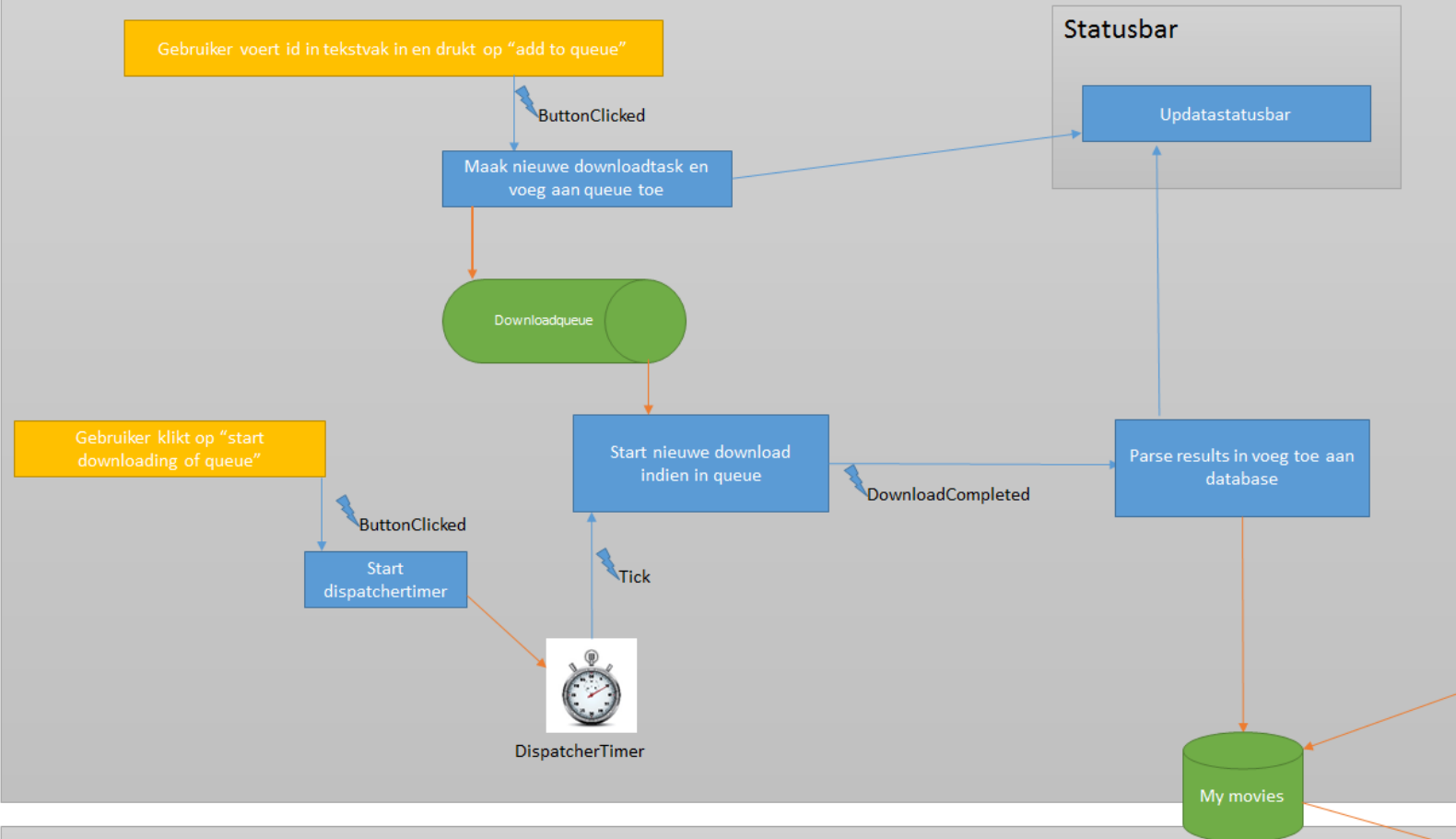
#### **Find in database:**

- Enkel indien een id in de dictionary is gevonden zal de info over die film getoond worden. Wordt het id niet gevonden dan zal via een MessageBox verteld worden dat deze film niet in de db staat

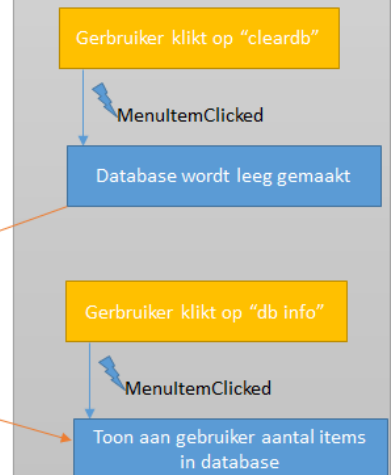
#### **Algemeen:**

- Je voorziet overal , waar relevant, exception handling
- Je voorziet adhv messagebox's feedback naar de gebruiker indien er iets verkeerd gaat of indien een actie klaar is (download, clear, etc)
- Je tracht de invulvelden en buttons enkel actief/inactief te maken indien dat relevant is (bv, de download starten kan enkel als er minstens 1 element in de queue staat)

## Improve db



## Menu



## Find in database



## Extra

Je vult meer functionaliteit naar keuze toe. Dit is het finale 'project' dus hoe meer hoe beter .

Denk bijvoorbeeld aan:

- het bewaren van de reeds gedownloade content in een lokaal bestand
- Meer informatie over film bewaren
- Poster/Cover van film bewaren en gebruiken als imagesource om poster te tonen
- Content in database aanpassen
- Films uit de database verwijderen

## Hulp

Hier enkele zaken/tips die je zeker zal nodig hebben:

### Queue en dictionary

Je werkt best met een queue voor de downloads en een dictionary voor de 'database':

```
Dictionary<string, Film> myDB
```

```
Queue<DownloadTask> dlQueue
```

Waarbij een DownloadTask minstens bestaat uit het id en de url die moet gedownload worden.

### Url to download

De url die je wil scrapen is van de vorm <http://www.imdb.com/title/{imdbid}>

Voor de film Collateral is deze dus: <http://www.imdb.com/title/tt0369339>

### Timer maken

Volgende code toont hoe je een timer maakt die elke 10 seconden (interval) een 'Tick'-event zal uitvoeren. In de Tick eventhandler kan je dan de nodige downloadcode (mbv WebClient) plaatsen:

```
DispatcherTimer ts = new DispatcherTimer();
ts.Tick += ts_Tick;
ts.Interval= new TimeSpan(0,0,4);
ts.IsEnabled = true;
ts.Start();
```

### Ik ben niet zo sterk in scrapen

Geen probleem. Volgende code zal de filmtitel, jaar van uitgave en beschrijving uit de filmpagina op imdb halen:

```
Film filmtoadd=null;
foreach (HtmlNode p in dldoc.DocumentNode.Descendants("meta")) //parse year and title
{
    if (p.Attributes["property"] != null && p.Attributes["property"].Value == "og:title")
    {
        //<meta property="og:title" content="The Hunger Games: Catching Fire (2013)">
        string movietitle = p.Attributes["content"].Value.Split('(')[0];
        int releaseyear =
            Convert.ToInt32(p.Attributes["content"].Value.Split('(')[1].Replace(")", string.Empty));
        filmtoadd = new Film(movietitle,releaseyear);
        break;
    }
}

if (filmtoadd != null)
{
    string description = null;
    foreach (HtmlNode p in dldoc.DocumentNode.Descendants("meta")) //parse description
    {
        if (p.Attributes["name"] != null && p.Attributes["name"].Value == "description")
        {
            description = p.Attributes["content"].Value;
            break;
        }
    }
    filmtoadd.Description= description;
}
```

Voorgaande code veronderstelt dat je een Film klasse hebt gemaakt met minstens 3 properties Title (string), Year (int), Description (string). Alsook dat de klasse een constructor heeft die 2 paramaters aanvaardt Film(string Title, int Year). Deze constructor wordt gebruikt in de geel aangeduide code.

Finaal zal dus in “filmtoadd” je film staan die je vervolgens in je dictionary moet plaatsen.

Protip: hier de voorgaande code mbv een Linq query:

```
HtmlDocument dldoc= new HtmlDocument();
dldoc.LoadHtml(e.Result);

var filmtoadd =
    (from p in
        dldoc.DocumentNode.Descendants("meta")
        where p.Attributes["property"] != null && p.Attributes["property"].Value == "og:title"
        select new Film(
            p.Attributes["content"].Value.Split('<')[0], Convert.ToInt32(p.Attributes["content"].Value.Split('<')[1].Replace(">", string.Empty)))
    ).FirstOrDefault();

if (filmtoadd != null)
{
    filmtoadd.Description= (from p in
        dldoc.DocumentNode.Descendants("meta")
        where p.Attributes["name"] != null && p.Attributes["name"].Value == "description"
        select p.Attributes["content"].Value).FirstOrDefault();
}
```

Ik wil via de download extra parameters meegeven

Je beseft het niet, maar deze vraag ga je stellen. Wanneer je via DownloadStringAsync gaat downloaden dan wil je kunnen meegeven van welke film je de download start, op die manier kan je in het DownloadStringCompleted event dit id gebruiken als key om de film aan de db toe te voegen (volg je nog?)

Stel dat je dus het id als string wil meegeven dan kan je de download als volgt starten:

```
wc.DownloadStringAsync(myUri, , "tt0369339");
```

Vervolgens kan je in het completed event die string uitlezen

```
void wc_DownloadStringCompleted(object sender, DownloadStringCompletedEventArgs e)
{
    if (e.Error == null)
    {
        string id = e.UserState as string;
```

Merk op dat je eender welk type kan meegeven en dus niet verplicht bent een string door te geven, je mag hier zelfs heelder objecten meegeven.